Formulierung von Nutzerpräferenzen in Multimedia-Retrieval-Systemen

Dissertation

zur Erlangung des akademischen Grades

Doktoringenieurin (Dr.-Ing.)

angenommen durch die Fakultät für Informatik
der Otto-von-Guericke-Universität Magdeburg

von: Dipl.-Wirtsch.-Inf. Nadine Schulz,
geboren am: 4. Juli 1976 in Gardelegen

Gutachter: Prof. Dr. Gunter Saake
Prof. Dr. Klaus Meyer-Wegener
Prof. Dr. Andreas Nürnberger

Ort und Datum des Promotionskolloquiums: Magdeburg, den 23. März 2004

Schulz, Nadine:
Formulierung von Nutzerpräferenzen in Multimedia-Retrieval-Systemen
Dissertation, Otto-von-Guericke-Universität
Magdeburg, 2004.

Kurzfassung

Gegenstand dieser Arbeit ist die Einbindung von Nutzerpräferenzen bei der Formulierung von Multimedia-Anfragen. Präferenzen bezüglich einzelner Suchbedingungen können auf zwei alternative Weisen spezifiziert werden: in Form von Halbordnungen sowie in Form von numerischen Gewichten. In dieser Arbeit wird die Spezifikation von Präferenzen durch Gewichte fokussiert. Dabei werden verschiedene Semantiken mit den Gewichten assoziiert. So können neben Relevanzgewichten auch Schwellwert- und Operatorgewichte eingesetzt werden.

Um eine adäquate Spezifizierung von Nutzerpräferenzen bei der Formulierung von Anfragen gewährleisten zu können, wird ein Multi-Level-Gewichtungsmodell entworfen. Dieses Anfragemodell unterstützt die Formulierung verschiedener Möglichkeiten der Gewichtung in komplexen Multimedia-Anfragen. Das Modell bietet die notwendige theoretische Basis, um neben der Angabe von Präferenzen bei der Anfrageformulierung eine iterative Anfragemodifikation durch entsprechende Relevanz-Feedback-Mechanismen zu ermöglichen.

Bestehende Multimedia-Anfragesprachen erlauben die Einbeziehung von Nutzerpräferenzen bei der Anfrageformulierung nur unzureichend. Aus diesem Grund wird aufbauend auf dem Multi-Level-Gewichtungsmodell die visuelle Multimedia-Anfragesprache WS-QBE entwickelt. Der Sprachentwurf berücksichtigt die allgemeinen Anforderungen an Anfragesprachen, die multimediaspezifischen Anforderungen sowie die Anforderungen bezüglich der Präsentation der Ergebnisse.

Die formale Semantik der Sprache WS-QBE wird durch die Abbildung auf ein Ähnlichkeitskalkül beschrieben. Das Ähnlichkeitskalkül integriert die verschiedenen Arten der Gewichtung und erlaubt somit eine adäquate Umsetzung des Multi-Level-Gewichtungsmodells. Ferner werden die Ausdrücke des Kalküls in jeweils einen Ausdruck einer Ähnlichkeitsalgebra überführt. Damit wird die Grundlage für eine algebraische Optimierung einer Anfrage geschaffen.

Danksagung

Diese Arbeit entstand am Institut für Technische und Betriebliche Informationssysteme der Fakultät für Informatik der Otto-von-Guericke-Universität. Ich möchte meinem Doktorvater Prof. Dr. Gunter Saake für die gute Unterstützung meines Promotionsvorhabens danken. Er gab mir die notwendige Freiheit, um an meiner Dissertation zu arbeiten. Bedanken möchte ich mich bei Prof. Dr. Klaus Meyer-Wegener und Prof. Dr. Andreas Nürnberger für die Übernahme der Koreferate sowie für ihre wertvollen Hinweise zur Strukturierung der Arbeit und die inspirierenden Gespräche.

Dr. Ingo Schmitt danke ich für die tolle Zusammenarbeit in den letzten Jahren und die zahlreichen Diskussionen, die wesentlich zum Gelingen der Arbeit beigetragen haben. Weiterhin möchte ich mich für seine wertvollen Anmerkungen und Hinweise zu meiner Dissertation bedanken.

Mein Dank gilt meinen Kollegen und den Technikern der Arbeitsgruppe Datenbanken für ihre Unterstützung während meiner Zeit am ITI. Ich danke den Studenten Marcus Denecke, Christian Urban und Nico Ladewig für die Unterstützung bei der prototypische Umsetzung mancher meiner Ideen.

Für die kritischen Anmerkungen zu verschiedenen Kapiteln dieser Arbeit möchte ich mich bei Christian Döring bedanken. Schließlich danke ich Ulli Schulz, Margrit Schulz, Annie Schulz und Roland Jesse für das geduldige Korrekturlesen dieser Arbeit sowie für die großartige Unterstützung während des Schreibens.

Inhaltsverzeichnis

1 Einleitung **1**
- 1.1 Zielstellung der Arbeit . . . 2
- 1.2 Ergebnisse . . . 3
- 1.3 Gliederung . . . 4

2 Multimedia-Information-Retrieval **7**
- 2.1 Taxonomie . . . 7
- 2.2 Information-Retrieval . . . 9
- 2.3 Multimedia-Information-Retrieval . . . 11
  - 2.3.1 Besonderheiten und Charakteristika . . . 12
  - 2.3.2 Multimedia-Retrieval-Prozess . . . 13
- 2.4 Einbindung von Nutzerpräferenzen . . . 16
  - 2.4.1 Präferenzanfragen . . . 18
  - 2.4.2 Gewichtete Anfragen . . . 20
  - 2.4.3 Relevanz-Feedback . . . 26
  - 2.4.4 Kontext-Feedback . . . 30
  - 2.4.5 Nutzerprofile . . . 31
- 2.5 Zusammenfassung . . . 33

3 Multimedia-Anfragesprachen **35**
- 3.1 Anforderungen an eine Multimedia-Anfragesprache . . . 35
  - 3.1.1 Allgemeine Anforderungen . . . 36
  - 3.1.2 Multimediaspezifische Anforderungen . . . 38
  - 3.1.3 Anforderungen an die Präsentation der Ergebnisse . . . 45
- 3.2 Prinzipien für den Entwurf einer Multimedia-Anfragesprache . . . 49
- 3.3 Bestehende Ansätze und Erweiterungen . . . 50
  - 3.3.1 Multimedia-Anfragesprachen . . . 50
  - 3.3.2 Erweiterungen des Bereichskalküls . . . 57
  - 3.3.3 Erweiterungen der relationalen Algebra . . . 58
  - 3.3.4 Erweiterungen von SQL . . . 63
- 3.4 Zusammenfassung . . . 67

4 Gewichtungsmodell für Multimedia-Anfragen **69**
- 4.1 Basisanfragemodell . . . 69
  - 4.1.1 Modell für einfache, gewichtete Anfragen . . . 69
  - 4.1.2 Anforderungen an gewichtete Scoring-Funktionen . . . 71
- 4.2 Verfahren zur Relevanzgewichtung von Anfragen . . . 74

4.2.1 Allgemeingültige Ansätze . . . 75
4.2.2 Spezielle, gewichtete Scoring-Funktionen . . . 79
4.2.3 Vergleich der Verfahren . . . 82
4.3 Multi-Level-Gewichtungsmodell . . . 84
4.3.1 Alternativen der Spezifikation von Relevanzgewichten . . . 85
4.3.2 Gewichtete Multi-Level-Scoring-Funktion für komplexe Anfragen . . . 86
4.3.3 Transformationsregeln für komplexe, gewichtete Anfragen . . . 88
4.3.4 Gewichtung der Operatoren . . . 93
4.3.5 Schwellwertgewichtung . . . 94
4.4 Zusammenfassung . . . 95

5 Ähnlichkeitskalkül und Ähnlichkeitsalgebra 97
5.1 Bereichskalkül . . . 97
5.2 Sichere Anfragen . . . 98
5.3 Ähnlichkeitskalkül . . . 100
5.3.1 Charakteristika . . . 100
5.3.2 Syntax . . . 102
5.3.3 Semantik . . . 108
5.4 Relationale Ähnlichkeitsalgebra . . . 114
5.4.1 Syntax . . . 114
5.4.2 Semantik . . . 116
5.5 Überführung eines Kalkülausdrucks in einen Algebraausdruck . . . 121
5.5.1 Reduktion . . . 121
5.5.2 Beispiel . . . 125
5.6 Zusammenfassung . . . 129

6 Die Anfragesprache WS-QBE 131
6.1 Motivation . . . 131
6.2 Query-By-Example . . . 132
6.2.1 Grundlagen von QBE . . . 132
6.2.2 SubQuery-By-Example . . . 137
6.2.3 Zusammenfassung . . . 138
6.3 Grundlegende Prinzipien . . . 139
6.3.1 Operationen . . . 140
6.3.2 Junktoren . . . 141
6.3.3 Formulierung von Anfragen . . . 141
6.3.4 Kontrolleinträge . . . 144
6.4 Syntax . . . 147
6.4.1 Prinzipien der Abbildung von WS-QBE auf Textkonstrukte . . . 147
6.4.2 Syntaxdefinition . . . 148
6.5 Semantik . . . 151
6.5.1 Operationen . . . 152
6.5.2 Junktoren . . . 152
6.5.3 Transformation einer WS-QBE-Anfrage in das $\mathcal{SDC}$ . . . 153
6.6 Beispiel . . . 155
6.7 Zusammenfassung . . . 158

7 Implementierung und Evaluierung **161**
7.1 Implementierung . . . 161
7.1.1 Transformationsregeln . . . 161
7.1.2 Ähnlichkeitskalkül und -algebra . . . 162
7.1.3 Gewichtete Ähnlichkeitssuche . . . 164
7.1.4 Räumliche Anfrageformulierung . . . 165
7.1.5 Anfrageformulierung mit WS-QBE . . . 166
7.2 Evaluierung der Anfragesprache . . . 168
7.2.1 Allgemeine Anforderungen . . . 169
7.2.2 Multimediaspezifische Anforderungen . . . 171
7.2.3 Anforderungen an die Präsentation . . . 172
7.2.4 Fazit . . . 172
7.3 Fallstudie zum Multi-Level-Gewichtungsmodell . . . 173
7.3.1 Vorbereitung . . . 173
7.3.2 Durchführung . . . 176
7.3.3 Auswertung . . . 177
7.4 Zusammenfassung . . . 179

8 Zusammenfassung und Ausblick **181**
8.1 Zusammenfassung . . . 181
8.2 Ausblick . . . 184
8.2.1 Algebraische Optimierung . . . 185
8.2.2 Relevanz-Feedback . . . 188
8.2.3 Fazit . . . 188

Literaturverzeichnis **191**

A Semantik der Ähnlichkeitsalgebra **209**

B Anfragen zur Evaluierung **213**

C Beweise der Transformationsregeln **215**
C.1 Idempotenz . . . 215
C.2 Involution . . . 215
C.3 De Morgan . . . 216
C.4 Distributivität . . . 217
C.5 Assoziativität . . . 226

1 Einleitung

In einer Welt der immer stärkeren Verbreitung digitaler Technologien nimmt Multimedia-Retrieval eine wachsende Rolle bei den informationsverarbeitenden Systemen ein. In zunehmendem Maße gewinnt die Recherche in digitalen Bibliotheken an Bedeutung. Dabei werden Datenbanken in einem immer größer werdenden Umfang von Personen genutzt, die über keine formale Datenbank-Ausbildung verfügen. So besteht beispielsweise der Wunsch eines Nutzers, in seiner digitalen Bildkollektion nach Bildern zu recherchieren, die einem gegebenen Musterbild ähnlich sind. Aber auch in der Geschäftswelt wird deutlich, dass der Bedarf der Multimedia-Suche in großen Kollektionen gegeben ist. Beispielsweise wird in dem LostArt-System die Recherche nach kriegsbedingt verbrachten Kulturgütern unterstützt und durch eine Bildähnlichkeitssuche ergänzt (Sattler u. a., 2002). Damit kann die Suche nach Doubletten und möglichen Übereinstimmungen zwischen Fund- und Suchmeldungen nicht nur auf Basis der textuellen Beschreibungen, die oftmals fehleranfällig sein können, sondern auch direkt auf Basis der Abbildungen erfolgen. Auf Grund des sich durch die Verbreitung digitaler Technologien neu erschließenden Personenkreises muss die Formulierung von Anfragen einfach sein, so dass auch Nutzer ohne detailliertes Datenbank-Wissen die gewünschten Informationen aus der Datenbank beschreiben können.

Daneben stellt die zunehmende Personalisierung der Recherche neue Herausforderungen an die Datenbank-Technologie. Dies verlangt das Bereitstellen flexibler und mächtiger Modellierungstechniken für komplexe Anfragen (Kießling, 2002). Eine wichtige Aufgabe ist dabei die Integration von Nutzerpräferenzen. In unserem täglichen Leben sind Präferenzen allgegenwärtig und nehmen eine wichtige Stellung in der Software-Gemeinschaft ein, wie die diversen Veröffentlichungen in (ACM, 2000) zeigen. Präferenzen können auf die unterschiedlichste Art und Weise ausgedrückt werden. So erweitern Präferenzanfragen die klassischen Datenbankanfragen um zusätzliche weiche Anfragebedingungen, wodurch die Probleme des *Empty-Result-Set* und *Flooding* klassischer Datenbankanfragen gelöst werden können (Kießling, 2002). Eine andere Möglichkeit besteht in dem Einsatz von Gewichten, wie beispielsweise von Relevanzgewichten. Diese Alternative wird oftmals bei der Abbildung der Nutzerwünsche im Rahmen des Multimedia-Retrievals verwendet (Fagin und Wimmers, 2000).

Eine weitere wichtige Tatsache, die in diesem Kontext immer mehr an Bedeutung gewinnt, liegt in der Diskrepanz zwischen der realen Welt und der exakten Datenbank-Welt begründet. In der klassischen exakten Datenbank-Welt sind Anfragen typischerweise durch scharfe Bedingungen charakterisiert, so dass diese Anfragen entweder voll oder gar nicht erfüllt werden. Hingegen werden in der realen Welt Anfragen oftmals vage ausgedrückt. Im Zusammenhang mit inhaltsbasierten Multimedia-Anfragen, also der Suche nach ähnlichen Objekten in einer Kollektion, sind unscharfe Anfragen unumgänglich. Darüber hinaus sind auch hier Nutzerpräferenzen bei der Formulierung von Anfragen zu berücksichtigen. Aus diesen Gründen erfolgt ein Paradigmenwechsel von den *exact-match Anfragen* zu den *best-match Anfragen*. Das bedeutet, dass nun diejenigen Objekte das Anfrageergebnis bilden, die

den Anfragebedingungen einschließlich der spezifizierten Präferenzen am besten gerecht werden. Die Realisierung unscharfer Anfragen und ein damit verbundenes Ranking der Ergebnisobjekte wird bereits seit langem von Information-Retrieval-Systemen unterstützt. Aber auch hier fehlt es bisher an Methoden zur Integration von Nutzerpräferenzen. Aufgabe der Datenbank-Technologie ist es nun, entsprechende Methoden und Lösungsansätze für Multimedia-Retrieval-Systeme bereitzustellen, die es erlauben, sowohl unscharfe Multimedia-Anfragen als auch Nutzerpräferenzen adäquat abzubilden.

1.1 Zielstellung der Arbeit

Derzeitig basiert die Anfrageformulierung in Datenbanken auf scharfen Selektionsbedingungen. Damit ergibt sich in klassischen Datenbanken mit Booleschen Anfragen das Problem, dass Anfragen häufig eine leere oder aber eine zu grosse Ergebnismenge zurückliefern.

Bei Multimedia-Anfragen, die neben exakten Prädikaten auch unscharfe Prädikate berücksichtigen, kann die Kardinalität der Ergebnismenge durch *top-k*-Anfragen derart beeinflusst werden, dass nur die k besten Objekte zurückgeliefert werden. Eine weitere Möglichkeit die Ergebnismenge einzuschränken, besteht in der Vorgabe von Schwellwerten.

Dennoch besteht bei Multimedia-Anfragen das Problem, dass das gesuchte Objekt mitunter nicht in der Ergebnismenge ist. Dies liegt zum Teil in der semantische Lücke zwischen Anwender und System begründet. Aus diesem Grund werden Techniken des Relevanz-Feedbacks eingesetzt, um das Anfrageergebnis so lange zu verfeinern, bis der Nutzer dem Ergebnis zustimmt. Durch diese Interaktion zwischen Nutzer und System wird versucht, die semantische Lücke zu schließen beziehungsweise dahingehend zu verringern, so dass der Suchwunsch des Nutzers ausreichend zufrieden gestellt wird.

Der Nachteil dieser Herangehensweise ist, dass erst im Nachhinein, also auf Basis der Anfrageergebnisse, die Präferenzen des Nutzers zum Tragen kommen. Dies geht jedoch nicht mit der realen Welt konform. Denn hier ist es oftmals so, dass der Nutzer bereits bei der Formulierung der Anfrage über bestimmte Präferenzen verfügt. Daher sollte dem Anwender schon bei der ersten Anfrageformulierung die Möglichkeit geboten werden, Präferenzen, beispielsweise bezüglich der Priorität oder Relevanz der einzelnen Kriterien, auszudrücken. Erstere finden in Form von Präferenzanfragen vermehrt Anwendung beim Daten- und Information-Retrieval (Kießling, 2002), während Letztere in Form von Gewichten besonders im Multimedia-Retrieval eingesetzt werden (Fagin und Wimmers, 2000). Die verschiedenen Modelle für eine Gewichtung von Anfragen berücksichtigen derzeitig meist nur die Formulierung von Präferenzen in einfachen Anfragen und unterstützten oftmals nur eine Art von Gewichten: die Relevanzgewichte. Andere Arten, wie Operator- oder Schwellwertgewichte, finden nur selten Anwendung. Eine Ausnahme stellt das Anfragemodell von Herrera-Viedma (2001a,b) dar, welches verschiedene Gewichtungssemantiken integriert. Der Nachteil dieses Ansatzes liegt jedoch darin begründet, dass in jeder Ebene einer komplexen Anfrage nur eine spezielle Art der Gewichtung unterstützt wird. So dürfen auf der Ebene der atomaren Anfrageterme nur Schwellwertgewichte und auf der Ebene der Teilanfragen nur Relevanzgewichte vergeben werden. Damit werden die Ausdrucksmöglichkeiten des Nutzers bezüglich der Formulierung seiner Präferenzen eingeschränkt.

Ein Ziel dieser Arbeit liegt in der Entwicklung eines *Multi-Level-Gewichtungsmodells* für die Abbildung von Nutzerpräferenzen durch unterschiedliche Arten von Gewichten in komple-

xen Multimedia-Anfragen. Damit können Suchanforderungen weit detaillierter als in gewöhnlichen Multimedia-Anfragen spezifiziert werden. Dadurch ist bereits die Initialanfrage gut auf die Nutzerbedürfnisse zugeschnitten, so dass bessere Anfrageergebnisse erzielt werden können.

Auch die Evaluierung gewichteter Anfragen durch sogenannte Scoring-Funktionen weist derzeit Defizite auf. So kann die Aggregation der Relevanzwerte unter Beachtung der Gewichte oftmals nicht adäquat erfolgen, da die verwendeten Scoring-Funktionen beispielsweise werden lokal linear noch streng monoton sind. Aus diesem Grund sollen zunächst im Gewichtungsmodell Anforderungen an solche Funktionen spezifiziert werden. Unter Berücksichtigung dieser Anforderungen können dann entsprechende Scoring-Funktionen abgeleitet werden.

Damit der Nutzer seine Präferenzen durch die verschiedenen Gewichte adäquat spezifizieren kann, müssen entsprechende Sprachmittel durch eine geeignete Multimedia-Anfragesprache bereitgestellt werden. Die in den vergangenen Jahren entwickelten Multimedia-Anfragesprachen sind oftmals textbasiert und stellen Erweiterungen der Structured Query Language (SQL) oder der Object Query Language (OQL) dar. Beliebt in Multimedia-Information-Retrieval-Systemen – da sehr einfach und nutzerfreundlich – ist die Formulierung von Query-By-Example-Anfragen. Aus diesem Grund wurden neue visuelle Multimedia-Anfragesprachen entwickelt. Nachteilig an diesen Sprachen ist, dass sie nicht universell einsetzbar sind und die Spezifikation von Nutzerpräferenzen nur bedingt unterstützen. Somit besteht auch hier ein erheblicher Forschungsbedarf.

Auf Grund der Defizite bestehender Multimedia-Anfragesprachen und der Vorteile visueller Sprachen ergibt sich direkt ein weiteres Ziel dieser Arbeit: der Entwurf einer Multimedia-Anfragesprache, welche der visuellen Anfragesprache Query-By-Example (QBE) ähnlich ist. Diese Multimedia-Anfragesprache wird als *Weighted-Similarity-Query-By-Example (WS-QBE)* bezeichnet. Sie soll die Umsetzung des Multi-Level-Gewichtungsmodells durch die Bereitstellung verschiedener Sprachkonstrukte zur Beschreibung der Präferenzen gewährleisten. Die formale Semantik der Sprache wird durch die Abbildung auf ein *Ähnlichkeitskalkül* beschrieben. Dieses wird, wie auch eine *Ähnlichkeitsalgebra*, formal eingeführt. Dabei wird zum Teil auf bestehende Erweiterungen der relationalen Algebra zurückgegriffen. Beispielsweise wird der *cut*-Operator aus der SAME^W-Algebra (Ciaccia u. a., 2000) übernommen.

Bedingt durch diverse, skalierbare Möglichkeiten der Anfrageformulierung in WS-QBE kann die Initialanfrage gut auf die Nutzerbedürfnisse abgestimmt werden. Dadurch können wiederum bessere Retrieval-Ergebnisse erzielt werden. Dennoch kann eine weitere iterative Anfrageverfeinerung notwendig sein. Das Gewichtungsmodell bietet für den Entwurf entsprechender Relevanz-Feedback-Methoden die notwendige theoretische Basis.

1.2 Ergebnisse

In dieser Arbeit wird das Multi-Level-Gewichtungsmodell als neuer Ansatz für die Abbildung von Präferenzen in komplexen Multimedia-Anfragen entwickelt. Dabei handelt es sich um ein Anfragemodell, das die Spezifikation von verschiedenen Arten von Gewichten auf allen Ebenen in komplexen Anfragen unterstützt. Damit ist neben einer Gewichtung einzelner atomarer Anfrageterme auch eine Gewichtung von Teilanfragen möglich. Der Nutzer erhält die Freiheit, zwischen verschiedenen Gewichtungssemantiken zu wählen. Dieses er-

laubt im Vergleich zu den bisherigen Ansätzen eine ausdrucksstärkere Formulierung von Nutzerpräferenzen.

Darüber hinaus werden Anforderungen an Scoring-Funktionen, die bei der Evaluierung gewichteter Anfragen genutzt werden, propagiert und Transformationsregeln für relevanzgewichtete, komplexe Anfragen entwickelt. Diese sind im Rahmen einer Anfrageoptimierung von Bedeutung.

Weiterhin wird eine neue visuelle QBE-ähnliche Multimedia-Anfragesprache entwickelt. Diese basiert auf dem Multi-Level-Gewichtungsmodell, integriert unterschiedliche multimediaspezifische Funktionalitäten und erlaubt die Abbildung von Nutzerpräferenzen. Die Semantik der Multimedia-Anfragesprache WS-QBE wird durch ein erweitertes Ähnlichkeitskalkül und durch eine entsprechende Ähnlichkeitsalgebra beschrieben. Letztere bildet die Grundlage für eine Anfrageoptimierung.

Das Neuartige an dieser Arbeit ist, dass eine durchgängige, in sich schlüssige Lösung – beginnend beim Anfragemodell, über eine entsprechende Erweiterung des Kalküls und der relationalen Algebra bis hin zu einer nutzerfreundlichen Multimedia-Anfragesprache realisiert wird.

1.3 Gliederung

Die Arbeit besteht im Wesentlichen aus drei Haupteilen (vgl. Abbildung 1.1). Im ersten Teil wird der Stand der Forschung beschrieben. Dem folgt der Kern der Arbeit mit der Entwicklung eines Gewichtungsmodells, der Beschreibung eines Ähnlichkeitskalküls und einer Ähnlichkeitsalgebra sowie dem Entwurf der Multimedia-Anfragesprache WS-QBE. Den letzten Hauptteil der Arbeit bildet die Beschreibung der Implementierung, die Evaluierung der Sprache und des Gewichtungsmodells sowie die Zusammenfassung der Arbeit. Ein Ausblick auf potentielle, weiterführende Forschungstätigkeiten rundet diesen Teil ab.

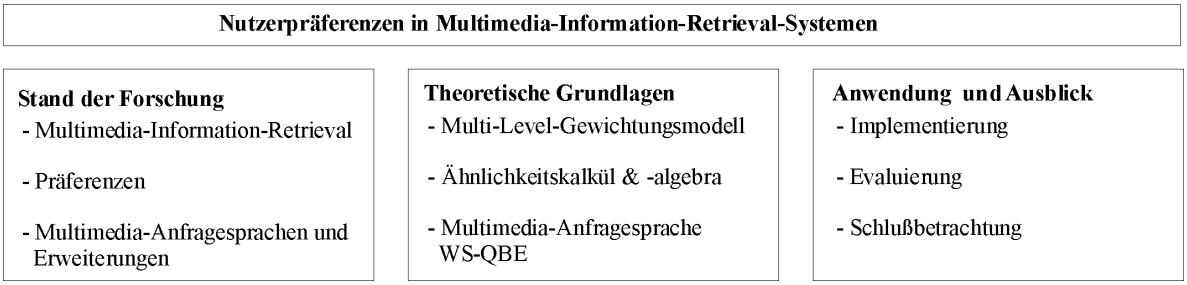


Abbildung 1.1: Aufbau der Arbeit

Konkret ist die Arbeit wie folgt aufgebaut: In Kapitel 2 werden die grundlegenden Begriffe dieser Arbeit definiert. Es wird ferner in das Gebiet des Information-Retrievals eingeführt und dabei auf die wichtigsten Retrieval-Modelle eingegangen. Darauf aufbauend erfolgt die Beschreibung des Multimedia-Information-Retrievals. Dabei werden die unterschiedlichen Arten des Multimedia-Retrievals skizziert. Ferner erfolgt eine Abgrenzung zum herkömmlichen Daten- und Information-Retrieval, indem die Besonderheiten und Charakteristika des Multimedia-Retrievals herausgearbeitet werden. Daran anschließend werden

die einzelnen Schritte des Multimedia-Retrieval-Prozesses beschrieben. Des Weiteren wird in diesem Kapitel auf die vielfältigen Möglichkeiten der Einbindung von Nutzerpräferenzen in den Retrieval-Prozess eingegangen. Es werden hierbei Möglichkeiten der Formulierung von Präferenzen sowohl bei der Formulierung der Anfrage als auch auf Basis der Ergebnisse durch Relevanz-Feedback dargelegt.

In Kapitel 3 steht die Evaluierung bestehender Multimedia-Anfragesprachen im Vordergrund. Von besonderem Interesse ist hierbei die Nutzerfreundlichkeit. Daher wird untersucht, inwieweit die Spezifikation von Nutzerpräferenzen durch die verschiedenen Multimedia-Anfragesprachen unterstützt wird. Grundlage für diese Untersuchung bilden die Anforderungen an Multimedia-Anfragesprachen. Diese ergeben sich zum einen aus den allgemeinen Anforderungen an Anfragesprachen und zum anderen aus den Besonderheiten des Multimedia-Retrievals. Anders als beim Daten- und Information-Retrieval ergeben sich weiterhin auf Grund der Verarbeitung von Multimedia-Daten spezielle Anforderungen an die Präsentation der Ergebnisobjekte. Diese Anforderungen werden zu Beginn des Kapitels genauer erläutert.

Neben verschiedenen Multimedia-Anfragesprachen existieren einige Erweiterungen des Bereichskalküls und der relationalen Algebra, die die Formulierung von Nutzerpräferenzen ermöglichen. Ferner gibt es diverse Erweiterungen der Anfragesprache SQL, die zwar selbst keine eigenständige Multimedia-Anfragesprache ausmachen, jedoch interessante Ansätze aufweisen, welche bei der Entwicklung einer Multimedia-Anfragesprache Berücksichtigung finden sollten. Diese Erweiterungen werden ebenfalls in diesem Kapitel vorgestellt.

In Kapitel 4 wird das Gewichtungsmodell zur Formulierung von Präferenzen in komplexen Multimedia-Anfragen durch Verwendung verschiedener Arten von Gewichten eingeführt. Die Ausgangsbasis dafür bildet ein einfaches Anfragemodell. Dieses unterstützt zunächst nur eine Relevanzgewichtung der Anfrageterme. Da die Evaluierung solcher relvanzgewichteten Anfragen unter Verwendung einer gewichteten Scoring-Funktion erfolgt, werden verschiedene Anforderungen an eine solche Funktion gestellt. Diese werden beschrieben. Daraufhin erfolgt eine Evaluierung derzeitig im Multimedia-Retrieval eingesetzter Scoring-Funktionen. Aufbauend auf dem Basisanfragemodell erfolgt die Entwicklung des Multi-Level-Gewichtungsmodells. Dieses gewährleistet die Formulierung von Präferenzen auf allen Ebenen in komplexen Multimedia-Anfragen. In diesem Modell werden verschiedene Arten der Gewichtung integriert. Hierbei findet die Anfrageevaluierung unter Berücksichtigung der Relevanzgewichte durch Verwendung einer gewichteten Multi-Level-Scoring-Funktion statt. Daher werden die Anforderungen, die für einfache gewichtete Scoring-Funktionen spezifiziert sind, erweitert. Es wird anschließend eine konkrete gewichtete Multi-Level-Scoring-Funktion vorgeschlagen. Daraufhin werden Transformationsregeln für gewichtete, komplexe Anfragen propagiert.

Das Kapitel 5 beschreibt die formale Semantik der Anfragesprache WS-QBE. Grundlage dafür bildet das Multi-Level-Gewichtungsmodell. Dieses wird in dem Ähnlichkeitskalkül umgesetzt. Damit werden die Grundlagen für eine deklarative Multimedia-Anfragesprache, welche die Möglichkeit bietet, Nutzerpräferenzen auf unterschiedliche Art und Weise abzubilden, gelegt. Ferner wird eine Ähnlichkeitsalgebra beschrieben. Sie bildet die Ausgangsbasis für eine algebraische Optimierung. Es ist somit möglich, einen beliebigen Anfrageausdruck aus dem Kalkül in einen entsprechenden Algebraausdruck zu überführen. Dafür wird ein Reduktionsalgorithmus angegeben. Die Überführung wird anhand eines Beispiels ausführlich beschrieben.

Den Schwerpunkt des Kapitels 6 bildet der Entwurf der visuellen Multimedia-Anfragesprache WS-QBE. Nach einer Motivation erfolgt eine kurze Beschreibung der Grundlagen der Anfragesprache QBE. Sie bildet die Basis für WS-QBE. Im Anschluss daran werden die grundlegenden Prinzipien von WS-QBE dargelegt. Es wird auf die unterstützten Prädikate und Junktoren eingegangen. Ferner werden die Möglichkeiten zur Formulierung von Anfragen und zur Spezifikation von Präferenzen geschildert. Es wird aufgezeigt, wie die Parametrisierung der Ergebnisausgabe erfolgen kann. Des Weiteren wird in diesem Kapitel sowohl die Syntax als auch die Semantik der Multimedia-Anfragesprache formal beschrieben.

In Kapitel 7 erfolgt die Beschreibung der Implementierung der verschiedenen umgesetzten Systemkomponenten. Darauf aufbauend findet eine Evaluierung der Multimedia-Anfragesprache WS-QBE statt. Hierfür werden einige typische Anfragen aus dem Kontext von LostArt mittels WS-QBE formuliert. Bei der Evaluierung steht die Untersuchung der Sprache WS-QBE bezüglich der Erfüllung der Anforderungen an eine Multimedia-Sprache im Vordergrund. Es wird verifiziert, inwiefern die in Kapitel 3 beschriebenen Anforderungen an eine Multimedia-Anfragesprache von WS-QBE erfüllt werden. Ferner wird in diesem Kapitel eine Fallstudie zum Multi-Level-Gewichtungsmodell vorgestellt. Dabei wird untersucht, ob mit diesem Gewichtungsmodell Nutzerpräferenzen adäquat umgesetzt werden können, so dass mit gewichteten Anfragen im Vergleich zu herkömmlichen Anfragen tatsächlich bessere Retrieval-Ergebnisse erzielt werden können.

Das Kapitel 8 enthält eine Zusammenfassung dieser Arbeit und einen Ausblick auf weitere Forschungsansätze. Auf zwei besonders interessante Forschungsthemen, die sich aus dieser Arbeit ergeben, wird dabei näher eingegangen. Dazu zählt zum einen die Entwicklung von neuen Relevanz-Feedback-Methoden auf Basis des Multi-Level-Gewichtungsmodells. Zum anderen stellt die algebraische Optimierung der Anfrageausdrücke eine vielversprechende Herausforderung dar.

2 Multimedia-Information-Retrieval

In diesem einleitenden Kapitel wird eine kurze Einführung in das Gebiet des Multimedia-Information-Retrievals gegeben, um die im weiteren Verlauf der Arbeit verwendete Terminologie festzulegen. Neben der Abgrenzung und Herausarbeitung der Charakteristika des Multimedia-Information-Retrievals erfolgt eine kurze Darstellung des Prozesses zum Retrieval von Multimedia-Objekten. Es wird ferner darauf eingegangen, inwieweit Nutzerpräferenzen in den Retrieval-Prozess eingebunden werden können.

2.1 Taxonomie

Zunächst soll der Begriff des *Mediums* festgelegt werden. Da sich ein Medium aus technischer Sicht anders als aus der Sicht der Rezipienten definiert und der Begriff des Mediums somit aus verschiedenen Blickwinkeln betrachtet werden kann, ist eine eindeutige Definition für diesen Begriff nur schwer zu finden. Es lassen sich also mehrere Medienbegriffe formulieren.

Im Allgemeinen wird der Begriff des Mediums als *Mittel* oder *Mittler* aufgefasst. Dementsprechend ist ein Medium etwas, was in irgendeiner Form vermittelt. Ein Sender erzeugt eine Nachricht, indem er auf ein bestimmtes Medium einwirkt. Das Medium ist sodann der Träger einer Nachricht und wird von einem Rezipienten betrachtet und analysiert. Nach Meyer-Wegener (2003) ist ein Medium ein Nachrichtenträger, welcher über eigene Darstellungsmittel verfügt, mit denen eine Nachricht gestaltet oder formuliert werden kann. Unter dem Begriff *Multimedia* wird eine Kombination verschiedener Medien verstanden.

Eine Einteilung von Medien kann entsprechend der menschlichen Sinnesorgane sowie unter sprachlichen und nicht-sprachlichen Aspekten erfolgen. Eine Unterscheidung der Medien in Abhängigkeit von der Zeit ist ebenso möglich. Hierbei wird zwischen *zeitabhängigen* (kontinuierlichen) Medien und *zeitunabhängigen* (diskreten) Medien differenziert (Prabhakaran, 1997, S. 3).

Aus Sicht der Datenbanken werden derzeitig sowohl akustische als auch visuelle Medien unterstützt. Dazu werden Text, Bild, Video und Audio gezählt. Eine Einbindung von haptischen, gustorischen oder olfaktometrischen Medien in Informationssysteme findet oftmals auf Grund fehlender Ein- und Ausgabegeräte nicht statt (Meyer-Wegener, 2003, S. 33). Um Nachrichten auf Computern verarbeiten zu können, müssen diese in digitaler Form vorliegen beziehungsweise in diese überführt werden. Ein *Medien-Objekt* ist daher ein Datenobjekt, das einem einzigen Medium angehört. Ein *Multimedia-Objekt* ist, entsprechend der Definition von Multimedia, eine Komposition von Medien-Objekten unterschiedlichen Typs (ebd., S. 35f).

Für das Information-Retrieval ist im Weiteren eine Abgrenzung der Begriffe *Daten*, *Wissen* und *Information* notwendig. Für die Begriffe Wissen und Information gibt es in den verschiedenen Anwendungsgebieten, ähnlich wie beim Medienbegriff, eine Vielzahl unterschiedlicher Definitionen. Losee (1998) führt eine disziplinübergreifende und grundsätzliche Dis-

kussion bezüglich der Definition des Begriffs Information. Er definiert Information als die Eigenschaften des Ergebnisses eines Prozesses, welche Rückschlüsse auf die Eingabe sowie den Prozess selbst zulassen. In diesem Vorschlag setzt Information stets bis zu einem gewissen Grad Wissen voraus. Diese Definition ist jedoch zu allgemein.

Eine engere Definition dieser Begriffe wird von Fuhr (1996) vorgeschlagen. Als Daten werden demnach Einträge verstanden, deren Typ oder syntaktische Struktur bekannt ist. Damit sind Daten auf einer rein syntaktischen Ebene angesiedelt. In diesem Kontext wird unter *Medien-* beziehungsweise *Multimedia-Daten* eine heterogene Menge von Medien- beziehungsweise Multimedia-Objekten verstanden. Ist zusätzlich zu den Daten eine Semantik gegeben, das heißt, es ist bekannt, wie die Daten zu interpretieren sind, handelt es sich um *Wissen*. Während Wissen auf der semantischen Ebene angesiedelt ist, so ist *Information* auf der pragmatischen Ebene angesiedelt und kann als Wissen in einem Kontext betrachtet werden. Information ist kontextabhängig. Somit ist Information im Gegensatz zu Wissen flüchtig (Fuhr, 1996, S. 16).

Andere Definitionen verwenden die Begriffe Information und Wissen umgekehrt, wie beispielsweise in (Shedroff, 1999). Als Information werden dann semantisch interpretierte Daten bezeichnet. Sie werden zu Wissen, wenn Menschen sie wahrnehmen, verarbeiten und reproduzieren oder nutzen können (Ferber, 2003).

Nachdem nun die Begriffe des Mediums und der Information im Kontext von Datenbanken dargelegt wurden, erfolgt die Definition eines *Multimedia-Information-Retrieval-Systems (MIRS)*. Nach Steinmetz ist ein *Multimedia-System* durch die rechnergesteuerte, integrierte Erzeugung, Manipulation, Darstellung, Speicherung und Kommunikation von unabhängigen Informationen, die in mindestens einem diskreten und einem kontinuierlichen Medium kodiert sind, charakterisiert (Steinmetz, 1999, S. 13). Wesentlich an dieser Definition eines Multimedia-Systems ist, dass sowohl zeitabhängige als auch zeitunabhängige Medien-Objekte verarbeitet werden.

Ein Multimedia-Information-Retrieval-System wird nach Schäuble (1997) durch die folgenden vier Merkmale charakterisiert:

1. *Multimedia-Objekte;* Das System kann verschiedene Multimedia-Objekte speichern. Diese bestehen aus diversen Multimedia-Daten und können semistrukturiert sein. Das heißt, dass die Multimedia-Objekte sowohl strukturierte als auch unstrukturierte Daten enthalten.

2. *Retrieval-Modell;* Das System basiert auf einem effizienten Retrieval-Modell, welches ein Ranking der Ergebnisobjekte unterstützt.

3. *Objektanalyse;* Die Medien-Objekte werden aufbereitet, um die implizite Semantik der Medien-Objekte zu extrahieren. Das Ergebnis einer Objektanalyse sind statistische Informationen, auf denen das Retrieval aufsetzt.

4. *Interaktive Retrieval-Techniken;* Das System unterstützt Interaktionen mit dem Nutzer, um die Wahrscheinlichkeit einer erfolgreichen Suche zu erhöhen. Hierzu zählen Techniken des Relevanz-Feedbacks. Das Retrieval-System kombiniert die Feedback- und die statistischen Informationen zu einer neuen Anfrage (Schäuble, 1997, S. 5ff).

Auf diese einzelnen Merkmale wird im Zusammenhang mit der Vorstellung des Multimedia-Information-Retrievals in Abschnitt 2.3 näher eingegangen.

Zusammenfassend ist ein MIRS ein Informationssystem, das eine Aufbereitung, Speicherung, Suche und Präsentation von zeitabhängigen und zeitunabhängigen Multimedia-Daten unterstützt sowie eine Interaktion mit dem Nutzer erlaubt. Die *Information-Retrieval-Systeme (IRS)* bilden die Vorgänger der MIRS und unterstützen nur ein Retrieval von Textdokumenten. Ein IRS ist somit lediglich durch die Merkmale 2 bis 4 gekennzeichnet und verarbeitet ausschließlich das Medium Text.

Im Folgenden wird auf das *Information-Retrieval* und das *Multimedia-Information-Retrieval*, welche jeweils durch die entsprechenden Retrieval-Systeme unterstützt werden, näher eingegangen.

2.2 Information-Retrieval

Ziel des Information-Retrievals ist es, alle Medien-Objekte beziehungsweise Dokumente zu finden, die bezüglich der Anfrage eines Anwenders relevant sind. Wichtig hierbei ist, dass sich das Information-Retrieval nur auf Textdokumente bezieht. Daher wird der Begriff *Text-Retrieval* synonym für Information-Retrieval verwendet. Den zentralen Aspekt in Information-Retrieval-Systemen stellt die Relevanz der Dokumente dar (Baeza-Yates und Ribeiro-Neto, 1999, S. 19). Zur Bestimmung der Relevanz werden entsprechende Ranking-Algorithmen eingesetzt.

Im Gegensatz zu klassischen Datenbankanfragen zeichnen sich Anfragen im Information-Retrieval durch ihre unscharfe Formulierung aus, so dass eine gewisse Toleranz gegenüber Fehlern besteht. Des Weiteren können Anfragen im Information-Retrieval iterativ verfeinert werden. Eine Anfrage liefert unscharfe Ergebnisse, die anhand ihrer Relevanz bezüglich der Anfrage sortiert werden. Daher wird im Gegensatz zum Daten-Retrieval anstatt einer Ergebnismenge eine -liste geliefert. Ferner ist beim Information-Retrieval im Gegensatz zum Daten-Retrieval eine Interpretierung der Dokumente durch eine Extraktion von syntaktischen und semantischen Informationen notwendig (ebd., S. 1f). Somit sind vage Anfragen und unsicheres Wissen kennzeichnend für das Gebiet des Information-Retrievals (Fuhr, 1996, S. 16).

Dem Information-Retrieval liegen verschiedene Modelle zu Grunde. Jedes IRS setzt dabei auf einem Retrieval-Modell auf. Zu den bekanntesten Modellen zählen folgende Ansätze (Baeza-Yates und Ribeiro-Neto, 1999, S. 24ff):

Boolesches Modell; Bei dem Booleschen Modell handelt es sich um ein einfaches Retrieval-Modell. Anfragen werden durch Boolesche Ausdrücke scharf formuliert. Jedem Anfrageterm, also jeder einzelnen Suchbedingung, wird ein Gewicht zugeteilt wird. Diese werden als Termgewichte bezeichnet und sind in diesem Modell binär. Die Retrieval-Strategie basiert auf Boolescher Logik. Daher ist ein Dokument entweder relevant oder nicht relevant bezüglich einer Anfrage. Ein Ranking des Anfrageergebnisses ist nicht möglich. Die Möglichkeit der Gewichtung von Anfragetermen ist im Booleschen Modell nicht gegeben.

Fuzzy-Modell; Das Fuzzy-Retrieval basiert auf Fuzzy Logik (Zadeh, 1965) und stellt eine Erweiterung des Booleschen Retrievals dar. Die Termgewichte liegen in dem Wertebereich $[0, 1]$. Der Vorteil dieses Retrieval-Modells besteht darin, dass ein Ranking der Ergebnisse auf Basis des errechneten Retrieval-Status-Values (RSV) stattfinden kann.

Der RSV wird auch als Relevanzwert bezeichnet. Eine Möglichkeit zur Gewichtung von Anfragetermen wurde von Salton u. a. (1983b) vorgeschlagen.

Vektorraum-Modell; Im Vektorraum-Modell, vorgestellt von Salton (1971) und überarbeitet von Raghavan und Wong (1986), werden sowohl die Dokumente als auch die Anfrage selbst als Punkte im Vektorraum aufgefasst. Der Raum wird durch die einzelnen Terme aufgespannt. Dadurch können zu einer Anfrage beziehungsweise zu ihrem Anfragevektor die ähnlichsten Dokumente gefunden werden. Als Retrieval-Funktionen werden verschiedene Vektor-Ähnlichkeits- beziehungsweise -Unähnlichkeitsmaße, wie beispielsweise das Kosinusmaß oder die Euklidische Distanz, eingesetzt. Die Rangfolge der Dokumente kann auf Basis des errechneten Ähnlichkeitswertes festgelegt werden.

Probabilistisches Modell; Das probabilistische Modell, eingeführt von Robertson und Jones (1976), basiert auf dem Wahrscheinlichkeitsmodell und arbeitet mit statistischen Methoden. Ziel ist es, für eine Anfrage die Wahrscheinlichkeit zu bestimmen, mit der ein Dokument relevant für den Anwender ist. Ein Ranking erfolgt auf Basis des errechneten Wahrscheinlichkeitswertes.

Alle Retrieval-Modelle können eine Boolesche Fragestruktur umsetzen. Das heißt, die Anfragen setzten sich aus einzelnen Anfragetermen (Suchbegriffen) zusammen, die mittels der Booleschen Operatoren verknüpft werden. Darüber hinaus können Anfragen im Vektorraum-Modell sowie im Probabilistischen Modell eine lineare Fragestruktur aufweisen. Damit besteht hier die Möglichkeit, Anfragen natürlichsprachlich zu formulieren.

Tabelle 2.1 fasst die hier vorgestellten Information-Retrieval-Modelle zusammen. Für die Einschätzung der Modelle ist die theoretische Basis, die Unterstützung einer gewichteten Indexierung, die Art der Fragestruktur sowie die Möglichkeit der Gewichtung von einzelnen Anfragetermen von Bedeutung.

Merkmal	Boolesches Modell	Fuzzy-Modell	Vektor-Modell	Probabilistisches Modell
Theorethische Basis:				
- Boolesche Logik	×			
- Fuzzy-Logik		×		
- Vektoralgebra			×	
- Wahrscheinlichkeitstheorie				×
Gewichtete Indexierung		×	×	×
Gewichtete Anfrageterme		×	×	×
Fragestruktur:				
- linear			×	×
- Boolesch	×	×	×	×

Tabelle 2.1: Überblick über Information-Retrieval-Modelle (Auszug aus Fuhr (1996), S. 85)

2.3 Multimedia-Information-Retrieval

Das Multimedia-Information-Retrieval stellt eine Erweiterung des Information-Retrievals dar, so dass neben Text-Dokumenten auch andere Medien- beziehungsweise Multimedia-Objekte in das Retrieval einbezogen werden. So bildet bereits seit den 80er Jahren des 20. Jahrhunderts das Bild-Retrieval einen Forschungsschwerpunkt (Chang und Fu, 1981). Seit Beginn der 1990er Jahre wird jedoch erst das Problem des Video-Retrievals fokussiert (Amato u. a., 1998).

Um ein Retrieval der Multimedia-Daten zu gewährleisten, sind Interpretationen der Daten notwendig. Diese werden als *Metadaten* bezeichnet. Metadaten werden automatisch, semiautomatisch oder manuell gewonnen und können wie folgt klassifiziert werden (Prabhakaran, 1997, S. 53f):

- *Inhaltsabhängige Metadaten* hängen vom Inhalt der Medien-Objekte ab. Dazu zählen automatisch extrahierte Features. Typische Features eines Bildes sind beispielsweise Farbe, Form und Textur. Rui u. a. (1998b) modellieren formal den Inhalt eines Medien-Objektes O als Tripel:

  $$O(D, F, R),$$

  wobei

  - D die Rohdaten des Objektes sind,
  - $F = \{f_i\}$ eine Menge von Features, wie beispielsweise Farbe und Form, ist und
  - $R = \{r_{i,j}\}$ eine Menge von Feature-Repräsentationen ist, wie beispielsweise Farbhistogramme und Farbmomente für das Feature Farbe. Jede Feature-Repräsentation kann ein Vektor sein, der die entsprechenden Feature-Werte enthält.

  Diese Darstellung eines Medien-Objektes ist im Vergleich zu anderen Ansätzen sehr feingranular. Amato u. a. (1998) extrahieren Feature-Werte für jedes Feature, so dass die einzelnen Feature-Repräsentationen zu einem Feature zusammengefasst sind.

- *Inhaltsbeschreibende Metadaten* können zur Zeit nicht automatisch extrahiert werden, sondern werden durch den Nutzer, beispielsweise durch Annotation der Medien-Objekte, vorgegeben. Eine Beschreibung kann als zusätzliches Attribut abgelegt oder in Form von Freitext formuliert werden. Ferner kann auch eine Indexierung der Medien-Objekte durch die Vergabe von Schlüsselwörtern aus einem festgelegten Vokabular erfolgen. Eine solche Verschlagwortung beziehungsweise Beschreibung von Medien-Objekten ist häufig aufwändig, subjektiv und somit fehleranfällig.

- *Inhaltsunabhängige Metadaten* sind vollständig unabhängig vom Inhalt der Medien-Objekte. Dazu zählen zum Beispiel der Name des Regisseurs eines Videos oder die Dauer der Videoerstellung. Diese Metadaten können als zusätzliche Attribute abgelegt werden.

Die einfachste Möglichkeit des Retrievals von Multimedia-Objekten stellt das *schlüsselwortbasierte Retrieval* dar. Auf Basis der inhaltsbeschreibenden und inhaltsunabhängigen Metadaten der Medien-Objekte können bestehende Konzepte des Daten- oder Text-Retrievals auch auf andere Medien-Objekte, wie Bilder oder Videos, angewendet werden.

Beim *inhaltsbasierten Multimedia-Information-Retrieval* werden Objektinformationen automatisch mittels verschiedener Extraktionsverfahren als Feature-Werte ermittelt. Pro Feature werden die dabei gewonnen Feature-Werte in einem Vektor abgelegt. Beim inhaltsbasiertem Retrieval wird ein Medien-Objekt somit durch einen Feature-Vektor charakterisiert, so dass die Objekte als Punkte im Raum aufgefasst werden können. Aus diesem Grund kann als Multimedia-Retrieval-Modell das Vektorraum-Modell verwendet werden. Inhaltsbasierte Anfragen werden oftmals auch als *Ähnlichkeitsanfragen* bezeichnet.

Neben diesen beiden möglichen Ansätzen des Multimedia-Information-Retrievals heben Gudivada und Raghavan (1995) weiterhin das *konzeptbasierte Multimedia-Retrieval* hervor. Für ein derartiges Retrieval wird anwendungsspezifisches Wissen benötigt, um den Inhalt der Medien-Objekte zu interpretieren. Bei der Interpretation werden Konzepte ermittelt, die es erlauben, direkt auf die Objekte zurückzugreifen. Konzepte werden durch Deskriptoren beschrieben und sind hierarchisch organisiert, so dass es zu jedem Oberbegriff mehrere spezifischere Unterbegriffe geben kann. Beispielsweise ist das durch den Deskriptor *Flugzeug* bezeichnete Konzept ein Oberkonzept des durch den Deskriptor *Sportflugzeug* bezeichneten Konzepts. Jedem Medien-Objekt kann somit ein entsprechendes Konzept zugeordnet werden. Im Allgemeinen ist ein konzeptbasiertes Retrieval anwendungsabhängig und erfordert beim Interpretationsprozess die Unterstützung des Nutzer.

2.3.1 Besonderheiten und Charakteristika

Ein wesentliches Charakteristikum von Multimedia-Information-Retrieval-Systemen besteht darin, dass sie eine Vielzahl unterschiedlicher Daten unterstützen (Bertino u. a., 1999). Diese Heterogenität macht die Handhabung verschiedenster Medientypen und Speicherformate notwendig. So müssen beispielsweise für jeden Medientyp spezielle Verfahren für die Feature-Extraktion oder die Ähnlichkeitsberechnung bereitstehen. Die Ein- und Ausgabe der Medien-Objekte ist vom jeweiligen Medientyp und Speicherformat abhängig. Des Weiteren müssen Methoden für eine Umwandlung des Speicherformates oder für Medienumwandlungen zur Verfügung stehen.

Ähnlich wie beim Information-Retrieval ist die Semantik implizit in den Medien-Objekten gegeben. Daher müssen spezielle Feature-Werte extrahiert werden. Die dafür eingesetzten Verfahren isolieren die semantiktragenden Eigenschaften aus den Medien-Objekten. Ein wesentliches Problem, welches hierbei auftritt, ist die sogenannte *semantische Lücke* (Gudivada und Raghavan, 1995). Diese bezeichnet den Unterschied zwischen dem subjektiven Eindruck, den der Anwender von einem Medien-Objekt hat und der Semantik, die durch Algorithmen extrahiert werden kann.

MIRS zeichnen sich weiterhin dadurch aus, dass sie ein sehr hohes Datenvolumen verwalten müssen. Dieses macht effiziente Methoden zum Speichern der Medien-Objekte erforderlich. Nicht nur die Kompression der Daten ist dafür wichtig, sondern auch eine kompakte Präsentation der Medien-Objekte. Für Darstellungszwecke, wie die überblicksartige Präsentation der Retrieval-Ergebnisse, können daher beispielsweise Thumbnails für die Visualisierung von Bildern verwendet werden.

Neben den Besonderheiten, die direkt aus den Multimedia-Daten herrühren, müssen MIRS in der Lage sein, temporale und räumliche Beziehungen zwischen den Medien-Objekten abzubilden (Fuhr u. a., 1998). Daher muss die Möglichkeit bestehen, neben typischen Daten-Retrieval-Anfragen auch inhaltsbasierte, unscharfe, zeitliche und räumliche Anfragen zu for-

mulieren. Diese Besonderheiten treten in Information-Retrieval-Systemen und klassischen Datenbank-Management-Systemen (DBMS) nicht auf.

2.3.2 Multimedia-Retrieval-Prozess

Der Multimedia-Retrieval-Prozess läuft, wie in Abbildung 2.1 verdeutlicht, in mehreren Schritten ab:

1. **Datenbankbestückung:** In diesem ersten Schritt wird die Datenbank aufgebaut. Neben der Speicherung der Medien-Objekte als Rohdaten können zusätzlich beschreibende Informationen als relationale Daten gespeichert werden. Des Weiteren können Strukturdaten und Metadaten, wie beispielsweise das vorliegende Speicherformat, abgelegt werden.

   Um Medien-Objekte in einer Datenbank abzuspeichern, werden diese zunächst aufbereitet. Konkret werden komplexe Multimedia-Objekte in Medien-Objekte und Strukturdaten zerlegt. Zur Unterdrückung beziehungsweise Eliminierung von Störfaktoren werden die Medien-Objekte normalisiert. Dann kann eine Segmentierung des Medien-Objekts erfolgen. Beispielsweise kann ein Bild in verschiedene Regionen zerlegt werden.

2. **Feature-Extraktion:** Für das Multimedia-Retrieval ist es notwendig, dass neben den Rohdaten auch inhaltsabhängige Metadaten zur Verfügung stehen. Diese beschreiben die implizite Semantik der Medien-Objekte. Für die Beschreibung der Semantik werden Feature-Werte aus den Objekten extrahiert. Die Feature-Extraktion kann dynamisch oder statisch erfolgen. Bei der dynamischen Feature-Extraktion werden die Feature-Werte direkt nach oder während der Anfrageaufbereitung ermittelt. Dieses bietet den Vorteil, dass die Extraktion relevanter Features flexibel an die Anfrage angepasst werden kann. Dies ist jedoch mit einem sehr hohen Aufwand verbunden. Die Kosten für die Anfragebearbeitung sind bei einer dynamischen Feature-Extraktion somit sehr hoch. Demgegenüber bietet die statische Feature-Extraktion den Vorteil, dass die Anfrage kosteneffizienter ausgeführt werden kann, da hier, bevor die Anfrage gestellt wird, bereits alle Feature-Werte für die Datenbankobjekte ermittelt und unter Umständen bereits indexiert wurden. Eine Kombination beider Verfahren ist möglich. Gängige Features können damit vor der Anfrageaufbereitung extrahiert werden. Spezielle Features werden dynamisch während oder nach der Anfrageaufbereitung berechnet.

   Die ermittelten Feature-Werte werden dann aufbereitet, so dass Abhängigkeiten zwischen den Feature-Werten oder vernachlässigbare Werte eliminiert werden. Dafür werden spezielle Transformationsalgorithmen eingesetzt.

   Um ein effizientes Retrieval zu unterstützen, werden Indexe für die Feature-Werte angelegt. Da Feature-Werte als Vektoren im n-dimensionalen Raum dargestellt werden können, werden als Indexstrukturen oftmals das Grid-File (Nievergelt u. a., 1984), der R-Baum (Guttman, 1984) und verschiedene Variationen davon (Beckmann u. a., 1990; Sellis u. a., 1987) sowie der TV-Baum (Lin u. a., 1994) eingesetzt.

   Auf Basis der gewonnenen Feature-Werte und mit Hilfe von anwendungsspezifischem Wissen kann die Ableitung von Konzepten erfolgen. Dadurch werden Zusammenhänge zwischen den Objekten hergestellt.

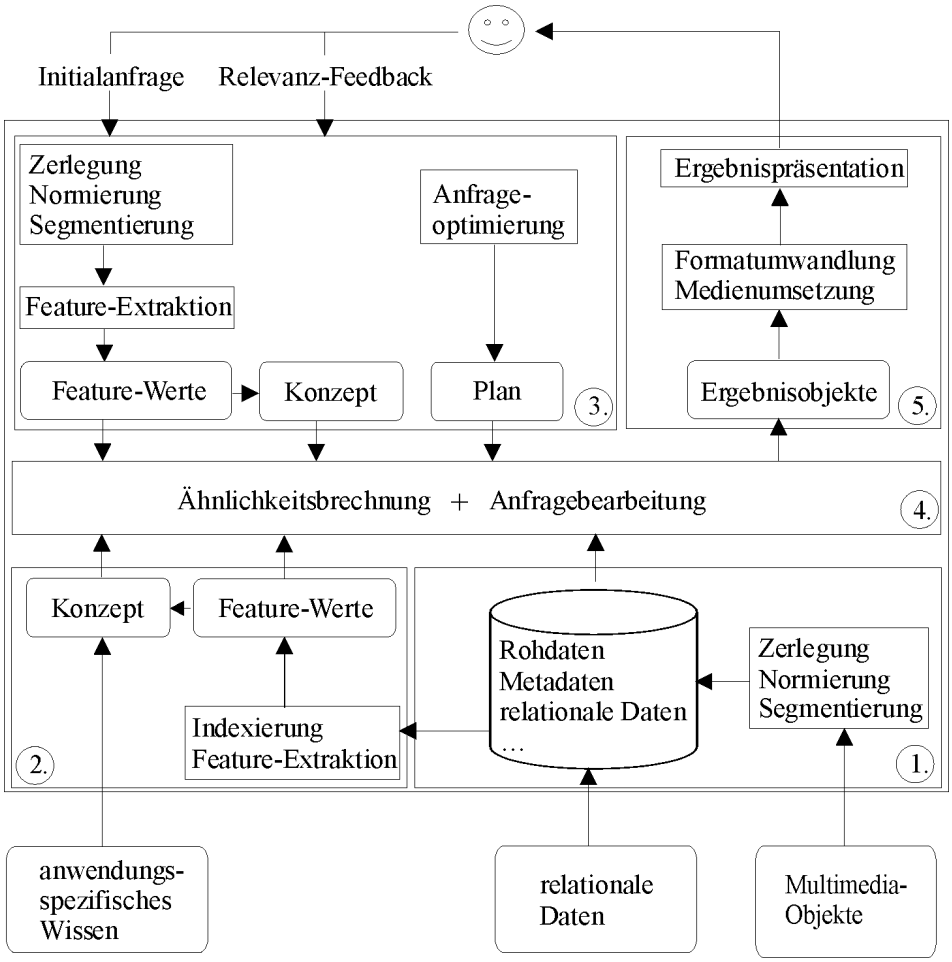


Abbildung 2.1: Multimedia-Retrieval-Prozess (angelehnt an Schmitt)

3. **Anfrageaufbereitung:** Zunächst wird eine Anfrage vom Anwender formuliert. Jain (1996) unterscheidet zwischen *symbolischen Anfragen* und *Query-by-Example*. Bei symbolischen Anfragen hat der Nutzer bereits eine präzise Vorstellung davon, was gesucht ist und welche Informationen mit den Multimedia- beziehungsweise Medien-Objekten in der Datenbank assoziiert werden. Die Anfragen werden in einer SQL-ähnlichen Anfragesprache formuliert. Bei der Formulierung von Query-by-Example-Anfragen gibt der Nutzer ein oder mehrere Medien-Objekte vor, welche die Basis für eine Ähnlichkeitsanfrage bilden.

   Nach der Anfrageformulierung erfolgt die Anfrageaufbereitung. Die Medien- beziehungsweise Multimedia-Objekte der Anfrage werden gegebenenfalls zerlegt und normalisiert. Es werden Feature-Werte extrahiert und Konzepte interpretiert. Eine Anfrageoptimierung wird in diesem Schritt ebenfalls vorgenommen. Als Ergebnis liegt ein entsprechender Anfrageplan vor.

4. **Anfragebearbeitung:** Bei einer Ähnlichkeitsanfrage erfolgt auf Basis der gewonnenen Fea-

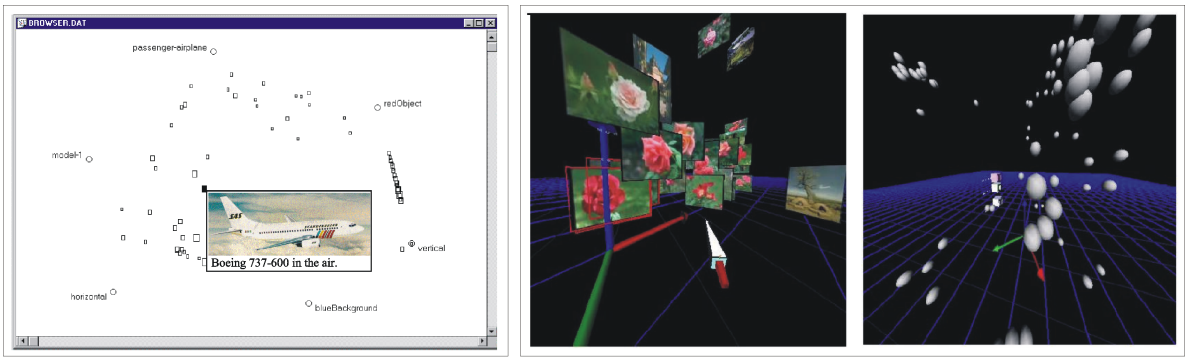
ture-Werte für das Anfrageobjekt und dem berechneten Anfrageplan die Bearbeitung der Anfrage. Für jede atomare Teilanfrage beziehungsweise jedes Feature wird eine Ähnlichkeitsberechnung zwischen Anfrageobjekt und den Datenbankobjekten durchgeführt. Für diesen Zweck werden Distanzfunktionen, beispielsweise die Euklidische Distanz oder die Mahalanobis-Distanz, eingesetzt. Da diese Funktionen die Distanz und nicht die Ähnlichkeit zwischen Objekten ermitteln, ist eine Überführung des Distanzwertes in einen Ähnlichkeitswert für jedes Objekt und jedes betrachtete Feature notwendig. Je höher der Grad der Ähnlichkeit ist, desto höher ist die Ähnlichkeit zwischen Anfrage und Datenbankobjekt und somit steigt auch der Grad der Relevanz des Datenbankobjektes für den Anwender. Der Grad der Ähnlichkeit entspricht dem Relevanzwert. Im Anschluss wird für jedes Objekt der Gesamtähnlichkeitsgrad unter Verwendung geeigneter Algorithmen und Scoring-Funktionen bestimmt.

Die Evaluierung von scharfen, zeitlichen und räumlichen Anfragen erfolgt unter Einbeziehung der Meta- und Strukturdaten, der relationalen Daten sowie den Konzepten. Eine Kombination von scharfen, zeitlichen und räumlichen Bedingungen mit Ähnlichkeitsbedingungen in einer Anfrage ist möglich. Hierfür werden für jedes Datenbankobjekt die RSVs der atomaren Teilanfragen zu einem Gesamtrelevanzwert mittels einer Scoring-Funktion zusammengefasst.

5. **Ergebnispräsentation:** Nachdem eine Ergebnisliste, bestehend jeweils aus Datenbankobjekt und dazu gehörigem RSV erzeugt wurde, werden die Ergebnisse dem Anwender präsentiert. Dafür werden die Ergebnisse entsprechend den vom Nutzer gegeben Vorgaben aufbereitet. Beispielsweise sind Formatumwandlungen oder Medienumsetzungen notwendig. Die relevanten Objekte werden entsprechend ihres Relevanzwertes in sortierter Reihenfolge präsentiert. Besondere Visualisierungstechniken zur Darstellung multidimensionaler Daten, wie beispielsweise der *Parallel Coordinate Plot* (Inselberg und Dimsdale, 1990), der *Star Plot* (Chambers u. a., 1983) oder der Einsatz von *Self-Organizing Maps* in dem Ansatz von Nürnberger und Klose (2002), sollen den Anwender bei der Interpretation des Retrieval-Ergebnisses unterstützen. Für das inhaltsbasierte Bild-Retrieval existieren bereits eine Vielzahl von zwei- und dreidimensionalen Visualisierungssystemen (Nakazato und Huang, 2001; Cinque u. a., 1998; Santini und Jain, 1999; Chen u. a., 2000; Massari u. a., 1997; Rubner u. a., 2000; Tian und Taylor, 2000).

   Oftmals erfolgt eine Kombination von multidimensionaler Visualisierung und Browsing, wie beispielsweise im *ImageVibe* von Cinque u. a. (1998). Hier werden in einem zweidimensionalen Raum eine Anzahl interessanter nutzerdefinierter Punkte (POI – Points of Interest) entsprechend des Farb- oder Formattributes dargestellt. Bilder werden in Abhängigkeit ihrer Ähnlichkeit zu den definierten Punkten ebenfalls als Punkte abgebildet (siehe Abbildung 2.2a). Das *3D-Mars* System erlaubt ein Bild-Retrieval in einer Virtual-Reality-Umgebung (Nakazato und Huang, 2001). Die Retrieval-Ergebnisse werden im dreidimensionalen Raum präsentiert, wo diese interaktiv vom Nutzer bewertet werden können. Im *Sphere-Modell* werden die Bilder als Kugeln dargestellt, wodurch Cluster ähnlicher Bilder leicht zu erkennen sind (siehe Abbildung 2.2b).

   Für eine Verfeinerung beziehungsweise Modifizierung der Anfrage werden Relevanz-Feedback-Verfahren in MIRS integriert. Diese Verfahren unterstützen eine iterative An-



a) b)

Abbildung 2.2: a) *ImageVibe*: Darstellung der Bilder und POIs als Punkte im zweidimensionalen Raum (Cinque u. a., 1998) b) *3D-Mars*: Visualisierung der Anfrageergebnisse im dreidimensionalen Raum (links), die rechte Abbildung zeigt die Darstellung der Bilder als Kugeln im *Sphere-Modell* (Nakazato und Huang, 2001)

frageverfeinerung beziehungsweise -modifikation, indem eine neue Anfrage auf Basis einer Relevanzbewertung der Ergebnisobjekte durch den Nutzer generiert wird. Die Umsetzung einer Relevanzbewertung beim konzeptbasierten Retrieval kann beispielsweise auf Basis der Konzepte erfolgen. Lieferte etwa die ursprüngliche Bildanfrage bezüglich einer Suche nach Bildern, die dem Konzept *Flugzeug* zugeordnet sind, zu viele Ergebnisse, kann sie durch eine positive Nutzerbewertung der Bilder, die Passagierflugzeuge darstellen, und damit durch die Einschränkung auf das Konzept *Passagierflugzeug*, restriktiver formuliert werden. Die verfeinerte Anfrage wird erneut ausgewertet und die besten Ergebnisobjekte werden dem Nutzer präsentiert. Dieser kann den Verfeinerungszyklus so lange wiederholen, bis das Ergebnis zufriedenstellend ist oder bis keine neuen Objekte mehr gewonnen werden.

2.4 Einbindung von Nutzerpräferenzen

Bereits bei der Formulierung von Anfragen ist es möglich, Nutzerpräferenzen zu spezifizieren. Dabei werden der *qualitative* und der *quantitative* Ansatz voneinander unterschieden (Chomicki, 2002). Beim qualitativen Ansatz werden die Präferenzen zwischen den Tupeln der Ergebnismenge direkt angegeben (Lacroix und Lavency, 1987; Govindarajan u. a., 2000; Borzsonyi u. a., 2001; Chomicki, 2002; Torlone und Ciaccia, 2002b; Kießling, 2002). Dafür können binäre Präferenzrelation verwendet werden. Diese Art von Präferenzen werden typischerweise bei Präferenzanfragen verwendet. Hingegen sind die Präferenzen beim quantitativen Ansatz, wie bei der Relevanzgewichtung von Anfragetermen, indirekt durch Verwendung von Scoring-Funktionen für die gesamte Relation gegeben (Waller und Kraft, 1979; Bookstein, 1980; Salton u. a., 1983b; Yager, 1987; Dubois und Prade, 1986; Fagin und Wimmers, 1997; Sung, 1998; Pasi, 1999; Herrera-Viedma, 2001b).

Durch Einsatz von Präferenzanfragen oder durch Relevanzgewichte in einer Anfrage kann

der Nutzer ausdrücken, welcher Anfrageterm von größerer Bedeutung innerhalb der Anfrage ist. So kann die Anfrage in Beispiel 2.1 sowohl als Präferenzanfrage als auch als relevanzgewichtete Anfrage formuliert werden. Für eine Präferenzanfrage kann beispielsweise die Präferenzrelation $(Farbe \sim rot) \succ (Farbe \sim blau)$ angegeben werden, während bei einer Relevanzgewichtung jedem Anfrageterm ein numerisches Relevanzgewicht, wie beispielsweise $(Farbe \sim rot)_{0.7} \vee (Farbe \sim blau)_{0.3}$, zugeordnet wird.

Beispiel 2.1 *Gesucht werden alle Flugzeuge, wobei ein rotes Flugzeug gegenüber einem blauen Flugzeug bevorzugt wird.*

Wie dieses kleine Beispiel zeigt, gibt es einen Zusammenhang zwischen relevanzgewichteten Anfragen und Präferenzanfragen. Jede relevanzgewichtete Anfrage kann durch eine Präferenzanfrage ausgedrückt werden, sofern die Präferenzrelation als entsprechende Scoring-Funktion definiert wird. Ferner können auch Präferenzrelationen, die eine schwache Ordnung beschreiben, durch eine Scoring-Funktion ausgedrückt werden (Fishburn, 1999). Es wird von Chomicki (2002) argumentiert, dass es Fälle gibt, in denen einige Eigenschaften einer solchen Ordnung, wie Transitivität oder Irreflexivität, verletzt werden. In diesen Fällen können die Präferenzanfragen nicht durch entsprechende Scoring-Funktionen formuliert werden.

Der Vorteil des quantitativen Ansatzes liegt darin begründet, dass konkret angegeben werden kann, um wieviel mehr ein Anfrageterm im Verhältnis zu den anderen Termen von Bedeutung ist. Dies ist speziell für Multimedia-Anfragen relevant, da auf Grund des unscharfen Charakters dieser Anfragen mit Relevanzwerten gearbeitet wird. Bei Präferenzanfragen hingegen steht die Beschreibung einer einfachen, unbewerteten Präferenz zwischen zwei Objekten beziehungsweise Tupeln im Vordergrund. Es wird nicht angegeben, wie stark ein Objekt oder Tupel bevorzugt wird. Im Bereich des Multimedia-Retrievals hat sich in den vergangenen Jahren der quantitative Ansatz durchgesetzt (Fagin, 1998; Fagin und Wimmers, 2000).

Neben der Unterscheidung zwischen Präferenzanfragen und relevanzgewichteten Anfragen wird zwischen weiteren Arten der Gewichtung differenziert. So gibt es ferner Quantitäts-, Schwellwert und Operatorgewichte. Beispielsweise kann durch die Gewichtung eines Operators sein Verhalten beeinflusst werden. Eine strenge Konjunktion kann dadurch aufgeweicht werden. Damit wird die Anfrage weniger restriktiv und die Anzahl potenzieller Ergebnisobjekte steigt. Ebenso wie eine direkte Quantitätsgewichtung beeinflusst auch die Schwellwertgewichtung die Größe der Ergebnismenge.

Eine weitere Möglichkeit zur Berücksichtigung von Nutzerpräferenzen besteht in dem Einsatz von Relevanz-Feedback-Verfahren (Rocchio, 1971; Salton und Buckley, 1990; Rui u. a., 1998a). Durch eine Bewertung der Ergebnisobjekte kann der Nutzer seine Präferenzen bezüglich der gewünschten Zielobjekte beschreiben. Auf Grund dieser Feedback-Informationen kann eine Anfrage iterativ verfeinert beziehungsweise modifiziert werden. Eine erweiterte Form des Relevanz-Feedbacks stellt das Kontext-Feedback dar (Santini und Jain, 2000). Auch der Einsatz von Nutzerprofilen, welche die Präferenzen eines Nutzers verwalten, ist in diesem Zusammenhang von Bedeutung.

In den folgenden Abschnitten wird auf diese vielfältigen Möglichkeiten zur Einbindung von Nutzerpräferenzen in den Retrieval-Prozess näher eingegangen.

2.4.1 Präferenzanfragen

Dieser qualitative Ansatz zur Spezifikation von Nutzerpräferenzen zeichnet sich dadurch aus, dass die Präferenzen zwischen den Tupeln der Ergebnismenge direkt gegeben sind. Die Angabe der Präferenzen kann dabei tupel- oder attributweise erfolgen. Typischerweise werden für die Spezifikation der Präferenzen binäre Präferenzrelationen verwendet (Chomicki, 2002). Die Idee für die Entwicklung von Präferenzanfragen entsprang der Suche nach einer Möglichkeit, die Probleme klassischer Datenbankanfragen zu lösen. Dazu zählt das *Empty-Result-Problem* sowie der *Flooding Effekt* (Lacroix und Lavency, 1987; Kießling, 2002). Präferenzanfragen erlauben es, Anfragen weniger restriktiv zu gestalten, falls es keine Objekte gibt, die alle spezifizierten Bedingungen erfüllen. Dementsprechend können Anfragen aber auch beschränkt werden, falls die Ergebnismenge zu groß ist. Präferenzanfragen stellen somit eine Erweiterung traditioneller Boolescher Anfragen dar. Bei Booleschen Anfragen müssen stets alle Prädikate erfüllt werden. Im Gegensatz dazu bestehen Präferenzanfragen sowohl aus harten als auch aus weichen Bedingungen – den Präferenzen. Mit Präferenzanfragen kann daher auch eine einfache Art des Rankings für Boolesche Anfragen unterstützt werden.

Lacroix und Lavency (1987) waren die ersten, die sich dieser Thematik widmeten. Sie entwickelten eine Anfragesprache mit Präferenzbedingungen. Ferner haben sie gezeigt, dass ihre Präferenzanfragen in das Bereichskalkül überführt werden können. Die verwendeten Präferenzrelationen beschreiben strenge Halbordnungen. Bei einer Präferenzanfrage wird zunächst die Anfrage ohne die Präferenzbedingung evaluiert. Auf die ermittelte Ergebnismenge wird dann die Präferenzbedingung angewendet. Wird als Ergebnis eine leere Menge geliefert, so wird die Präferenzbedingung nichtig, ansonsten wird die Kardinalität der Ergebnismenge verringert. Auf diesen Vorschlag aufbauend, entwickelten Govindarajan u. a. (2000) Präferenzanfragen für deduktive Datenbanken.

Ferner kann in dem Vorschlag von Lacroix und Lavency (1987) eine relative Gewichtung der Präferenzen durch Verschachtelung der Präferenzbedingungen erfolgen. So werden Präferenzen, die in der Hierarchie am weitesten unten stehen, gegenüber solchen, die weiter oben stehen, als weniger wichtig erachtet. Um eine solche Präferenzhierachie aufzubauen, steht in der vorgeschlagenen Anfragesprache das Schlüsselwort *from which* bereit. Ebenso ist eine Verkettung von verschiedenen Präferenzen möglich, welche dann alle die gleiche relative Wichtigkeit besitzen. Das folgende Beispiel zeigt eine Präferenzanfrage entsprechend der von Lacroix und Lavency vorgegebenen Syntax.

Beispiel 2.2 *Es soll der Typ der Flugzeuge ermittelt werden, die die Bedingung Q erfüllen. Dabei sollen solche Typen bevorzugt werden, die auch P1 erfüllen, wobei hier wiederum solche bevorzugt werden, die auch P2 erfüllen. Die Präferenzanfrage kann wie folgt angegeben werden:*

```
select the Typ of Flugzeug
 having Q
from which
 prefer those having P1
from which
  prefer those having P2
```

Das heißt also, dass zunächst die Flugzeugtypen ermittelt werden, die $Q \wedge P1 \wedge P2$ erfüllen. Ist diese

Ergebnismenge leer, dann werden die Typen ermittelt, die $Q \wedge P1$ erfüllen. Ist auch diese Ergebnismenge leer, so werden letztlich nur diejenigen Typen bestimmt, die Q erfüllen.

Mit diesem Ansatz kann bereits eine einfache Form des Rankings unterstützt werden. So ist es möglich, zunächst all die Tupel zurückzuliefern, die sowohl die harten Bedingungen als auch die Präferenzen erfüllen ($Q \wedge P1 \wedge P2$), gefolgt von den Tupeln, die die harten und nur die wichtigsten Präferenzen erfüllen ($Q \wedge P1$), gefolgt von denen, die letztendlich nur die harten Bedingungen erfüllen (Q).

Einen ähnlichen Ansatz verfolgt Chomicki (2002) mit dem *winnow*-Operator $w_C(R)$. Dieser Operator erweitert die relationale Algebra und erlaubt das Selektieren von Tupeln einer Relation R, die entsprechend einer gegebenen Präferenzformel C am stärksten bevorzugt werden. Eine Präferenzformel C setzt eine entsprechende Präferenzrelation $\succ$ um. Die Präferenzrelation erlaubt die Formulierung von Präferenzen sowohl bezüglich der Attribute als auch zwischen den einzelnen Tupeln. Das folgende Beispiel 2.3 zeigt die Formulierung einer Präferenzanfrage mit Hilfe des *winnow*-Operators.

Beispiel 2.3 *Betrachtet wird die Relation Flugzeug(Typ, Hersteller, Preis). Die Präferenzrelation $\succ_1$ wird wie folgt formuliert:*

> *Sind zwei Flugzeuge vom gleichen Typ, so bevorzuge das Flugzeug mit dem kleineren Preis.*

und ist durch die Formel C_1:

$$(t, h, p) \succ_1 (t', h', p') \Leftrightarrow t = t' \wedge p < p'$$

definiert. Als Ergebnis der Präferenzanfrage $w_{C_1}(Flugzeug)$ werden sodann der Typ, der Hersteller und der Preis derjenigen Hersteller ermittelt, die den kleinsten Preis für einen Flugzeugtyp anbieten.

Der *winnow*-Operator kann iterativ angewendet werden, so dass ein Ranking der Ergebnisse möglich ist. So würde die Anfrage $w^2_{C_1}(Flugzeug)$ die zweitbesten Tupel ermitteln. Für Flugzeuge vom gleichen Typ werden dann die Flugzeugdaten der Hersteller ermittelt, die den zweitbesten Preis anbieten. Der in dem Ansatz von Torlone und Ciaccia (2002b) vorgestellte *best*-Operator entspricht dem *winnow*-Operator.

Chomicki hat gezeigt, dass die Präferenzanfragen aus dem Vorschlag von Lacroix und Lavency auch mittels des *winnow*-Operators umgesetzt werden können. Während in dem Ansatz von Chomicki Präferenzen bezüglich eines Minimum- beziehungsweise Maximumkriteriums implizit im *winnow*-Operator angegeben werden (siehe Beispiel 2.3), ist bei Lacroix und Lavency der explizite Einsatz einer Aggregationsfunktion notwendig. Darüber hinaus ist der Ansatz von Chomicki mächtiger, da auch iterative Präferenzanfragen unterstützt werden. Dies ist im Vorschlag von Lacroix und Lavency nicht möglich.

In dem Ansatz von Kießling (2002) liegt den Präferenzanfragen eine strenge Halbordnung als Semantik zu Grunde. Dabei werden verschiedene Typen von Präferenzen definiert und gezeigt, wie diese in einer Präferenzalgebra und in SQL abgebildet werden können (Kießling, 2002; Kießling und Köstler, 2002).

Einen Spezialfall der Ergebnisse einer Präferenzanfrage stellt die sogenannte *Skyline* dar. Als Skyline werden die Punkte, die von keinen anderen Punkten dominiert werden verstanden (Borzsonyi u. a., 2001). Dabei dominiert ein Punkt einen anderen Punkt, wenn er in allen

Dimensionen mindestens genauso gut und in mindestens einer Dimension besser ist. Für diesen speziellen Zweck schlagen Borzsonyi u. a. (2001) den Einsatz des *skyline*-Operators vor.

2.4.2 Gewichtete Anfragen

Der Einsatz von Gewichten in Anfragen hat sich in den vergangenen 20 Jahren als adäquates Mittel zur Beschreibung der Nutzerpräferenzen in Retrieval-Systemen bewährt. Dabei können Gewichte unterschiedliche Semantiken abbilden. Neben der Verwendung von Gewichten zur Festlegung von Schwellwerten (Radecki, 1979; Buell und Kraft, 1981; Herrera-Viedma, 2001b) können Gewichte in Anfragen mit einer quantitativen Semantik (Herrera-Viedma, 2001a,c) assoziiert werden. Ferner besteht die Möglichkeit, die Semantik der klassischen Booleschen Junktoren *UND* und *ODER* zu modifizieren, so dass die Gewichte eine Kontrollsemantik aufweisen (Salton u. a., 1983a; Waller und Kraft, 1979). Am weitesten verbreitet ist jedoch die Relevanzgewichtung von Termen. Durch die Vergabe von Relevanzgewichten für Anfrageterme kann ausgedrückt werden, wie stark der Einfluss eines einzelnen Anfrageterms innerhalb der Anfrage ist (Waller und Kraft, 1979; Bookstein, 1980; Salton u. a., 1983b; Fagin und Wimmers, 1997; Sung, 1998; Yager, 1987; Dubois und Prade, 1986). Zusammenfassend werden die folgenden Gewichtungen unterschieden, die im Folgenden kurz vorgestellt werden:

- die Relevanzgewichtung,
- die Schwellwertgewichtung,
- die Quantitätsgewichtung sowie
- die Gewichtung der Operatoren.

Typischerweise werden die Gewichte durch numerische Werte spezifiziert. Anstelle dieser können jedoch auch linguistische Variablen, wie beispielsweise *Wichtigkeit* mit den Werten *sehr wichtig*, *wichtig*, *unwichtig* et cetera, genutzt werden (Bordogna u. a., 1995; Herrera-Viedma, 2001b,a,c). Diese weisen, wie in Abbildung 2.3 dargestellt, eine Fuzzy-Semantik auf und können daher gut von den Anwendern verstanden werden. Allerdings erhöht sich die Komplexität bei der Anfragebearbeitung, da zusätzlich die linguistischen Variablen interpretiert und ausgewertet werden müssen.

Relevanzgewichtung

In diesem quantitativem Ansatz werden die Präferenzen des Nutzers durch eine Gewichtung der Anfrageterme entsprechend ihrer Relevanz innerhalb der Anfrage, wie im Beispiel 2.4 verdeutlicht, ausgedrückt (Waller und Kraft, 1979; Bookstein, 1980; Salton u. a., 1983b; Fagin und Wimmers, 1997; Sung, 1998; Yager, 1987; Dubois und Prade, 1986). Je höher das Gewicht für einen Term ist, desto einflussreicher ist dieser in der Anfrage. Ein Gewicht von *null* bedeutet, dass dieser Term vollkommen unwichtig innerhalb der Anfrage ist, während ein mit *eins* gewichteter Term absolut notwendig ist. Dabei ist zu berücksichtigen, dass ein Term relativ zu den anderen Termen der Anfrage gewichtet wird. Aus Nutzersicht wird folglich das perfekte Objekt beschrieben.

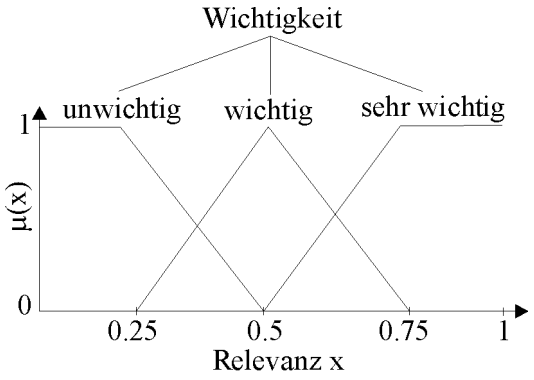


Abbildung 2.3: Mögliche Beschreibung der linguistischen Variablen *Wichtigkeit*

Beispiel 2.4 *Suche Abbildungen von Flugzeugen, die der vorgegebenen Abbildung ähnlich sind. Dabei ist die Ähnlichkeit bezüglich der Form doppelt so wichtig wie die Ähnlichkeit bezüglich der Farbe.*

Die Präferenzen in Form von Gewichten aus dem Wertebereich $[0,1]$ fließen durch den Einsatz einer gewichteten Scoring-Funktion $S^{\Theta} : [0,1]^n \times [0,1]^n \rightarrow [0,1]$ mit in die Anfragebearbeitung ein. Bei Verwendung einer Scoring-Funktion werden die Gewichte indirekt durch den berechneten Relevanzwert abgebildet. Dabei ordnet die Scoring-Funktion jedem Ergebnistupel einen numerischen Wert zu. Ein Tupel t_1 wird dementsprechend vor einem Tupel t_2 bevorzugt, wenn der ermittelte RSV des Tupels t_1 größer ist als der RSV des Tupels t_2. Im Zusammenhang mit der Untersuchung geeigneter Scoring-Funktionen wird in Kapitel 4 näher auf die Relevanzgewichtung eingegangen.

Schwellwertgewichtung

Im Gegensatz zu den Relevanzgewichten bestehen bei Schwellwertgewichten keine Abhängigkeiten untereinander. Dadurch stellen sie ein absolutes Kriterium dar. Herrera-Viedma (2001b) unterscheidet zwei Semantiken bei der Schwellwertgewichtung:

Präsenzgewichte; Es werden die Objekte belohnt, deren RSV für einen Anfrageterm über dem spezifizierten Schwellwertgewicht liegen.

Absenzgewichte; Es werden die Objekte belohnt, deren RSV für einen Anfrageterm unter dem spezifizierten Schwellwertgewicht liegen.

Im Allgemeinen wird ein Schwellwertgewicht als Präsenzkriterium verstanden. Es ist somit ein Kriterium, welches ein Objekt erfüllen muss, um vom Nutzer akzeptiert zu werden (Buell und Kraft, 1981). Durch Angabe von Schwellwertgewichten beschreibt der Nutzer demnach das minimal akzeptable Objekt und verlangt nach all den Objekten, die diesen Anforderungen genügen. Dies ist im Beispiel 2.5 dargestellt. Ein Schwellwertgewicht von *null* ergibt wenig Sinn, da der Nutzer verlangt, dass tatsächlich alle Dokumente zurückgeliefert werden.

Beispiel 2.5 *Suche Abbildungen von Flugzeugen, die der gegebenen Abbildung mindestens zu einem Grad von 0.7 bezüglich der Form und mindestens zu einem Grad von 0.4 bezüglich der Farbe ähnlich sind.*

Analog zu den Relevanzgewichten fließen auch Schwellwertgewichte in eine Scoring-Funktion ein und beeinflussen so indirekt das Anfrageergebnis. Somit wird auch bei der Schwellwertgewichtung ein Tupel t_1 vor einem Tupel t_2 bevorzugt, wenn der ermittelte RSV des Tupels t_1 größer ist, als der RSV des Tupels t_2.

Eine einfache Interpretation der Schwellwertgewichte durch eine Scoring-Funktion wäre eine *Gewinner-bekommt-alles* Strategie. Das würde bei Einsatz von Präsenzgewichten bedeuten, dass einem Objekt, dessen RSV unter dem Schwellwert liegt, der Wert *null* zugeteilt wird, ansonsten der Wert *eins*. In Abbildung 2.4a ist diese Strategie dargestellt.

Diese Herangehensweise ist jedoch zu drastisch, da nicht berücksichtigt wird, in welchem Umfang der Schwellwert über- beziehungsweise unterschritten wird. Eine Möglichkeit dieses Problem zu lösen, besteht in dem Einsatz einer stetigen Scoring-Funktion (Buell und Kraft, 1981; Herrera-Viedma, 2001b). Dies ist beispielhaft in Abbildung 2.4b dargestellt. Demzufolge muss bei einer Überschreitung beziehungsweise Unterschreitung des spezifizierten Schwellwertgewichtes der Grad der Über- beziehungsweise Unterschreitung durch die Scoring-Funktion ausgedrückt werden. Buell und Kraft (1981) schlagen die folgende Scoring-Funktion $e(\mu, a) : [0,1] \times [0,1] \rightarrow [0,1]$ mit dem Schwellwertgewicht a und dem RSV μ vor:

$$e(\mu, a) = \begin{cases} P(a) * (\mu / a) & \text{wenn } \mu < a \\ P(a) + Q(a) * (\mu - a)/(1 - a) & \text{wenn } \mu > a \end{cases}$$

wobei $P(a)$ eine Normalisierungsfunktion in Abhängigkeit von a und $Q(a)$ eine nur von a abhängige Funktion ist. Gilt $\mu = a$, so sind beide Funktionen identisch. Durch diese Funktion kann berücksichtigt werden, in welchem Ausmaß der Grenzwert überschritten wird beziehungsweise bis zu welchem Grad der Relevanzwert an ihn heranreicht. Dadurch wird dem Grenzwertansatz die strenge Absolutheit genommen (Fühles-Ubach, 1997).

Während in den Ansätzen von Buell und Kraft (1981) und Herrera-Viedma (2001b) jeder Anfrageterm mit einem Schwellwertgewicht versehen werden kann, gilt bei Radecki (1979) das Schwellwertgewicht gleichzeitig für alle Terme einer Anfrage. Der Schwellwert wird in diesem Ansatz als *Lambda-Level* bezeichnet. Durch den Einsatz des Lambda-Levels wird das Suchergebnis auf solche Objekte begrenzt, deren Relevanzwerte bezüglich aller einzelnen Anfrageterme mindestens den geforderten Schwellwert erfüllen. Damit werden die beiden vorgestellten Strategien zur Auswertung von Schwellwertgewichten kombiniert (vgl. Abbildung 2.4c).

Die Berechnung des Relevanzwertes unter Beachtung der Schwellwertgewichte für die einzelnen Anfrageterme erfolgt bei Radecki (1979) in zwei Schritten:

1. Es wird für jedes Objekt und jede Teilanfrage der RSV bezüglich eines spezifizierten Schwellwertes berechnet.

2. Es wird der Gesamtrelevanzwert entsprechend des Booleschen Ausdrucks in der Anfrage gebildet.

Zenner u. a. (1985) entwickelten den Ansatz von Radecki weiter, so dass erst im letzten Schritt der Schwellwert verwendet. Zunächst wird der Gesamtrelevanzwert der Objekte be-

rechnet. Im Anschluss daran werden die Objekte, die unterhalb des spezifizierten Schwellwertes liegen, vom Ergebnis entfernt.

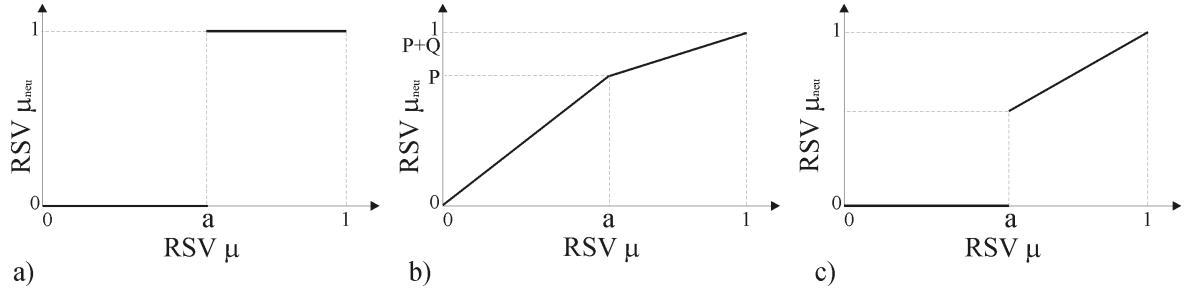


Abbildung 2.4: Alternativen zur Abbildung von Schwellwertgewichten: a) *alles-oder-nichts* Strategie b) kontinuierliche Funktion von Buell und Kraft (1981) c) Kombination dieser beiden Strategien

Im Gegensatz zu den anderen Ansätzen zeichnen sich die Schwellwertmodelle von Zenner u. a. (1985) und Herrera-Viedma (2001b) dadurch aus, dass Schwellwerte nicht nur als numerische Werte, sondern auch als linguistische Variablen definiert werden. Auf diese Weise wird eine Unabhängigkeit gegenüber einem festen Grenzwert erreicht (Fühles-Ubach, 1997).

Quantitätsgewichtung

Bei der Quantitätsgewichtung einer Anfrage oder einzelner Anfrageterme wird ein Kriterium formuliert, welches die Quantität der zu suchenden Objekte beeinflusst. Dieses ist in der Anfrage in Beispiel 2.6 dargestellt. Eine quantitative Gewichtungssemantik impliziert die Beschränkung der Anzahl der Objekte in der Ergebnismenge. Bei einer Quantitätsgewichtung besteht der Wunsch des Nutzers darin, die minimale Anzahl der besten Objekte, die den qualitativen Kriterien des Nutzers genügen, als Ergebnis zu erhalten (Herrera-Viedma, 2001c).

Beispiel 2.6 *Suche* einige *Abbildungen von Flugzeugen, die der gegebenen Abbildung ähnlich sind.*

Insbesondere bei Verwendung von Quantitätsgewichten bietet sich der Einsatz einer linguistischen Variablen *Quantität* an. Dieser Variable können beispielsweise die unscharfen Werte *wenige*, *viele*, *einige* zugeordnet werden.

Operatorengewichtung

Durch eine Gewichtung der Operatoren kann das Verhalten dieser parametrisiert werden. Das ist von Bedeutung, wenn es nicht um einen strenge Konjunktion beziehungsweise Disjunktion geht, sondern um Forderungen, wie in Beispiel 2.7 dargestellt.

Beispiel 2.7 *Suche Abbildungen von Flugzeugen, die der gegebenen Abbildung bezüglich Farbe, Schriftzug, Form sowie Textur ähnlich sind. Dabei sollen* möglichst viele *der geforderten Eigenschaften, also eine Ähnlichkeit bezüglich Farbe, Schriftzug, Form und Textur, erfüllt werden.*

Wird die Anfrage in Beispiel 2.7 mit einer Konjunktion oder Disjunktion der einzelnen Anfragebedingungen umgesetzt, so hängt der Gesamtrelevanzwert für ein Objekt jeweils nur von einem einzigen Wert ab. Dieses Verhalten ist zu drastisch und damit nicht wünschenswert, da verlangt wird, dass *möglichst viele* – aber nicht zwingend alle – Suchbedingungen erfüllt werden.

Durch ein Gewicht kann das Verhalten des Operator beeinflusst werden. Damit liegt das Verhalten des Operators in Abhängigkeit vom Gewicht zwischen *UND* und *ODER*. Verhält sich der Operator eher wie ein *ODER*, so ist anzunehmen, dass die Ergebnismenge größer wird, als bei einem restriktiven Operator, der sich nahe dem *UND* verhält.

Aus der Fuzzy Logik stammen die *kompensatorischen Operatoren*, die eine Verknüpfung von mehreren Fuzzy-Mengen ermöglichen. Das Essentielle an den kompensatorischen Operatoren ist, dass fehlende Eigenschaften durch andere Eigenschaften, die besonders gut erfüllt werden, kompensiert werden (Biewer, 1997, S. 87). Dadurch ist nicht die Erfüllung jeder Eigenschaft, sondern lediglich der Gesamteindruck von Bedeutung. Um einen Kompensationseffekt der Relevanzwerte $\mu_1(o)$ und $\mu_2(o)$ für ein Objekt o unter Beachtung der Suchbedingungen 1 und 2 zu erzielen, werden ein Durchschnitts- (f_D) und ein Vereinigungsoperator (f_V), typischerweise T-Normen und T-Conormen, miteinander kombiniert. Dies kann in Form einer Linearkombination (vgl. Formel 2.1) oder einer Exponentialverknüpfung (vgl. Formel 2.2) erfolgen (Zimmermann und Zysno, 1980, S. 47ff).

$$f_{komp_l}(\mu_1(o), \mu_2(o)) = \gamma * f_V(\mu_1(o), \mu_2(o)) + (1-\gamma) * f_D(\mu_1(o), \mu_2(o)) \tag{2.1}$$

$$f_{komp_e}(\mu_1(o), \mu_2(o)) = (f_V(\mu_1(o), \mu_2(x)))^{\gamma} * (f_D(\mu_1(o), \mu_2(o)))^{1-\gamma} \tag{2.2}$$

Bei den kompensatorischen Operatoren kann durch den Parameter $\gamma \in [0,1]$ festgelegt werden, wo zwischen *UND* und *ODER* der Operator liegt. Der bekannteste Operator ist der γ-Operator. Er kombiniert das algebraische Produkt und die algebraischen Summe in Form einer Exponentialfunktion, so dass er bei $\gamma = 0$ ein *UND*-Verhalten und bei $\gamma = 1$ ein *ODER*-Verhalten aufweist. Daher entspricht dieser Operator einem *kompensatorischen UND*. Er ist wie folgt definiert (Zimmermann und Zysno, 1980, 1983):

$$f_{\gamma}(\mu_1(o), \mu_2(o)) = (\mu_1(o) * \mu_2(o))^{1-\gamma} * (\mu_1(o) + \mu_2(o) - \mu_1(o) * \mu_2(o))^{\gamma}.$$

Eine andere Möglichkeit stellt der λ-Operator dar. Mittels des Parameters $\lambda \in [0,1]$ wird bestimmt, wo zwischen *UND* ($\lambda = 1$) und *ODER* ($\lambda = 0$) der Operator liegt. Damit entspricht der λ-Operator einem *kompensatorischen ODER* und ergibt sich aus der Kombination des algebraischen Produkts (T-Norm) und der algebraischen Summe (T-Conorm):

$$f_{\lambda}(\mu_1(o), \mu_2(o)) = \lambda * (\mu_1(o) * \mu_2(o)) + (1-\lambda) * (\mu_1(o) + \mu_2(o) - \mu_1(o) * \mu_2(o)).$$

Der *ANDOR*-Operator, von Waller und Kraft (1979) eingeführt, setzt sich aus der Kombination von *UND* und *ODER* zusammen und kann durch ein Gewicht $z \in [0,1]$ parametrisiert werden:

$$f_{ANDOR}(\mu_1(o), \mu_2(o)) = z * min(\mu_1(o), \mu_2(o)) + (1-z) * max(\mu_1(o), \mu_2(o)).$$

Der Operator wird demnach als Linearkombination aus dem Minimum und dem Maximum der einzelnen Relevanzwerte, sowie unter Beachtung des z-Wertes berechnet. Wird z ein Wert von 1 zugewiesen, so agiert der Operator als *UND*. Dementsprechend führt ein z-Wert von 0 zu einem *ODER*-Verhalten des Operators. Damit fällt auch der *ANDOR*-Operator in die Klasse der kompensatorischen Operatoren. Er entspricht dem λ-Operator, wobei als T-Norm anstelle des algebraischen Produktes die Funktion *min* und als T-Conorm anstelle der algebraischen Summe die Funktion *max* verwendet wird. Ferner ist anzumerken, dass bei einem z-Wert von 0.5 der *ANDOR*-Operator dem arithmetischen Mittel entspricht.

Neben diesen kompensatorischen Operatoren können auch *Ordered-Weighted-Averaging-Operatoren* (*OWA*-Operatoren) eingesetzt werden. Sie wurden von Yager eingeführt (Yager, 1988, 1991, 1993). Auch das Verhalten der *OWA*-Operatoren liegt zwischen *UND* und *ODER*:

$$min(\mu_1(o), \ldots, \mu_n(o)) \leq OWA(\mu_1(o), \ldots, \mu_n(o)) \leq max(\mu_1(o), \ldots, \mu_n(o)).$$

Anders als bei den reinen kompensatorischen Operatoren erlauben die *OWA*-Operatoren eine Gewichtungen der Suchbedingungen. Um dennoch einen Kompensationseffekt zu erzielen, werden die Gewichte $w_i \in [0,1]$ mit $i = 1, \ldots, n$ nicht fest einer Suchbedingung zugeordnet. Der *OWA*-Operator ist somit wie folgt definiert:

$$OWA_{w_1,\ldots,w_n}(\mu_1(o), \ldots, \mu_n(o)) = \sum_{i=1}^{n} w_i \mu_{(i)}(o)$$

mit $\sum_{i=1}^{n} w_i = 1$ und einer aufsteigenden Sortierung der Scoring-Werte $\mu_{(1)}(o) \leq \mu_{(2)}(o) \leq \ldots \leq \mu_{(n)}(o)$. Bei einer Gewichtung der Suchbedingungen mit $w_i = 1/n$ ergibt sich das arithmetische Mittel, während eine Gewichtung $w_* = (1, 0, \ldots, 0)$ zum *min*-Operator und dementsprechend eine Gewichtung $w^* = (0, \ldots, 0, 1)$ zum *max*-Operator führt. Die Abbildung 2.5 zeigt, wie aus jedem kompensatorischen Operator und dem *OWA*-Operator durch entsprechende Parametrisierung jeweils der reine *UND*- beziehungsweise *ODER*-Operator rekonstruiert werden kann.

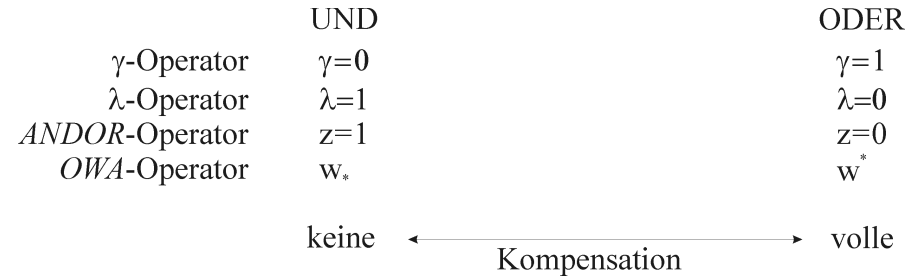


Abbildung 2.5: Wirkung der kompensatorischen Operatoren und des *OWA*-Operators

Eine weitere Möglichkeit für die Gewichtung von Operatoren stellt das von Salton u. a. (1983b) vorgeschlagene *p-norm Modell* dar. Über den Parameter p mit $1 \leq p \leq \infty$ kann die Striktheit der Booleschen Operatoren *UND* und *ODER* aufgeweicht und die Berechnung der Ähnlichkeitsmaße beeinflusst werden. Im Gegensatz zu den kompensatorischen Operatoren steht dabei die Anfrage-Objekt-Ähnlichkeitsberechnung im Vordergrund und nicht die rein

logische Kombination von bereits ermittelten Relevanzwerten (Fuzzy-Mengen).

Sowohl die Anfrage als auch die Objekte werden in diesem Modell als n-dimensionale Vektoren aufgefasst, wobei die Dimension gleich der Anzahl der Suchbedingungen der Anfrage ist. Die Termgewichte liegen im Intervall $[0, 1]$. Bei einer Konjunktion wird verlangt, dass alle Suchbedingungen erfüllt werden. Daher werden Objekte um so besser bewertet, je näher sie am optimalen Punkt $(1, \ldots, 1)$ liegen. Analog dazu gilt, dass ein Objekt bei einer Disjunktion dann am wichtigsten ist, wenn sein Abstand vom Punkt $(0, \ldots, 0)$ am größten ist. Für die Berechnung des Abstandes liegt bei $p = 1$ das Vektorraum-Modell zu Grunde. Die Ähnlichkeit zwischen Anfrage und Objekt wird aus dem Skalarprodukt der jeweiligen Vektoren ermittelt. Damit verschwindet der Unterschied zwischen dem *UND* und *ODER*. Bei $p = \infty$ liegt der Anfrageberechnung das Fuzzy-Modell zu Grunde, womit eine klare Trennung zwischen *UND* und *ODER* stattfindet.

2.4.3 Relevanz-Feedback

Beim Begriff der Relevanz wird im Zusammenhang mit Retrieval-Systemen zwischen zwei Bereichen unterschieden (Womser-Hacker, 1989, S. 29):

Systemrelevanz: Hierbei ist die Entscheidung des Systems bezüglich der Relevanz eine Objektes zur Anfrage gemeint. Dabei ist der Grad der formalen Übereinstimmung zwischen der Anfrage und der internen Darstellung eines Objektes von Bedeutung.

Benutzerrelevanz: Sie gibt den Grad der vom Nutzer angegeben Übereinstimmung zwischen Retrieval-Ergebnis und Nutzerbedürfnis an. Der Nutzer entscheidet, ob ein Objekt bezüglich der Anfrage für ihn relevant ist oder nicht. Die Benutzerrelevanz kann von Nutzer zu Nutzer unterschiedlich sein.

Der Unterschied zwischen Benutzerrelevanz und Systemrelevanz kann als Indikator für die Effektivität des Retrieval-Systems herangezogen werden. Beim Relevanz-Feedback bezieht sich der Begriff der Relevanz auf die Übereinstimmung zwischen einer Anfrage und einem Ergebnisobjekt aus der Sicht eines Nutzers (ebd. S. 28). Damit ist im Rahmen des Relevanz-Feedbacks die Benutzerrelevanz von Bedeutung.

Relevanz-Feedback, ein einfaches Verfahren zur iterativen Verfeinerung beziehungsweise Modifizierung einer Anfrage, ist eine klassische Methode, um den Nutzer aktiv in den Retrieval-Prozess einzubinden. Der Einsatz von Relevanz-Feedback-Methoden ist sinnvoll, da die Initialanfrage oftmals nicht ausreichend genau formuliert werden kann, um das Informationsbedürfnis des Nutzers abdecken zu können. Durch die Interaktion zwischen Nutzer und Retrieval-System wird versucht, die Präferenzen des Nutzers so gut wie möglich in den Retrieval-Prozess zu integrieren. Damit soll die Effektivität des Retrieval-Systems gesteigert und das Informationsbedürfnis des Nutzers gedeckt werden.

Beim Relevanz-Feedback wird durch den Nutzer eine einfache Relevanzbewertung der Ergebnisobjekte, die durch die Initialanfrage ermittelt wurden, vorgenommen. Neben der Bewertung der Ergebnisobjekte kann ebenso eine Bewertung der Features oder der Dimension der Features stattfinden. Oftmals wird bei der Bewertung lediglich zwischen *relevant* und *nicht relevant* unterschieden. Jedoch kann auch eine detailliertere Bewertung beispielsweise auf einer Skala von 1-5 (*irrelevant* bis *sehr relevant*) vorgenommen werden. Die Beurteilung der Objekte wird dem System als Feedback zurückgegeben. Auf dieser Basis wird die

ursprüngliche Anfrage verändert. Diese kombiniert durch die Junktoren $\wedge$ und $\vee$ verschiedene Suchbedingungen, welche Objekte, Features oder auch numerische und alphanumerische Terme beinhalten können. Für die Modifizierung der ursprünglichen Anfrage kommen verschiedene Techniken zum Einsatz. Dazu zählen

- die Verschiebung des Anfragepunktes,
- die Modifizierung der Gewichte von Anfragetermen,
- die Erweiterung der Anfrage oder
- die Gewichtung der Dimension.

In den folgenden Abschnitten wird auf diese Techniken näher eingegangen.

Das Ziel der Modifikation einer Anfrage ist es, auf Basis der Nutzerbewertung die ursprüngliche Anfrage des Nutzers so weit zu verändern, dass sich die neue Anfrage der optimalen Anfrage nähert oder zumindest zwischen den vom Nutzer als relevant und irrelevant bewerteten Objekten besser trennt. Eine optimale Anfrage liefert genau die Objekte, die relevant für den Nutzer sind. Ein solches Ergebnis ist nicht immer erreichbar. Die modifizierte Anfrage sollte demnach eine hohe Ähnlichkeit mit den als relevant bewerteten Objekten, hingegen eine geringe Ähnlichkeit mit den als irrelevant bewerteten Objekten aufweisen.

Die auf Basis der modifizierten Anfrage gelieferten Ergebnisobjekte können erneut vom Nutzer bewertet werden. Der Verfeinerungszyklus kann beliebig oft wiederholt werden, bis das Ergebnis für den Nutzer zufriedenstellend ist.

Ein Vorteil des Relevanz-Feedbacks besteht darin, dass der Nutzer keine Systeminterna kennen muß, sondern lediglich das Retrieval-Ergebnis bewertet. Nachteilig ist hingegen, dass der Nutzer vom System vorgenommene Änderungen nicht nachvollziehen kann. Damit ist die Anfrage nicht mehr transparent. Aus diesem Grund kann es für den Nutzer unverständlich sein, warum ein Objekt vom System besser bewertet wurde als ein anderes Objekt.

Verschiebung des Anfragepunktes

Bei der Verschiebung des Anfragepunktes wird der neue Anfragevektor aus dem ursprünglichen Anfragevektor, den relevanten und irrelevanten Objektvektoren und einer Gewichtung erzeugt (Rocchio, 1971; Salton und Buckley, 1990). Dabei wird, wie in Abbildung 2.6 dargestellt, der ursprüngliche Anfragepunkt zu den relevanten Objekten ($\oplus$) hinbewegt und von den irrelevanten Objekten ($\ominus$) entfernt.

Sei q der Anfragevektor, R und R' die Menge der relevanten beziehungsweise irrelevanten Objekte und d und d' die Objektvektoren aus R und R'. Der neue Anfragevektor $\hat{q}$ ergibt sich nach Rocchio (1971) als:

$$\hat{q} = \alpha q + \frac{\beta}{|R|} \sum_{d \in R} d - \frac{\lambda}{|R'|} \sum_{d' \in R'} d'.$$

Die Gewichte α, β, λ sind experimentell bestimmte Konstanten.

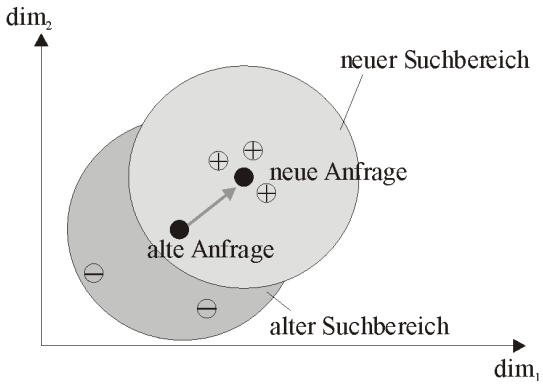


Abbildung 2.6: Verschiebung des Anfragepunktes

Modifizierung der Gewichte von Anfragetermen

Wie bereits in Abschnitt 2.4.2 beschrieben, können Gewichte für Anfrageterme, wie beispielsweise Features oder Objekte, bereits bei der Anfrageformulierung durch den Nutzer vergeben werden. Anfrageterme mit einem hohen Gewicht haben einen starken Einfluß auf die Berechnung des Relevanzwertes und somit auf die in der Ergebnisliste enthaltenen Objekte.

Auf Basis der vom Nutzer vorgenommenen Relevanzbewertung können diese Gewichte modifiziert werden. Prinzipiell erfolgt ein Abgleich zwischen den bewerteten Objekten der Ergebnisliste und den verschiedenen Feature-Ergebnislisten (Rui u. a., 1998a). Befindet sich ein als relevant bewertetes Objekt sowohl in der Ergebnisliste als auch in der Feature-Ergebnisliste, so wird das Gewicht für das entsprechende Feature erhöht. Analog dazu wird bei einem als irrelevant bewerteten Objekt, welches sowohl in der Ergebnisliste als auch in der Feature-Liste erscheint, das Gewicht für dieses Feature verringert. Schließlich werden die Gewichte normalisiert. Die Vorgehensweise für die Bestimmung der Gewichte für Objekte erfolgt analog. An die Stelle der Feature-Ergebnislisten treten nun die einzelnen Objekt-Ergebnislisten. Das Beispiel 2.8 verdeutlicht anhand Abbildung 2.7 die Modifikation von Feature-Gewichten.

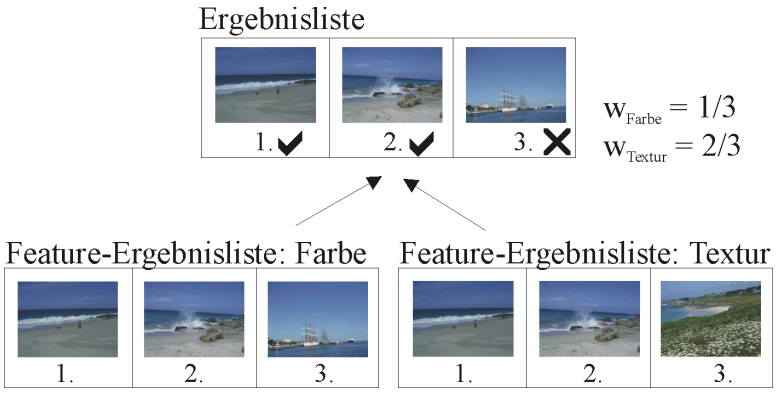


Abbildung 2.7: Gewichtung der Features (angelehnt an Bosshard (2000), S. 26)

Beispiel 2.8 *Abbildung 2.7 zeigt zwei Teilergebnislisten für die Features Farbe und Textur mit jeweils drei Objekten. In der Ergebnisliste sind die Objekte an erster und zweiter Position als relevant und das Objekt an dritter Stelle als irrelevant markiert. In der Ergebnisliste für das Feature Farbe sind davon zwei relevante und ein irrelevantes Objekt enthalten, so dass sich ein Gewicht von 1/3 ergibt. Für das Feature Textur ist das Gewicht 2/3, da die beiden relevanten Objekte in der Textur-Ergebnisliste enthalten sind.*

Für die Evaluierung gewichteter Anfragen werden, wie bereits in Abschnitt 2.4.2 beschrieben, Scoring-Funktionen eingesetzt. Diese unterstützen für jedes Objekt eine gewichtete Kombination der einzelnen Relevanzwerte aus den Feature-Ergebnislisten. Somit wird für jedes Objekt ein entsprechender Gesamtrelevanzwert ermittelt.

Erweiterung der Anfrage

Der Suchraum kann durch Hinzufügen positv bewerteter Objekte oder Features ($\oplus$), wie in Abbildung 2.8 dargestellt, oder durch Entfernen negativ bewerteter Objekte oder Features ($\ominus$) modifiziert werden.

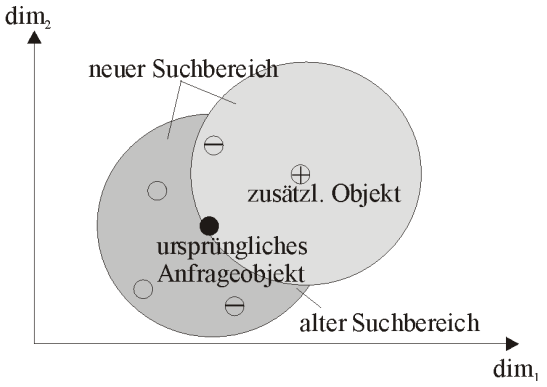


Abbildung 2.8: Erweiterung der Anfrage um ein zusätzliches Objekt

Das Entfernen von Objekten oder Features kann relativ einfach über eine Gewichtung, wie im vorhergehenden Abschnitt dargelegt, realisiert werden. Für eine Reduktion der Anfrage können Objekte oder Features mit einem Gewicht von *null* oder einem sehr kleinen Gewicht entfernt werden. Für die Erweiterung der Anfrage um Objekte werden die als relevant bewerteten Objekte mit in die Anfrage aufgenommen.

Gewichtung der Dimensionen

Durch eine Gewichtung der Dimensionen im Dokumentenraum kann der Suchbereich, welcher dadurch gestaucht beziehungsweise gestreckt wird, modifiziert werden. Aufgrund des verzerrten Suchbereichs können weitere passende Objekte gefunden werden. Die Bestimmung der Gewichtung erfolgt auf Basis von Statistiken. Hierfür wird die Standardabweichung σ der n als relevant bewerteten Objekte d von der Anfrage q bezüglich einer Dimensi-

on dim_j verwendet:

$$\sigma(dim_j) = \sqrt{\frac{\sum_{i=1}^{n}(q[dim_j] - d_i[dim_j])^2}{n}}.$$

Das Gewicht w_j für die Dimension dim_j mit $j = 1, \ldots, dim$ ergibt sich aus

$$w_j = \frac{1}{\sigma(dim_j),}$$

wobei $\sum_{j=1}^{dim} w_j = 1$ durch anschließende Normierung gilt. Es werden somit nur relevante Objekte entlang der einzelnen Dimensionen im Dokumentenraum berücksichtigt. In Abbildung 2.9 gilt $\sigma(dim_1) > \sigma(dim_2)$ bezüglich der relevant markierten Objekte ($\oplus$), so dass der Suchbereich zu einem Ellipsoid wird. Die Dimension dim_2 ist in diesem Beispiel wichtiger als dim_1, so dass dim_2 ein größeres Gewicht zugeteilt wird. Die Anfrage wird daher bezüglich der Dimension dim_1 toleranter und bezüglich Dimension dim_2 restriktiver.

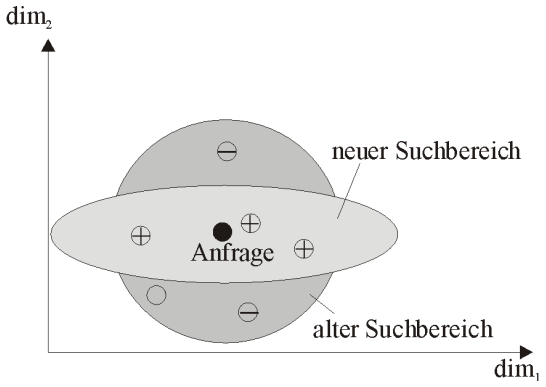


Abbildung 2.9: Dimensionsgewichtung

2.4.4 Kontext-Feedback

Kontext-Feedback ist eine erweiterte Form des Relevanz-Feedbacks. Neben der Unterscheidung zwischen relevanten und nicht-relevanten Objekten kann der Nutzer die Objekte entsprechend seines Ähnlichkeitsempfindes grafisch relativ zueinander positionieren (Santini und Jain, 2000). In einer zwei- oder auch dreidimensionalen Darstellung des Anfrageraums (Abbildung 2.10a) kann der Nutzer ähnliche Objekte nahe beieinander platzieren und unähnliche Objekte von einander entfernen (Abbildung 2.10b und 2.10c).

Die vom Nutzer vorgegebene gewünschte Ähnlichkeit wird vom System nachgebildet. Dafür findet eine neue Ähnlichkeitsberechnung auf Basis des durch die verschobenen Objekte neu aufgespannten Suchraums und der Darstellungsparameter, wie Zoom-Faktor und Präsentationsachsen, statt. Analog zum Relevanz-Feedback wird das neue Anfrageergebnis präsentiert (Abbildung 2.10d). Daraufhin kann eine neue Bewertung beziehungsweise Positionierung der Objekte durch den Nutzer vorgenommen werden.

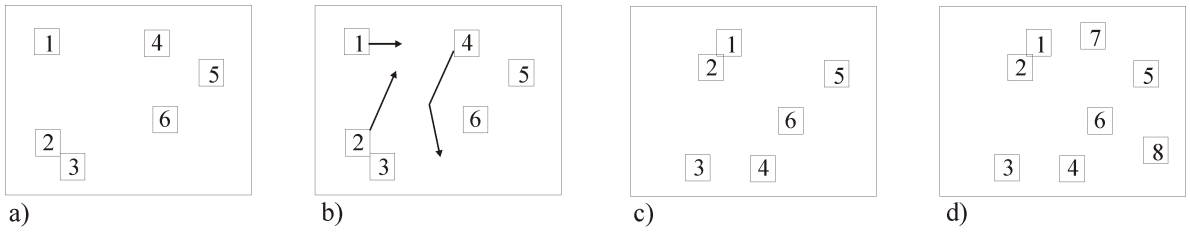


Abbildung 2.10: Kontext-Feedback im zweidimensionalen Raum

Die Idee des *visuellen Konzeptes* stellt eine Erweiterung dieser Methode dar (Santini und Jain, 2000). Objekte, die der Nutzer als ähnlich oder äquivalent empfindet, können einem visuellen Konzept zugeordnet werden. In Abbildung 2.11b werden die Objekte 1 und 2 in einer Konzept-Box (KB) zusammengefasst. Eine Konzept-Box kann auch als Cluster aufgefasst werden. Beschreibende Informationen können als Metadaten der Konzept-Box hinzugefügt werden. Im Benutzerinterface werden die Objekte 1 und 2 durch die Konzept-Box ersetzt. Die Positionierung der Objekte beziehungsweise der Konzept-Box (Abbildung 2.11c) und die Ergebnisdarstellung (Abbildung 2.11d) findet analog zum normalen Kontext-Feedback statt.

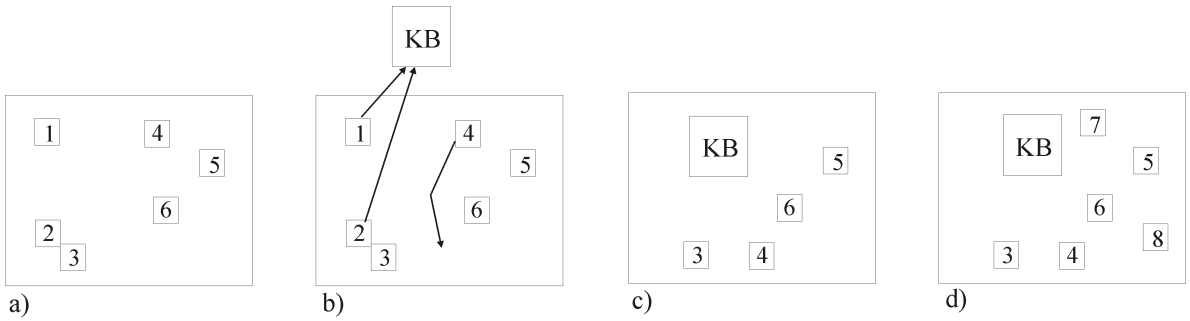


Abbildung 2.11: Kontext-Feedback mit einem visuellen Konzept

2.4.5 Nutzerprofile

Da das Suchverhalten und der -bedarf von Nutzer zu Nutzer differiert, ist die Verwaltung der Nutzerinformationen in einem sogenannten Nutzerprofil sinnvoll. Durch eine Klassifizierung der Nutzer in verschiedene Nutzergruppen können anstelle der individuellen Nutzerprofile auch Profile für die einzelnen Nutzergruppen erstellt werden. Voraussetzung dafür ist natürlich die Identifikation des jeweiligen Nutzers beziehungsweise der Nutzergruppe. Die Subjektivität des Ähnlichkeitsempfindens wird für jeden Nutzer oder jede Nutzergruppe als Profil modelliert.

Die in einem Profil gespeicherten Informationen können explizit oder implizit bestimmt werden. Zu den explizit gegebenen Nutzerinformationen zählen beispielsweise demographische Merkmale, die vom Nutzer eingegeben werden. Die impliziten Informationen wer-

den während des Retrievals, beispielsweise aus der Relevanz-Bewertung des Nutzers im Rahmen der iterativen Anfrageverfeinerung, gewonnen.

Auf Grund der in einem Nutzerprofil verwalteten Informationen ist es möglich, eine Anpassung der Gewichtung bei der Anfrageformulierung des entsprechenden Nutzers vorzunehmen. Dies ermöglicht somit bereits bei einer Initialanfrage eine implizite Anpassung der Anfrage an die allgemeinen Präferenzen des jeweiligen Nutzers. Ferner können die gewonnenen Informationen direkt für die Bearbeitung sowie zur Verfeinerung von nachfolgenden Anfragen genutzt werden. Ein mögliche Strategie dabei ist die Haltung von *schwarzen Listen*. Diese enthalten all diejenigen Objekte, die bereits einmal vom Nutzer negativ bewertet wurden. Somit kann gewährleistet werden, dass negativ bewertete Objekte innerhalb einer Sitzung nicht noch einmal als Ergebnisobjekt erscheinen. Neben einer schwarzen Liste, können auch positiv bewertete Objekte im Profil verwaltet werden.

Eine andere Möglichkeit ist die Verwendung der Informationen eines Nutzers zur Beantwortung ähnlicher Anfragen eines anderen Nutzers. So kann beispielsweise davon ausgegangen werden, dass Objekte, die von vielen Nutzern bei ähnlichen Anfragen als irrelevant beurteilt wurden, auch in Zukunft für die Beantwortung ähnlicher Anfragen nicht von Bedeutung sind. Eine solche Überführung der einzelnen Nutzerinformationen hin zu allgemeinen Informationen ist somit sinnvoll.

Profile werden also verwendet, um die gesammelten Informationen für eine effizientere und personalisierte Anfragebearbeitung zu nutzen. Dabei können zwei methodische Ansätze unterschieden werden (Balabanovic und Shoham, 1997):

- inhaltsbasiertes Filtern (content-based Filtering) und
- kollaboratives Filtern (collaborative Filtering).

Beim inhaltsbasierten Filtern wird die Anfrage unter Berücksichtigung des Profils des Nutzers, der die Anfrage stellt, bearbeitet. Hingegen wird beim kollaborativen Filtern auf die Informationen ähnlicher Nutzer zurückgegriffen. Abbildung 2.12 verdeutlicht diesen Zusammenhang.

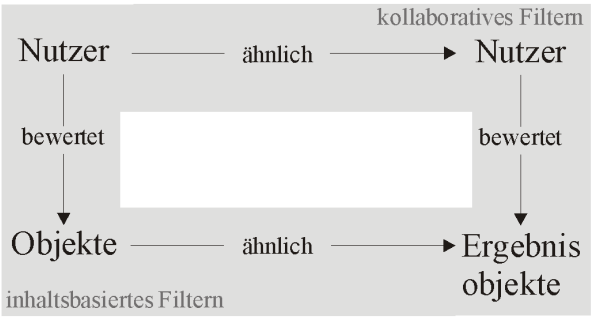


Abbildung 2.12: Alternativen beim Retrieval auf Basis von Profilen

2.5 Zusammenfassung

In diesem Kapitel wurden die grundlegenden Begriffe im Zusammenhang mit Multimedia-Information-Retrieval eingeführt. Es wurde kurz auf die Prinzipien des Information-Retrievals eingegangen. Dabei wurden die wichtigsten Retrieval-Modelle vorgestellt.

Multimedia-Information-Retrieval setzt auf dem Information-Retrieval und dem Daten-Retrieval auf. Es erweitert diese um die Möglichkeit zur Suche in Multimedia-Daten. Wesentlich dabei ist, dass sowohl zeitabhängige als auch zeitunabhängige Multimedia-Daten verarbeitet werden. Die besonderen Charakteristika des Multimedia-Information-Retrievals sind eng mit der Art der zu verwaltenden Daten verbunden. So müssen große Datenmengen sowie eine große Vielfalt an unterschiedlichen Datentypen verwaltet und dem Nutzer zugänglich gemacht werden. Eine wesentliche Besonderheit liegt ferner in der impliziten Semantik von Multimedia-Objekten, welche für das Retrieval die Extraktion verschiedenster Features notwendig macht. Weiterhin zeichnet sich das Multimedia-Information-Retrieval dadurch aus, dass sowohl zeitliche, räumliche und inhaltsbasierte als auch scharfe sowie unscharfe Anfragen unterstützt werden müssen.

Neben der Hervorhebung der Besonderheiten beim Multimedia-Retrieval wurde die prinzipielle Arbeitsweise eines Multimedia-Information-Retrieval-Systems beschrieben. Der Retrieval-Prozess läuft in verschiedenen Schritten ab. Vordergründig dabei ist die Aufbereitung der Multimedia-Objekte und der Anfrage, welche eine Zerlegung, Normalisierung, Segmentierung der Multimedia-Objekte sowie eine Feature-Extraktion impliziert. Dadurch wird ein inhaltsbasiertes Retrieval auf Basis der inhaltsabhängigen Metadaten gewährleistet. Daneben werden aber auch inhaltsunabhängige und inhaltsbeschreibende Metadaten, wie beispielsweise Strukturdaten oder zusätzliche beschreibende relationale Daten, verwendet. Aus diesem Grund wird im Allgemeinen zwischen dem schlüsselwort-, dem inhalts- sowie dem konzeptbasiertem Retrieval unterschieden. Eine wichtige Anforderung wird im Rahmen der Ergebnispräsentation an das Retrieval-System gestellt. Es müssen Methoden zur Darstellung der Retrieval-Ergebnisse, welche die unterschiedlichsten Medien-Objekte beinhalten können, vorhanden sein. Ferner sind Visualisierungstechniken erforderlich, die dem Nutzer die Zusammenhänge zwischen der Anfrage und den Ergebnisobjekten verständlich machen.

Einen weiteren Schwerpunkt in diesem Kapitel stellte die Einbindung von Nutzerpräferenzen in das Multimedia-Retrieval dar. Für die Entwicklung einer Anfragesprache sind die Möglichkeiten zur Spezifikation der Präferenzen bei der Anfrageformulierung von Bedeutung. Eine Möglichkeit dafür bieten die Präferenzanfragen. Sie sind im wesentlichen für Boolesche Anfragen gedacht, können jedoch im Multimedia-Retrieval beispielsweise im Zusammenhang mit einem schlüsselwort- oder einem konzeptbasierten Retrieval sinnvoll eingesetzt werden.

Eine andere Alternative zur Formulierung von Präferenzen bietet die Vergabe von Gewichten innerhalb einer Anfrage. Dabei wird zwischen Relevanz-, Quantitäts-, Operator- und Schwellwertgewichten unterschieden. Die Relevanzgewichte drücken das Empfinden der Wichtigkeit eines Nutzers bezüglich einer Teilanfrage aus. Diese Art der Gewichtung verfügt über eine hohe Ausdrucksfähigkeit und überlässt dem Nutzer Spielraum bei der Abbildung seiner Präferenzen. Die Relevanzgewichtung hat sich gegenüber den Präferenzanfragen im Multimedia-Retrieval durchgesetzt (Fagin und Wimmers, 2000). Die Operatorgewichtung bietet dem Nutzer die Möglichkeit, das Verhalten der Operatoren *UND* und *ODER* zu verändern und damit die Teilanfragen unscharf zu verknüpfen. Mittels eines Quantitätsgewichtes

kann die Anzahl der zu liefernden Anfrageergebnisse beeinflusst werden. Letztlich kann der Relevanzwert der Objekte in Abhängigkeit von einem Schwellwertgewicht verändert werden. Ein Schwellwertgewicht kann sowohl für einzelne Teilanfragen als auch für die ganze Anfrage angegeben werden.

Beim Relevanz- und Kontext-Feedback wird eine initiale Anfrage mit dem Ziel modifiziert, eine erneute Anfrage entsprechend der gewünschten Suchrichtung anzupassen. Entscheidend ist die Benutzerrelevanz, die durch die Bewertung der Ergebnisobjekte abgebildet wird. Dieses Feedback fließt in die Anpassung der Anfrage ein. Dabei werden verschiedene Methoden, wie die Verschiebung des Anfragepunktes oder die Modifizierung der Gewichte, genutzt. Letztere baut dabei auf den Konzepten der Relevanzgewichtung auf.

Durch Nutzerprofile ist eine Personalisierung des Suchverhaltens für einen Nutzer realisierbar. Die Anfragen werden auf Basis des Profils vom System so modifiziert, dass sie auf die über einen längeren Zeitraum beobachteten individuellen Präferenzen eines Nutzers oder ähnlicher Nutzer zugeschnitten sind.

3 Multimedia-Anfragesprachen

Traditionelle Anfragesprachen relationaler beziehungsweise objektorientierter Datenbank-Management-Systeme sind für den Einsatz in einem MIRS auf Grund der Charakteristika und Besonderheiten des Multimedia-Information-Retrievals (siehe Abschnitt 2.3.1) nicht mächtig genug. So können Anfragen in MIRS beispielsweise durch SQL, OQL oder QBE nicht angemessen formuliert werden. Aus diesem Grund wurden spezielle Multimedia-Anfragesprachen entwickelt, welche in diesem Kapitel näher untersucht werden.

Im Allgemeinen beschreiben Anfragesprachen die Struktur und die Auswahl von Daten, ohne spezielle Algorithmen zur Anfrageberechnung vorzugeben. Sie werden in *deklarative* und *prozedurale* Anfragesprachen unterteilt. Bei deklarativen Anfragesprachen gibt der Nutzer an, was gesucht werden soll, während bei prozeduralen Sprachen angegeben wird, was und insbesondere wie gesucht werden soll. Daher handelt es sich bei den Multimedia-Anfragesprachen um deklarative Sprachen.

In diesem Kapitel werden neben den allgemeinen Anforderungen an eine Anfragesprache weitere Anforderungen, bedingt durch die speziellen Charakteristika von MIRS, dargelegt, denen eine Multimedia-Anfragesprache gerecht werden muss. Auf der Formulierung dieser Anforderungen aufbauend, erfolgt die Evaluierung verschiedener bestehender Multimedia-Anfragesprachen. Hierbei stehen insbesondere die Möglichkeiten zur Abbildung von Nutzerpräferenzen durch entsprechende Sprachkonstrukte der jeweiligen Anfragesprachen im Vordergrund. Des Weiteren existieren verschiedene theoretisch fundierte Arbeiten zur Einbindung von Nutzerpräferenzen, die in diesem Kapitel vorgestellt werden. Diese Ansätze erweitern entweder die relationale Algebra oder das relationale Kalkül um entsprechende Mechanismen und werden nur zu Teilen in Form einer nutzerorientierten Anfragesprache angeboten. Ferner wird auf einzelne Erweiterungen von SQL eingegangen, die für Multimedia-Anfragen genutzt werden können.

3.1 Anforderungen an eine Multimedia-Anfragesprache

Auf Grund der Charakteristika von MIRS ergeben sich neben den allgemeinen Anforderungen an eine Anfragesprache weitere Anforderungen, denen eine Multimedia-Anfragesprache gerecht werden muss. Dazu zählen Anforderungen, die sich aus der Formulierung spezieller Anfragen in MIRS ergeben. Zwar kann unter Verwendung traditioneller Anfragesprachen eine exakte Suche auf den inhaltsbeschreibenden und inhaltsunabhängigen Metadaten erfolgen, jedoch wird ein inhaltsbasiertes, unscharfes Retrieval nicht unterstützt. Neben der Gewährleistung von inhaltsbasierten, unscharfen Anfragen fordern Fuhr u. a. (1998), dass ein MIRS weiterhin räumliche und zeitliche Anfragen unterstützen muss. Diese werden mit dem Begriff *Multimedia-Anfragen* zusammengefaßt. Mit der Formulierung von Multimedia-Anfragen und der Heterogenität der Daten geht die Notwendigkeit einher, spezielle Mechanismen zur Präsentation der Anfrageergebnisse bereitzustellen. Im Folgenden wird daher

näher auf:

- die allgemeinen Anforderungen an eine Anfragesprache,
- die Anforderungen zur Unterstützung von Multimedia-Anfragen sowie
- die Anforderungen bezüglich einer adäquaten Ergebnispräsentation eingegangen.

3.1.1 Allgemeine Anforderungen

Heuer und Scholl (1991) entwickelten einen Kriterienkatalog für Anfragesprachen. In der Arbeit von Paskamp (1999) wurden auf Basis einer verfeinerten Kriterienliste (Heuer und Saake, 2000; Saake u. a., 1997) die Anforderungen bezüglich der Gültigkeit für eine Multimedia-Anfragesprache kritisch untersucht. Die Ergebnisse dieser Untersuchung werden im Folgenden mit einer kurzen Darstellung der entsprechenden Kriterien zusammengefasst:

Ad-hoc-Formulierung; Anfragen können direkt über eine interaktive Benutzerschnittstelle formuliert werden. Dabei müssen keine vollständigen Programme geschrieben werden.

Eine Multimedia-Anfragesprache muss Methoden anbieten, die neben textbasierten Anfragekonstrukten auch andere medientypische Interaktionsverfahren unterstützen. Dafür bietet sich oftmals eine grafische Form der Anfrageformulierung an.

Generische Operatoren; Die Anfragesprache basiert auf wenigen generischen Operatoren, das heißt, für semantisch gleiche Operationen auf verschiedenen Objekten wird die gleiche Syntax verwendet.

Eine Multimedia-Anfragesprache muss generische Operationen für verschiedene Medien-Objekte anbieten.

Anwendungsunabhängigkeit; Der Sprachentwurf darf nicht im Hinblick auf ein bestimmtes Anwendungsgebiet erfolgen. Vielmehr soll eine universelle Anwendung unterstützt werden.

Dies gilt auch für eine Multimedia-Anfragesprache. Sie muss diverse Medientypen, mindestens jedoch Video-, Audio-, Bild- und Textdaten unterstützen. Ferner muss eine Unabhängigkeit von konkreten Medienformaten oder Kompressionsverfahren gewährleistet sein.

Deskriptivität; Der Zugriff auf Datenbankobjekte erfolgt durch die Beschreibung der Struktur und des Inhalts der gesuchten Objekte. Dabei werden keine Berechnungsalgorithmen vorgegeben. Deskriptivität garantiert eine Datenunabhängigkeit und lässt Freiraum für eine Anfrageoptimierung.

Diese Anforderung gilt uneingeschränkt für Multimedia-Anfragesprachen. Daher müssen auch für die Beschreibung der Ergebnispräsentation Sprachkonstrukte vorhanden sein, die die Struktur und den Inhalt der Präsentation beschreiben, jedoch nicht, wie diese zu erstellen ist.

Mengenorientiertheit; Klassischen Anfragen arbeiten auf Mengen. Dadurch kann das System das Ergebnis in einer beliebigen Reihenfolge zurückliefern. Durch die Unterstützung von Mengenoperationen kann eine optimierte Anfragenbearbeitung erfolgen.

In MIRS werden nicht nur exakte sondern auch unscharfe Anfragen formuliert. Dadurch liegen keine klassischen Mengen sondern Fuzzy-Mengen vor. Daher müssen die einzelnen Mengenoperationen um Konzepte zur Behandlung von Relevanzwerten erweitert werden.

Orthogonalität; Die Operationen müssen frei und möglichst beliebig kombinierbar sein. Die Orthogonalität gewährleistet eine Schachtelung von Anfragen.

Eine Multimedia-Anfragesprache sollte die Kombination von Konstrukten, sofern eine Typübereinstimmung gewährleistet ist, erlauben. Dabei ist es unwesentlich, ob es sich um einfache Datentypen oder Medientypen handelt.

Effizienz; Die Operationen sollen effizient realisierbar sein.

Prinzipiell muss dies auch für Multimedia-Anfragesprachen gelten. Jedoch treten insbesondere durch das hohe Datenvolumen und durch unscharfe Anfragen Effizienzprobleme auf. Ein weiteres Effizienzproblem kann hinsichtlich zusätzlicher anwendungsspezifischer Funktionen und Prädikate entstehen. Bezüglich der Ergebnispräsentation kann jedoch durch eine entsprechende Parametrisierung der Ausgabe, wie zum Beispiel der Qualität des auszugebenden Medien-Objektes, die Effizienz des Systems gesteigert werden. Für diesen Zweck muss die Anfragesprache entsprechende Ausdrucksmittel bereitstellen.

Erweiterbarkeit; Bei einer Erweiterung des zu Grunde liegenden Datenmodells soll auch die Anfragesprache leicht erweiterbar sein.

Insbesondere bei MIRS spielt die Erweiterbarkeit eine wichtige Rolle. So können neue Medientypen hinzugefügt oder zusätzliche Features extrahiert werden. Eine Multimedia-Anfragesprache muss solche Änderungen leicht adaptieren können und beispielsweise die Möglichkeit bieten, auf neue Features in Anfragen zuzugreifen.

Abgeschlossenheit; Das Ergebnis einer Anfrage muss konsistent im Datenbankmodell darstellbar sein und somit als Eingabe für nachfolgende Anfragen verwendbar sein.

In einem MIRS muss die Anfragesprache die Semantik der Medien-Objekte und des Relevanzwertes als Anfrageergebnis festlegen. Relevanzwerte können beispielsweise als zusätzliches Attribut den Ergebnisobjekten zugeordnet werden. Wesentlich dabei ist, dass die ermittelten Ergebnisrelationen wieder Eingangsrelationen für nachfolgende Anfragen darstellen.

Adäquatheit; Es wird gefordert, dass alle Konstrukte des Datenbankmodells durch die Anfragesprache ausgenutzt werden. Die Adäquatheit bildet das Gegenstück zur Abgeschlossenheit.

Für eine Multimedia-Anfragesprache gilt, dass Konstrukte zur Formulierung der zeitlichen und räumlichen Beziehungen sowie zur Formulierung unscharfer, inhaltsbasierter und gewichteter Anfragen zur Verfügung stehen. Auch müssen Sprachmittel für die Parametrisierung der Ergebnispräsentation angeboten werden.

Vollständigkeit; Die Mächtigkeit einer Standardsprache wird als Mindestanforderung für Anfragesprachen vorausgesetzt. Somit wird von relationalen Anfragesprachen zum Beispiel die gleiche Ausdrucksfähigkeit, wie sie die relationale Algebra aufweist, verlangt.

Eine Multimedia-Anfragesprache muss damit in Abhängigkeit vom zu Grunde liegenden Datenmodell mindestens die gleiche Mächtigkeit wie die relationale Algebra beziehungsweise das objektorientierte Äquivalent aufweisen.

Sicherheit; Jede syntaktisch korrekte Anfrage muss terminieren und ein endliches Ergebnis liefern. Um die Sicherheit gewährleisten zu können, darf die Anfragesprache nicht berechnungsvollständig sein.

Dieses gilt natürlich auch für Multimedia-Anfragesprachen. Auf den Aspekt der Sicherheit wird in Abschnitt 5.2 im Zusammenhang mit dem Entwurf einer Multimedia-Anfragesprache näher eingegangen.

Optimierbarkeit; Auf Basis von Optimierungsregeln kann eine automatische Anfrageoptimierung stattfinden.

Dieses Kriterium muss auch von Anfragen in MIRS erfüllt werden. Dabei treten auf Grund der Gewichtung in Anfragen Probleme auf. Dieses wird in Abschnitt 4.3.3 diskutiert.

Formale Semantik; Die Anfrageoperationen sind formal definiert. Durch die formale Semantik wird die Bedeutung eines Anfrageterms eindeutig festgelegt.

Eine formale Semantik muss auch für alle Operationen einer Multimedia-Anfragesprache existieren.

Eingeschränktheit; Unter Berücksichtigung von Effizienz, Optimierbarkeit und Sicherheit kann eine Anfragesprache nicht berechnungsvollständig sein.

Die Eingeschränktheit gilt auch für Multimedia-Anfragesprachen.

Diese allgemeinen Anforderungen an eine Anfragesprache verfolgen zum Teil gegensätzliche Ziele. Daher kann keine Anfragesprache alle Kriterien gleich gut erfüllen (Saake u. a., 1997, S. 236). Im Folgenden wird auf Anforderungen, die sich auf Grund der spezifischen Charakteristika eines MIRS ergeben, näher eingegangen.

3.1.2 Multimediaspezifische Anforderungen

Multimedia-Information-Retrieval-Systeme zeichnen sich insbesondere dadurch aus, dass sie neben exakten Anfragen auf numerischen und alphanumerischen Daten, wie beim Daten-Retrieval üblich, weitere Konzepte zur Formulierung multimediaspezifischer Anfragen berücksichtigen müssen. Dazu zählen:

- *Unscharfe Anfragen;* Für die Formulierung von Anfragen können, anders als beim Daten-Retrieval, auch Fuzzy-Prädikate genutzt werden.
- *Inhaltsbasierte Anfragen;* Die Suche erfolgt auf Basis der inhaltsabhängigen Metadaten der Medien-Objekte. Wesentlich dabei ist, dass diese Anfragen oftmals unscharf, beispielsweise unter Verwendung eines Ähnlichkeitsoperators, formuliert werden.

- *Räumliche Anfragen;* Eine Formulierung von Anfragen bezüglich der räumlichen Anordnung von Objekten und deren Beziehungen sowohl zwischen als auch innerhalb von Medien-Objekten ist möglich.
- *Zeitliche Anfragen;* Die Suche erfolgt über zeitliche Zusammenhänge innerhalb und zwischen Medien-Objekten.
- *Gewichtete Anfragen und Präferenzanfragen;* Bei der Anfrageformulierung kann eine Gewichtung, beispielsweise von Bedingungen oder Operatoren, zusätzliche Präferenzen des Nutzers abbilden. Gleiches gilt für die Formulierung von Präferenzanfragen.

Prinzipiell können die einzelnen Anfragearten miteinander verknüpft werden. Das Beispiel 3.1 zeigt eine Anfrage, die eine scharfe Bedingung (Datum = 12.12.2003), eine unscharfe inhaltsbasierte Anfrage (Bildähnlichkeit), eine räumliche Bedingung (Text rechts von Bild) sowie eine Gewichtung (Farbe versus Form) miteinander verknüpft.

Beispiel 3.1 *Suche die HTML-Dokumente, die am 12.12.2003 erstellt wurden. Der Text soll sich rechts von einem Bild befinden und das Bild soll zu dem gegebenen Bild in Farbe und Form ähnlich sein. Dabei ist die Ähnlichkeit bezüglich der Farbe wichtiger als bezüglich der Form.*

Auf Grund der Unschärfe von Multimedia-Anfragen ist das Ergebnis einer initialen Anfrage nicht immer zufriedenstellend. Aus diesem Grund muss eine Multimedia-Anfragesprache auch Sprachkonstrukte anbieten, die eine iterative Anfrageverfeinerung ermöglichen.

Beim Daten-Retrieval stehen die klassischen Vergleichsoperatoren ($<, >, =, \neq, \geq, \leq$) sowie Funktionen für die jeweiligen Datentypen, wie die Addition von zwei Zahlen, bereit. Da beim Multimedia-Information-Retrieval auch verschiedenste Multimedia-Daten vorliegen, sind die oben aufgeführten klassischen Funktionen und Vergleichsoperatoren nicht ausreichend. Aus diesem Grund müssen spezielle Prädikate, Funktionen und Konzepte für die Formulierung der verschiedenen Multimedia-Anfragen zur Verfügung stehen.

Im Folgenden wird auf die unterschiedlichen Möglichkeiten zur Formulierung von Anfragen in einem MIRS näher eingegangen. Dabei wird eine Grundmenge an notwendigen Prädikaten und Funktionen für die Unterstützung der jeweiligen Anfrageart skizziert. Je nach Anwendungsgebiet werden zusätzlich zu den dargelegten Operationen oftmals weitere Prädikate und Funktionen benötigt, auf die hier jedoch nicht weiter eingegangen wird.

Inhaltsbasierte und unscharfe Anfragen

Bei dieser Art von Anfragen steht die Suche auf Basis der inhaltsabhängigen Beschreibung der Medien-Objekte im Vordergrund. Meyer-Wegener definiert eine inhaltsbasierte Anfrage als die Suche in einer Menge von Medien-Objekten, bei der der Vergleich nicht auf einer syntaktischen Ebene sondern auf einer semantischen Ebene ausgeführt wird (Meyer-Wegener, 2003, S. 214). Im Folgenden sind einige typische Beispiele für inhaltsbasierte Anfragen, welche oftmals, aber nicht zwingend, als Query-By-Example-Anfragen formuliert werden, angegeben:

- Doublettensuche in einer Gemäldedatenbank auf Basis der Abbildungen der Gemälde,
- Suche nach farblich passenden Fotos für die Erstellung einer Broschüre in einer Werbeagentur oder

- Suche nach einem Cartoon, in dem eine Figur einen bestimmten Satz spricht, welcher als Audio-Clip angegeben wird.

Zu den inhaltsbasierten Anfragen zählen Ähnlichkeitsanfragen. Für die Umsetzung solcher Anfragen sind Ähnlichkeitsprädikate notwendig. Dabei wird zu einem oder mehreren gegebenen Medien-Objekten der Grad der Ähnlichkeit zu den Datenbank-Objekten bezüglich eines oder mehrerer Kriterien ermittelt. Die Anfragen selbst werden bereits unscharf unter Verwendung von Fuzzy-Prädikaten formuliert. Damit ist auch das Ergebnis unscharf. Dies bedeutet, dass zwar eine scharfe Menge an Ergebnisobjekten ermittelt wird, wobei jedoch die Ergebnisobjekte selbst die Anfrage nur zu einem Grad zwischen 0 und 1 erfüllen. Dieser Grad der Ähnlichkeit eines Datenbank-Objektes zur Anfrage wird durch den Relevanzwert ausgedrückt.

Klassischerweise werden Anfragen bezüglich der inhaltsabhängigen oder -unabhängigen Metadaten der Medien-Objekte scharf formuliert. Wesentlich beim Multimedia-Retrieval ist, dass diese Anfragen nicht nur scharf sondern auch unscharf formuliert werden können. Für die Umsetzung solcher unscharfen Anfragen kommen Techniken aus dem Information-Retrieval zum Tragen. Ferner finden in MIRS auch Fuzzy-Ausdrücke oder -Prädikate, wie *viel* oder *typisch*, Anwendung. Dies ist in Beispiel 3.2 dargestellt. Auch bei Ähnlichkeitsanfragen können Fuzzy-Prädikate, wie zum Beispiel *kreisförmig*, eingesetzt werden. Jeder dieser unscharfen Ausdrücke repräsentiert dabei eine Menge akzeptabler Werte, bezüglich der ein Attribut oder Feature abgeglichen wird (Bertino u. a., 1999).

Beispiel 3.2 *Suche die HTML-Dokumente, die viele Bilder enthalten.*

Räumliche Anfragen

Räumliche Beziehungen können sowohl innerhalb von Medien-Objekten (Beispiel 3.3) als auch zwischen den Medien-Objekten auftreten (Beispiel 3.4). Für die Abbildung dieser Beziehungen müssen entsprechende Prädikate, wie beispielsweise *links_von*, *teil_von* et cetera, zur Verfügung stehen.

Beispiel 3.3 *Suche Bilder mit einem Haus links von einem Baum.*

Beispiel 3.4 *Suche HTML-Dokumente, deren Text links von einem Bild liegt.*

Die Grundlage für die Formulierung räumlicher Anfragen bildet ein entsprechendes Datenmodell. Zumeist sind diese auf zwei Dimensionen beschränkt, jedoch leicht um eine dritte Dimension erweiterbar. Für die Bestimmung der räumlichen Beziehungen innerhalb eines Medien-Objektes müssen zunächst die relevanten Objekte im Medien-Objekt bestimmt werden.

Ermittlung der Objektgrenzen Die Abbildung von räumlichen Beziehungen innerhalb von Medien-Objekten macht eine Segmentierung und Identifizierung der relevanten Objekte, wie etwa *Haus* und *Baum* in Beispiel 3.3, erforderlich. Diese werden auch als *Salient-Objekte* bezeichnet. Wesentlich dabei ist die Bestimmung der Objektgrenzen. Dies ist automatisch schwer realisierbar, so dass dies überwiegend manuell beziehungsweise semi-automatisch erfolgt.

Oftmals wird anstelle der segmentierten Objekte eine vereinfachte Darstellung in der Datenbank abgelegt und zur räumlichen Indexierung und Suche genutzt. Beispielsweise kann das *kleinste umschreibende Rechteck* (Minimum Bounding Rectangle - MBR) oder ein *Polygonzug* des segmentierten Objektes genutzt werden (Prabhakaran, 1997, S. 98f). Diese beiden Alternativen sind in Abbildung 3.1 dargestellt.

Probleme bei einer solchen vereinfachenden Darstellung von Objekten treten jedoch dann auf, wenn exakte räumliche Beziehungen von Bedeutung sind. Beispielsweise können durch MBRs Überlappung entstehen, die real nicht existieren (siehe Abbildung 3.1a). Dadurch können falsche Anfrageergebnisse erzielt werden. Zur Vermeidung dieses Problems, können Objekte als *Punktmengen* abgespeichert werden. Dafür müssen jedoch höhere Speicherkosten und ein erhöhter Aufwand bei der Berechnung der räumlichen Beziehungen in Kauf genommen werden. Eine weitere Möglichkeit besteht in der Kombination dieser beiden Ansätze. So können zunächst die Objekte mittels eines MBR grob bestimmt werden. Dem schließt sich eine exakte Bestimmung der Objektgrenzen an. Für die ermittelten Salient-Objekte können dann die räumlichen Beziehungen bestimmt werden.

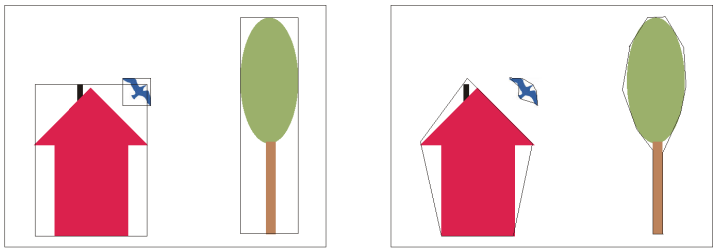


Abbildung 3.1: Darstellung der Salient-Objekte eines Bildes mittels a) MBR und b) Polygonzug

Bestimmung der räumlichen Beziehungen Räumliche Beziehungen können in *topologische*, *Richtungs-* und *Distanzbeziehungen* eingeteilt werden. Zur Definition von räumlichen Beziehungen in Multimedia-Information-Retrieval-Systemen setzen viele Arbeiten auf Ergebnissen aus der Forschung zu Geografischen-Informationssystemen (GIS) auf.

Die Distanzbeziehungen unterstützen Anfragen, wie in Beispiel 3.5 dargestellt. Sie beschreiben somit den räumlichen Abstand zwischen Objekten. Für die Berechnung der Distanz werden Distanzfunktionen, wie die Euklidische Distanz, eingesetzt. Da Distanzfunktionen den Abstand zwischen zwei Punkten berechnen, kann der Schwerpunkt der Objekte als jeweiliger Referenzpunkt gewählt werden.

Beispiel 3.5 *Suche die Bilder, bei denen sich ein Vogel in der Nähe eines Hauses befindet.*

Distanzen werden typischerweise durch Verwendung von vordefinierten Symbolen beschrieben. Hong u. a. (1995) verwenden dafür die vier symbolischen Distanzen *sehr nah*, *nah*, *fern* und *sehr fern*. Diese qualitativen Beziehungen müssen in entsprechende quantitative Beziehungen transformiert werden. Dabei korrespondiert jedes Distanzsymbol zu einem bestimmten Intervall an quantitativen Distanzen (siehe Abbildung 3.2). Die Festlegung der

Intervalle ist jedoch vom jeweiligen Anwendungskontext abhängig. Frank (1992) legt dar, wie aus einer kleinen Menge an qualitativen Distanzsymbolen weitere Distanzsymbole geschlussfolgert werden können.

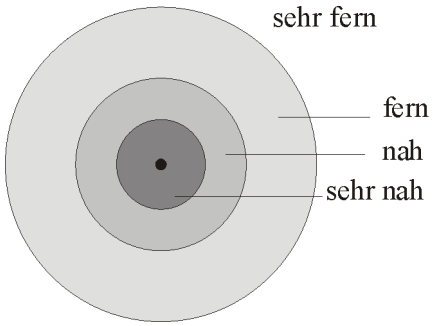


Abbildung 3.2: Qualitative Distanzen nach Hong u. a. (1995)

Richtungsbeziehungen beschreiben die räumliche Lage von Objekten zueinander (Li u. a., 1996). Ausgehend vom Referenzobjekt können die bekannten geografischen Richtungsbeziehungen, wie *Nord* und *Nordwest* im zweidimensionalen Raum, angewendet werden. Für eine vereinfachte Anfrageformulierung werden die einzelnen Richtungen zusammengefasst (Li u. a., 1996). Beispielsweise ergibt sich die Richtung *links* aus den Bereichen *Nordwest*, *West* und *Südwest* (ebd.). Dieser Ansatz ist sowohl für eine Darstellung der Salient-Objekte durch einen MBR, einen Polygonzug als auch durch eine Punktmenge geeignet.

Eine andere Möglichkeit zur Abbildung von Richtungsbeziehungen zwischen Objekten bietet der *2D-String* (Chang und Liu, 1984; Chang u. a., 1987). In einem 2D-String werden die räumlichen Beziehungen in einem Teilstring für die Beziehungen zwischen den Objekten bezüglich der x-Achse und in einem zweiten Teilstring für die Beziehungen zwischen den Objekten bezüglich der y-Achse repräsentiert. Es wird davon ausgegangen, dass die Objektgrenzen durch MBRs festgelegt sind, wobei die Grenzen parallel zur x- und y-Achse sind. Als Referenzpunkt für ein Objekt dient sein Schwerpunkt. Sei $S = \{O_1, O_2, \ldots, O_n\}$ eine Menge von segmentierten Objekten in einem Bild und $R = \{=, <\}$ ist die Menge der binären Vergleichsoperatoren mit folgender Bedeutung:

$O_1 = O_2$ O_1 ist an gleicher Stelle wie O_2,

$O_1 < O_2$ O_1 ist im Westen/Süden von O_2 (abhängig von der x- bzw. y-Achse).

Der Nachteil dieser Methode besteht darin, dass die räumlichen Beziehungen nur auf Basis des Schwerpunktes der Objekte bestimmt werden, so dass eine vollständige Wiedergabe der Beziehungen zwischen den Objekten nicht möglich ist. Das folgende Beispiel demonstriert die Verwendung des 2D-Strings.

Beispiel 3.6 *Betrachtet wird das Bild aus Abbildung 3.1a. Sei $S = \{Haus, Vogel, Baum\}$ dann ist der 2D-String $\{Haus < Vogel < Baum; Haus < Baum < Vogel\}$.*

Der *2D-C-String* erweitert den 2D-String, indem die räumlichen Beziehungen zwischen den Grenzen und nicht dem Schwerpunkt der Objekte angegeben werden (Lee und Hsu, 1990).

Daher wird die Menge der gültigen Vergleichsoperatoren entsprechend den von Allen (1983) vorgeschlagenen Relationen zwischen zwei Intervallen erweitert. Allens dreizehn mögliche Beziehungen pro Dimension zwischen zwei Intervallen sind in Abbildung 3.3 für die Beziehungen bezüglich der x-Achse aufgeführt. Diese Beziehungen sind auf beliebige Objekte übertragbar. Wenn die x- und die y-Richtung unabhängig voneinander betrachtet werden, so ergeben sich 169 räumliche Beziehungen zwischen zwei Objekten im zweidimensionalen Raum. Damit können die räumlichen Beziehungen in einem Bild angemessen repräsentiert werden (Petraglia u. a., 2001). Für den 2D-C-String gilt somit $R = \{<, :, |, \%, [,], /\}$. Die Beziehungen zwischen den Objekten werden unter Verwendung dieser Operatoren analog zum 2D-String für jede Dimension angegeben (Lee und Hsu, 1990).

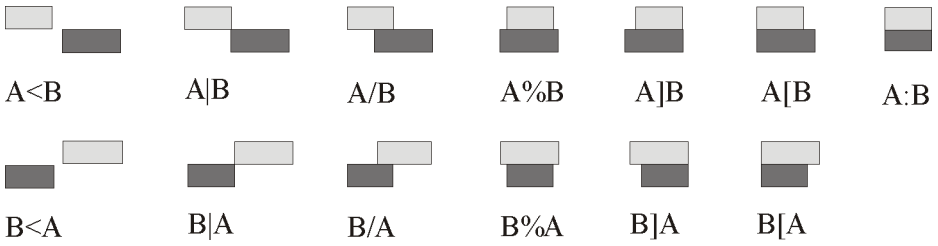


Abbildung 3.3: Eindimensionale Richtungsbeziehungen bezüglich der x-Achse

Neben diesen beiden Beziehungstypen sind auch die topologischen Beziehungen zwischen den Objekten von Bedeutung. Li u. a. (1996) argumentieren, dass nicht immer eine Richtungsbeziehung zwischen zwei Objekten besteht, jedoch stets eine topologische Beziehung. Nach Egenhofer und Franzosa (1991, 1995) gibt es acht grundlegende topologische Beziehungen zwischen zwei Objekten. Diese werden auf der Basis von neun verschiedenen Durchschnittsberechnungen unter Berücksichtigung der Grenzen sowie des Inneren und des Äußeren der Objekte im zweidimensionalen Raum ermittelt. So seien A^0 und B^0 das Innere, A^- und B^- das Komplement und ϑA und ϑB die Grenzen der Objekte A und B. Eine topologische Beziehung zwischen A und B ist ein neun-Tupel, das sich aus der Kombination der neun Durchschnitte $A^0 \cap B^0$, $A^0 \cap \vartheta B$, $\vartheta A \cap B^0$, $\vartheta A \cap \vartheta B$, $\vartheta A \cap B^-$, $A^0 \cap B^-$, $A^- \cap B^-$, $A^- \cap \vartheta B$ und $A^- \cap B^0$ zwischen dem Inneren, dem Komplement und den Grenzen der Objekte ergibt. Daraus resultieren die folgenden acht topologischen Beziehungen: *disjunkt*, *berührt*, *überlappt*, *bedeckt*, *bedeckt_von*, *in*, *enthält* sowie *gleich*. Auf diesen Ergebnissen aufbauend, schlagen Li u. a. (1996) sechs topologische Beziehungen für das Multimedia-Information-Retrieval vor. Abbildung 3.4 stellt diese Beziehungen grafisch dar.

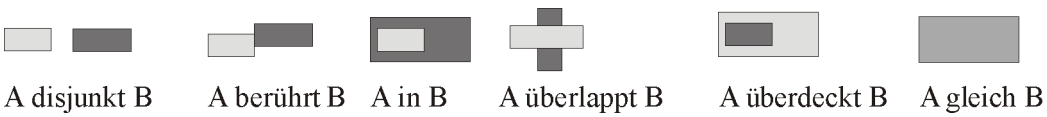


Abbildung 3.4: Topologische Beziehungen

Die Berechnung und Speicherung aller möglichen räumlichen Beziehungen von Objekten zueinander ist sehr aufwändig und oftmals nicht nötig, da der Nutzer meist nur an bestimmten räumlichen Beziehungen interessiert ist (Li u. a., 1996). Aus diesem Grund ist es sinnvoll,

gängige Beziehungen explizit zu speichern und andere weniger oft genutzte Beziehungen auf Anfrage berechnen zu lassen.

Räumliche Operationen Die Prädikate für die Formulierung räumlicher Anfragen lassen sich direkt aus den zuvor skizzierten Beziehungen zwischen zwei Objekten ableiten. Für die topologischen Beziehungen ergeben sich beispielsweise die Prädikate *disjunkt*, *berührt*, *in*, *überlappt*, *überdeckt* und *gleich*. Analog dazu können die Prädikate für die Richtungs- und Distanzbeziehungen abgeleitet werden. Zusätzlich sind spezielle Funktionen, beispielsweise zur Berechnung des Abstandes, des Flächeninhalts oder des Umfanges eines Objektes, sinnvoll. Auch Mengenoperation auf Objekten (Regionen), wie sie aus dem GIS-Umfeld bekannt sind, erscheinen praktikabel.

Zeitliche Anfragen

Zeitliche Anfragen sind insbesondere im Zusammenhang mit zeitabhängigen Medien, wie Video oder Audio, von Bedeutung. Wesentlich bei der Modellierung der Zeit ist die Granularität. Dabei wird zwischen verschiedenen diskreten Werten unterschieden. Im Allgemeinen wird zwischen Sekunde, Minute, Stunde, Tag, Monat und Jahr differenziert. Eine feinere beziehungsweise gröbere Einteilung ist möglich und vom jeweiligen Anwendungsgebiet abhängig.

Zeitliche Beziehungen Analog zu den räumlichen Beziehungen können zeitliche Beziehungen sowohl zwischen einzelnen Medien-Objekten als auch innerhalb eines Medien-Objektes auftreten. Die Abbildung von zeitlichen Beziehungen innerhalb von Medien-Objekten macht die Identifizierung wesentlicher zeitlicher Abschnitte erforderlich. Diese werden als *Sequenz* oder *Zeitintervall* bezeichnet. Im Folgenden werden sie als eigenständiges Objekt betrachtet.

Zwischen zwei Zeitintervallen und somit auch zwischen Medien-Objekten gibt es nach Allen (1983) dreizehn verschiedene Relationen. Diese entsprechen den räumlichen Beziehungen zwischen zwei Objekten, die bereits in Abbildung 3.3 dargestellt wurden. Bei der Betrachtung von zeitlichen Beziehungen wird die x-Achse dabei als eine systemweit gültige Zeitachse aufgefasst.

Zeitliche Operationen Die Prädikate für zeitliche Anfragen können direkt aus den skizzierten Beziehungen zwischen zwei Objekten abgeleitet werden. Des Weiteren sind spezielle Funktionen, beispielsweise zur Berechnung des Abstandes zweier Zeitpunkte oder zum Kürzen beziehungsweise Verlängern eines Zeitintervalls wünschenswert. Auch hier gilt, dass die konkreten Funktionen und deren Umfang stark vom jeweiligen Anwendungsgebiet abhängig sind. Die wichtigsten zeitlichen Operationen wurden von Paskamp (1999) zusammengetragen und sind in Tabelle 3.1 dargestellt.

Gewichtete Anfragen und Präferenzanfragen

Auf die verschiedenen Möglichkeiten zur Formulierung von gewichteten Anfragen wurde bereits ausführlich in Abschnitt 2.4.2 eingegangen.

Name	Operand 1	Operand 2	Ergebnis
$<, \%, :,], [, /, \|$	Intervall	Intervall	boolean
obere Intervallgrenze	Intervall		Zeitpunkt
untere Intervallgrenze	Intervall		Zeitpunkt
Intervalllänge	Intervall		Dauer
Durchschnitt, Vereinigung, Differenz	Intervall	Intervall	Intervall
$+, -$	Intervall	Dauer	Intervall
$<, >, =, \geq, \leq$	Zeitpunkt	Zeitpunkt	boolean
$+, -$	Zeitpunkt	Dauer	Zeitpunkt
Abstand	Zeitpunkt	Zeitpunkt	Dauer
Durchschnitt	Zeitpunkt	Intervall	Zeitpunkt
$<, >, =, \geq, \leq$	Dauer	Dauer	boolean
$+, -$	Dauer	Dauer	Dauer

Tabelle 3.1: Wichtige zeitliche Operationen (Paskamp, 1999, S. 32)

Im Zusammenhang mit Ähnlichkeitsanfragen und unscharfen Prädikaten ergibt die Anwendung von Präferenzanfragen entsprechend des Ansatzes von Lacroix und Lavency (1987) keinen Sinn, da bei diesen Anfragen jedes Datenbankobjekt zu einem bestimmten Grad die Anfrage erfüllt. Eine iterative Abschwächung der Anfrage kann somit nicht stattfinden. Bei Präferenzanfragen wird die Anfrage nur abgeschwächt, wenn eine leere oder eine zu kleine Ergebnismenge zurückgeliefert wird. Präferenzanfragen können daher nur auf Basis scharfer Prädikate formuliert werden. Dies schließt die Kombination von scharfen, schlüsselwortbasierten Anfragen mit zeitlichen und räumlichen Anfragen ein, wie das Beispiel 3.7 zeigt.

Beispiel 3.7 *Suche eine Videoszene vom 12.12.2003. Davon werden die Szenen bevorzugt, die nach 30 Sekunden einen Frame enthalten, das ein Haus und einen Baum zeigt. Von diesen Frames werden wiederum solche bevorzugt, in denen sich das Haus links vom Baum befindet.*

Ein weiteres sinnvolles Anwendungsgebiet der Präferenzanfragen ergibt sich im Zusammenhang mit dem konzeptbasierten Retrieval. Konzepte sind meist hierarchisch organisiert (siehe Abbildung 3.5), so dass durch Angabe von Präferenzen eine Einschränkung auf die relevanten Objekte erfolgen kann (Beispiel 3.8). Falls die Präferenzbedingung keine Ergebnisse liefert, ist eine weniger restriktive Bedingung entscheidend.

Beispiel 3.8 *Suche Abbildungen, die dem Konzept Flugzeug angehören. Dabei werden die Abbildungen bevorzugt, die zum Konzept Sportflugzeug gehören, wovon wiederum die bevorzugt werden, die einen Doppeldecker darstellen.*

3.1.3 Anforderungen an die Präsentation der Ergebnisse

Im Gegensatz zu klassischen Datenbank-Management-Systemen muss in einem MIRS die Darstellung diverser Medien-Typen gewährleistet werden. Neben der Ausgabe der Retrieval-

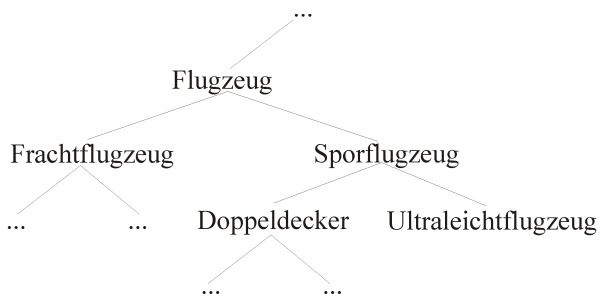


Abbildung 3.5: Ausschnitt aus einem Konzeptschema

Ergebnisse in Form einer Tabelle müssen daher auch Mechanismen für die Präsentation von Bildern, Audio und Videos bereit stehen.

Für eine Darstellung der Ergebnisobjekte müssen die Anfrageergebnisse entsprechend den vom Nutzer gegebenen Vorgaben aufbereitet werden. Beispielsweise sind Formatumwandlungen oder Medienumsetzungen notwendig. Darüber hinaus ist es wichtig, bestimmte Qualitätsparameter, wie die Auflösung und Größe eines Bildes, angeben zu können. Ferner muss die Möglichkeit bestehen, räumliche und zeitliche Vorgaben für die Ergebnispräsentation zu formulieren. Für diesen Zweck muss die Anfragesprache entsprechende Ausdrucksmittel bereitstellen. Auf verschiedene Möglichkeiten zur Spezifikation einer zeitlichen und räumlichen Präsentation wird im Folgenden kurz eingegangen.

Räumliche Präsentation

Oftmals ist es wünschenswert, die Medien-Objekte entsprechend bestimmter räumlicher Vorgaben in einem Ausgabefenster zu präsentieren. Eine solche räumliche Anordnung zweier Medien-Objekte ist in Abbildung 3.6 dargestellt.

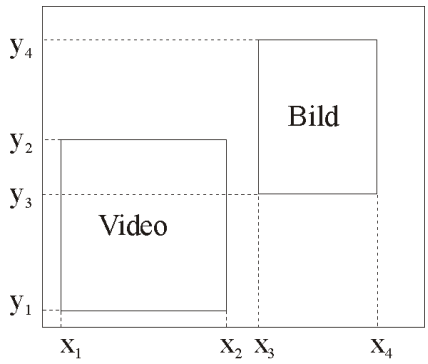


Abbildung 3.6: Räumliches Layout

Eine Möglichkeit, ein solches Layout zu beschreiben, besteht darin, die Position eines Medien-Objektes in einem Ausgabefenster durch Angabe der x- und y-Koordinaten der linken unteren sowie der rechten oberen Ecke des Objektes zu spezifizieren (Prabhakaran, 1997, S. 134f). Die Angabe der Bedingungen für die Koordinaten kann dabei flexibel oder statisch

erfolgen. Bei einer statischen Spezifikation werden die Koordinaten absolut angegeben. Beispielsweise gilt dann für das Videoobjekt aus Abbildung 3.6 $x_1 = 10, y_1 = 10, x_2 = 100, y_2 = 120$. Hingegen wird bei der flexiblen Spezifikation eine Menge von Bedingungen vorgegeben. Für die räumlichen Beziehungen der beiden Objekte in Abbildung 3.6 können beispielsweise folgende Beschränkungen formuliert werden:

- $x_2 - x_1 \leq 90$, $y_2 - y_1 \leq 110$ zur Beschreibung der Lage des Videoobjektes und
- $x_3 - x_1 \leq 50$, $y_3 - y_1 \leq 40$ zur Beschreibung der Beziehungen zwischen dem Video- und dem Bildobjekt.

Zur Auflösung dieser Beschränkungen kann nach Subrahmanian (1998) beispielsweise der Bellman-Ford-Algorithmus (siehe Saake und Sattler (2001), S. 427ff) verwendet werden.

Eine weitere Möglichkeit zur Formulierung der räumlichen Ausgabebedingungen besteht in der Verwendung des 2D-C-Strings. Für die Abbildung 3.6 ergibt sich dann $\{Video < Bild; Video / Bild\}$.

Zeitliche Präsentation

Insbesondere bei zeitlichen Anfragen und kontinuierlichen Medien spielt die zeitliche Präsentation der Medien-Objekte eine bedeutende Rolle. Die zeitliche Information, die mit der Präsentation eines Medien-Objektes assoziiert wird, ist durch folgende drei Parameter charakterisiert (Prabhakaran, 1997, S. 128):

- Beginn einer Objektpräsentation,
- Dauer der Präsentation und
- Synchronisation einer Objektpräsentation mit den Präsentationen anderer Objekte.

Der Beginn und die Dauer einer Objektpräsentation können analog zu den räumlichen Beziehungen statisch (Beispiel 3.9) oder flexibel (Beispiel 3.10) modelliert werden. Daher wird bei der Beschreibung zeitlicher Abhängigkeiten zwischen dem *statischen zeitlichen* und dem *flexiblen zeitlichen Modell* unterschieden.

Beispiel 3.9 *Zeige das Video um 12:00 für 5 Minuten.*

Beispiel 3.10 *Zeige das Video gleichzeitig mit den Untertiteln, nachdem das Musikstück, welches zwischen 12:00 und 12:05 beginnt, beendet ist.*

Mittels der dreizehn Relationen von Allen (1983) können zwar die zeitlichen Beziehungen zwischen Objekten beschrieben werden, jedoch ist es nicht möglich, die Parameter für die Präsentation der Medien-Objekte festzulegen. Aus diesem Grund wurden Modelle für die Abbildung der zeitlichen Beziehungen zwischen Objekten entwickelt. Die Schaffung eines zeitlichen Bezugs zwischen verschiedenen Medien-Objekten wird auch als Synchronisation bezeichnet (Steinmetz, 1999, S. 567). Dabei wird zwischen der *Interobjekt-Synchronisation* und der *Intraobjekt-Synchronisation* unterschieden, also der Abbildung der zeitlichen Beziehung innerhalb eines Medien-Objektes und zwischen verschiedenen Medien-Objekten (Blakowski und Steinmetz, 1996). Nach Blakowski u. a. (1991) existieren für die Synchronisation von Medien-Objekten drei verschiedene Ansätze, die im Folgenden kurz dargestellt werden.

Das *Timeline-Modell* ist ein einfaches, statisches, zeitliches Modell, bei dem die Medien-Objekte entlang eines Zeitstrahls platziert werden. Dadurch wird der Beginn und die Dauer der Objektpräsentation bestimmt. Es müssen sowohl der Startzeitpunkt als auch die Dauer aller Objekte bekannt sein. In diesem Modell werden nur die Parameter für einzelne Medien-Objekte, nicht jedoch die Abhängigkeiten zwischen ihnen beschreiben (Prabhakaran, 1997, S. 129f). Abbildung 3.7a zeigt eine Komposition verschiedener Medien-Objekte auf einer Zeitachse. Beispielsweise wird ab der Zeiteinheit t_3 für die Dauer von $t_4 - t_3$ sowohl ein Bildobjekt als auch ein zweites Audioobjekt präsentiert.

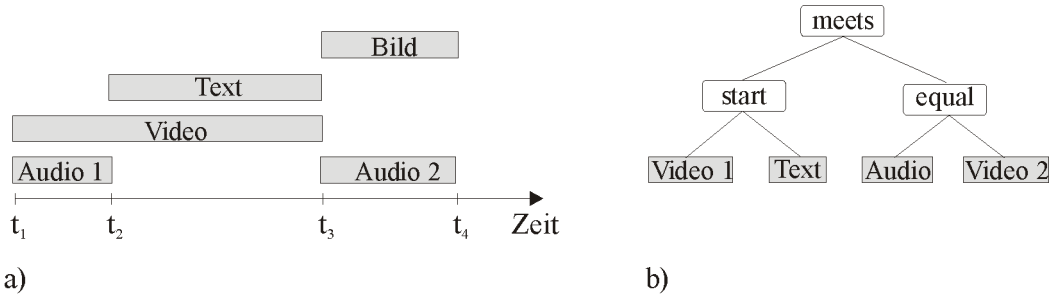


Abbildung 3.7: a) Komposition auf der Zeitachse (Timeline-Modell) b) Hierarchische Komposition

Die zeitliche Struktur zwischen Objektpräsentationen wird bei der *hierarchischen Komposition* in Form einer Baumstruktur modelliert (Herrtwich und Delgrossi, 1990). Die Blätter repräsentieren elementare Medien-Objekte. Ferner können in den Blättern Zeitintervalle zur zeitlichen Verzögerung der Präsentation angegeben werden. Die inneren Knoten und die Wurzel werden durch Operatoren gebildet, die die zeitlichen Beziehungen zwischen den Kindknoten herstellen. Hierfür wird auf die von Allen (1983) eingeführten Relationen zurückgegriffen. Ein Knoten repräsentiert somit zusammen mit seinen Kindknoten ein Multimedia-Objekt. Abbildung 3.7b zeigt ein Beispiel für die hierarchische Komposition. Der Wurzelknoten *meets* führt die Präsentation der Medien-Objekte in den beiden Kindknoten hintereinander aus. Der Operator *start* startet das erste Video- und das Textobjekt gleichzeitig, während der Operator *equal* veranlasst, dass das zweite Video- und das Audioobjekt zur gleichen Zeit gestartet und beendet werden.

Als dritter Ansatz wird die *Komposition über Referenzpunkte* (Steinmetz, 1990; Blakowski u. a., 1992) hervorgehoben. Als Referenzpunkte können beliebige Zeitpunkte der Präsentationsobjekte gewählt werden. Ferner wird eine Liste von gleichzeitig zu erreichenden Referenzpunkten angegeben, wodurch verschiedene Objekte zueinander in Beziehung gesetzt werden.

Daneben wurden in den letzten Jahren weitere zeitliche Modelle entwickelt, die nach Bertino und Ferrari (1998) in Graphenmodelle, Petri-Netz-Modelle, objektorientierte Modelle sowie sprachbasierte Modelle klassifiziert werden. Bei den Graphenmodellen werden die zeitlichen Beziehungen als Kanten im Graph dargestellt, während sie im Petri-Netz-Modell als Transitionen behandelt werden. Im objektorientierten Modell werden die Beziehungen als Objektattribute ausgedrückt und im sprachbasierten Modell wird dafür eine Programmiersprache eingesetzt (ebd.).

3.2 Prinzipien für den Entwurf einer Multimedia-Anfragesprache

Zunächst müssen für den Entwurf einer Multimedia-Anfragesprache die allgemeinen Anforderungen an Anfragesprachen Berücksichtigung finden. Sie gelten, wie in Abschnitt 3.1.1 dargelegt, auch für Multimedia-Anfragesprachen. Dabei ist die Forderung nach Anwendungsunabhängigkeit, also der Universalität, einer Multimedia-Anfragesprache wesentlich.

Anfragesprachen dienen zur Beschreibung von Prädikaten, mit denen Anfragen umgesetzt werden. Für eine Multimedia-Anfragesprache werden neben den traditionellen Vergleichsoperatoren zusätzlich unscharfe, zeitliche und räumliche Prädikate sowie applikationsspezifische Prädikate benötigt (Prabhakaran, 1997, S. 144f). Bertino u. a. (1999) klassifizieren die Prädikate, die durch eine Multimedia-Anfragesprache bereitgestellt werden müssen, wie folgt:

- *Attribut-Prädikate;* Darunter werden die Prädikate zur Formulierung scharfer Anfragen auf den Attributen, also dem strukturierten Inhalt, verstanden.
- *Struktur-Prädikate;* Diese Prädikate befassen sich mit der Struktur der Objekte.
- *Semantische Prädikate;* Sie beziehen sich auf die implizite Semantik und somit den unstrukturierten Inhalt der Medien-Objekte, welcher durch Features beschrieben wird.

Die Struktur-Prädikate und die semantischen Prädikate beziehen sich auch auf zeitliche und räumliche Eigenschaften der Medien-Objekte. Bei den semantischen Prädikaten sind jeweils die Beziehungen einzelner Objekte (Salient-Objekte, zeitliche Intervalle) innerhalb eines Medien-Objektes von Interesse. Die Struktur-Prädikate decken hingegen Beziehungen zwischen den verschiedenen Medien-Objekten ab.

Neben den Prädikaten können auch diverse Funktionen bei der Anfrageformulierung eingesetzt werden. Traditionell stehen dafür Funktionen auf einfachen Datentypen, wie Addition oder Subtraktion von Zahlen, zur Verfügung. Für eine Multimedia-Anfragesprache muss der Umfang der Funktionen erweitert werden, so dass ein adäquater Umgang mit den Multimedia-Daten möglich ist. Grundlegende multimediaspezifische Funktionen wurden bereits in den vorhergehenden Abschnitten vorgestellt.

Des Weiteren ergibt sich auf Grund der Möglichkeit zur Formulierung unscharfer Anfragen der Bedarf einer iterativen Anfrageverfeinerung. Dieser muss durch entsprechende Sprachmittel einer Multimedia-Anfragesprache abgedeckt werden. Ebenso müssen für die Gewichtung und Formulierung von Präferenzen Sprachkonstrukte bereitgestellt werden.

Letztlich muss eine Parametrisierung der Ergebnispräsentation gewährleistet werden. Dazu zählt die Angabe von räumlichen und zeitlichen Beziehungen bezüglich der Präsentation der Medien-Objekte aber auch weiterer Parameter, beispielsweise zur Beschreibung der Qualität der wiederzugebenden Medien-Objekte. Zusammenfassend muss eine Multimedia-Anfragesprache somit

1. die allgemeinen Anforderungen an eine Anfragesprache erfüllen und insbesondere diverse Medientypen unterstützen,
2. scharfe, inhaltsbasierte, unscharfe, zeitliche und räumliche Prädikate und Funktionen für die Medientypen Bild, Video, Audio und Text bereitstellen,
3. über Sprachkonstrukte zur Formulierung von Präferenzen verfügen,

4. eine iterative Anfrageverfeinerung unterstützen und

5. eine multimediale Ergebnispräsentation gewährleisten sowie für die Parametrisierung dieser entsprechende Sprachmittel bereitstellen.

Die Benutzeroberfläche muss so gestaltet werden, dass eine ad-hoc-Formulierung von Multimedia-Anfragen, eine Anfrageverfeinerung sowie die räumliche und zeitliche Präsentation der Ergebnisse gewährleistet wird. Spezielle Multimedia-Unterstützungen bei der Formulierung von Anfragen, wie beispielsweise das Skizzieren von Bildern oder das Auswählen einer bestimmten Textur oder Farbe, können dabei in die Benutzeroberfläche integriert werden.

3.3 Bestehende Ansätze und Erweiterungen

Derzeitig gibt es eine Reihe von Multimedia-Anfragesprachen, die versuchen, im Bereich des Multimedia-Retrievals Fuss zu fassen. Neben diesen Multimedia-Anfragesprachen existieren vielfältige Arbeiten, die sich die Einbindung von Nutzerpräferenzen beziehungsweise die Unterstützung spezifischer Multimedia-Funktionalitäten zum Ziel gesetzt haben. So gibt es verschiedene multimediaspezifische Erweiterungen von SQL. Diese fokussieren oftmals nur eine bestimmte Multimedia-Funktionalität, so dass sie nicht als eigenständige Multimedia-Anfragesprachen betrachtet werden können. Darüber hinaus gibt es einige Ansätze, die die relationale Algebra beziehungsweise das Bereichskalkül um Mechanismen zum Gewichten von Anfragetermen und zur Formulierung von Präferenzen erweitern.

In diesem Abschnitt steht die Untersuchung der Möglichkeiten zur Einbindung von Nutzerpräferenzen und der Bereitstellung von Multimedia-Funktionalitäten in den verschiedenen Ansätzen entsprechend dieser vier Kategorien, wie in Abbildung 3.8 dargestellt, im Vordergrund.

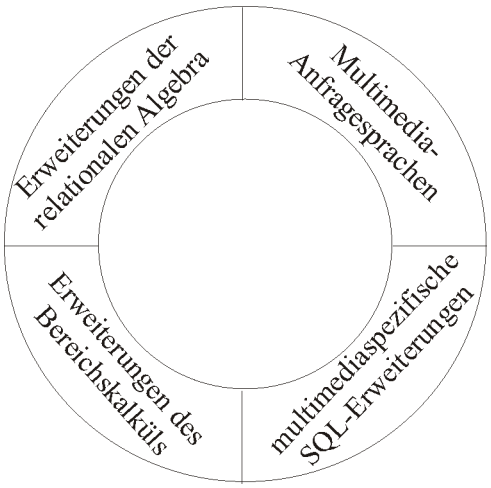


Abbildung 3.8: Kategorien bestehender Ansätze zur Einbindung von Nutzerpräferenzen und multimediaspezifischen Funktionalitäten

Zunächst wird auf die verschiedenen Multimedia-Anfragesprachen eingegangen. Dem schließt sich eine Beschreibung der Erweiterung des Bereichskalküls sowie die Vorstellung verschiedener Ansätze zur Einbindung von Nutzerpräferenzen in die relationale Algebra an. Letztlich werden einige multimediaspezifische SQL-Erweiterungen vorgestellt.

3.3.1 Multimedia-Anfragesprachen

Im Folgenden werden bestehende Multimedia-Anfragesprachen kurz vorgestellt. Dabei handelt es sich entweder um neuentwickelte Sprachen oder um Erweiterungen klassischer Anfragesprachen, wie SQL und QOL.

Paskamp (1999) hat verschiedene Multimedia-Anfragesprachen analysiert, verglichen und bewertet. Grundlage für die Evaluierung der Sprachen bilden die in Abschnitt 3.1.1 aufgeführten allgemeinen Kriterien für eine Anfragesprache. Ferner wurde evaluiert, inwieweit die Multimedia-Anfragesprachen inhaltsbasierte, räumliche, zeitliche und unscharfe Anfragen unterstützen und Konstrukte für die Parametrisierung der Ergebnispräsentation bereitstellen.

Da im Rahmen dieser Arbeit die Einbindung von Nutzerpräferenzen in eine Multimedia-Anfragesprache den Schwerpunkt bildet, ist es insbesondere von Interesse festzustellen, inwiefern bestehende Anfragesprachen eine Spezifikation von Nutzerpräferenzen unterstützen. Aus diesem Grund wird vordergründig auf die Evaluierung der Sprachen bezüglich der multimediaspezifischen Kriterien eingegangen. Die Möglichkeit der Relevanzgewichtung von Anfragetermen wurde von Paskamp im Zusammenhang mit der Möglichkeit der Formulierung unscharfer Anfragen untersucht.

Für eine ausführliche Darstellung sei an dieser Stelle auf die aufgeführten Originalquellen beziehungsweise auf die Arbeit von Paskamp (1999) verwiesen, in der ein Großteil der hier aufgeführten Multimedia-Anfragesprachen beschrieben und evaluiert wurde.

PICQUERY+ erweitert PICQUERY und ist eine wissensbasierte, objektorientierte Anfragesprache (Joseph und Cardenas, 1993; Cardenas u. a., 1993). Dabei handelt es sich um eine einfache, umgangsprachliche Anfragesprache, die für Bild- und Textdatenbanken entworfen wurde. Sie unterstützt sowohl zeitliche, räumliche als auch inhaltsbasierte Anfragen.

TVQL ist eine visuelle Anfragesprache für Video-Beschreibungsdaten (Hibino und Rundensteiner, 1996). Dabei wird die Analyse der zeitlichen und räumlichen Charakteristika der Videodaten unterstützt. Eine Ableitung von zeitlichen Trends aus den Beschreibungsdaten ist ebenso möglich wie eine Anfrageverfeinerung.

SCORE unterstützt ein reines Bildretrieval, so dass keine zeitlichen, sondern nur räumliche und scharfe Anfragen ermöglicht werden (Aslandogen u. a., 1995). Daneben wird eine iterative Anfrageverfeinerung angeboten.

SMDS wurde von Marcus und Subrahmanian (1996) entwickelt und ist eine logikbasierte Anfragesprache. Durch die Sprache können bei entsprechender Spezifikation von räumlichen und zeitlichen Beziehungen sowie einem Ähnlichkeitsmaß für die diversen Medien auch zeitliche, räumliche und inhaltsbasierte Anfragen formuliert werden. Eine Verfeinerung der Anfrage ist möglich.

SMDS-SQL basiert auf der logikbasierten Sprache SMDS, welche nun zur Optimierung der Anfragen genutzt wird (Marcus, 1996). Für die Anfrageformulierung werden die Sprachkonstrukte von SMDS auf SQL abgebildet. Der Funktionsumfang von SMDS-SQL entspricht damit dem von SMDS.

MMQSL wurde von Hirzalla und Karmouch (1996) entwickelt. Das Modell der Multimedia Query Specification Language beruht auf der Zerlegung der Multimedia-Objekte in Segmente. Somit können zeitliche, räumliche sowie inhaltsbasierte Anfragen medienübergreifend formuliert werden. Unscharfe Prädikate werden von MMQSL nicht unterstützt.

KEQL ist der Anfragesprache PICQUERY+ sehr ähnlich. So werden auch hier nur Bilddaten sowie die dazugehörigen textuellen Bildbeschreibungen unterstützt (Chu u. a., 1998). Die Anfragesprache ermöglicht inhaltsbasierte und räumliche Anfragen und bietet eine Anfrageverfeinerung an.

MQuery basiert auf KEQL und setzt diese grafisch um (Dionisio und Cárdenas, 1996). Somit können alle KEQL-Anfragen mittels einer grafischen Benutzeroberfläche auch als visuelle Anfragen formuliert werden. MQuery verfügt über denselben Funktionsumfang wie KEQL und beruht auf demselben Datenmodell.

CSQL erlaubt Anfragen an Bilddaten und die dazugehörigen Beschreibungen und wurde für das Bildretrievalsystem SEMCOG entwickelt (Li und Candan, 1998). Prinzipiell läßt die Architektur des System eine Erweiterung um andere Medientypen zu. Die Bilder in SEMCOG werden in einzelne Segmente zerlegt und in Form einer Hierarchie organisiert. Auf Basis einer Begriffshierarchie ist es möglich, Anfragen abzuschwächen oder zu verschärfen. CSQL unterstützt inhaltsbasierte, strukturbasierte und räumliche Anfragen. Die Anfrageformulierung erfolgt in SEMCOG jedoch über eine grafische Benutzeroberfläche, so dass CSQL nur für die interne Repräsentation der Anfragen genutzt wird.

SQL/MM ist ein mehrteiliger Standard, der auf SQL3 aufsetzt (ANSI, 1999, 2000, 2001, 2002). SQL/MM bietet verschiedene Module sowohl für die Volltextsuche und Textverarbeitung als auch für die Verwaltung von Multimedia-Objekten an. Es werden Klassenbibliotheken bereitgestellt, die häufig genutzte Datentypen unterstützen. So ist beispielsweise der abstrakte Datentyp *StillImage* definiert. Darüber hinaus gibt es Datentypen für eine Volltextsuche und räumliche Anfragen.

MADT steht für *medienspezifischer abstrakter Datentyp* und ist keine Anfragesprache im eigentliche Sinne, sondern vielmehr eine Bezeichnung von Modulen für die Erweiterung von kommerziellen DBMS um Multimedia-Funktionalität. Beispielsweise bietet das DBMS von Oracle9i das *interMedia-Paket* an, welches das Retrieval von Text-, Video-, Audio- und Bildobjekten ermöglicht (Oracle, 2003). Ähnlich vielfältig sind auch die *DataBlades* von Informix (Informix, 2003). Die Anfragesprache SQL wurde unter Verwendung von Konstrukten der SQL/MM entsprechend angepaßt. Sowohl das *interMedia*-Paket als auch das *Image DataBlade* (Informix, 1999) erlauben im Zusammenhang mit einem inhaltsbasierten Bild-Retrieval eine Relevanzgewichtung einiger vordefinierter Ähnlichkeitsprädikate und bieten zusätzlich eine Schwellwertgewichtung an.

VideoSQL ist eine objektorientierte Anfragesprache des Video-Information-Retrieval-Systems OVID (Oomoto und Tanaka, 1993). Dabei werden Anfragen auf Videosequenzen und -objekten, jedoch nicht auf Audio- oder Textobjekten unterstützt. Die Anfrageformulierung bei VideoSQL ist stark eingeschränkt. Zeitliche oder räumliche Anfragen können daher nur sehr umständlich formuliert werden. Inhaltsbasierte Anfragen werden nicht unterstützt.

MMSQL wurde von Amato u. a. (1998) entwickelt. Es basiert auf OQL und verfügt über die Standardfunktionalitäten einer objektorientierten Anfragesprache. Die Sprache enthält Erweiterungen zur Handhabung von vagen Informationen und erlaubt eine Relevanzgewichtung von Anfragetermen. Wesentlich ist, dass das zu Grunde liegende Datenmodell anwendungsspezifisch erweitert werden kann, so dass zeitliche, inhaltsbasierte, unscharfe und räumliche Anfragen formuliert werden können. Obwohl MMSQL auf einem Präsentationsmodell aufsetzt, so dass Bedingungen für die Präsentation der Medien-Objekte abgebildet werden können, bietet die Sprache selbst keine Konstrukte zur Parametrisierung der Darstellung von Medien-Objekten an.

MOQL erweitert OQL um räumliche und zeitliche Sprachmittel (Li u. a., 1996). Damit können zeitliche, räumliche und inhaltsbasierte Anfragen medienübergreifend formuliert werden. Ferner stehen Sprachkonstrukte bereit, die eine Parametrisierung der Ergebnispräsentation erlauben.

VisualMOQL aufbauend auf MOQL wurde die grafische Anfragesprache VisualMOQL entwickelt (Oria u. a., 1999). VisualMOQL ist derzeitig auf Bildanfragen beschränkt, so dass die räumlichen Prädikate und Funktionen denen von MOQL entsprechen. Ebenso ist das zu Grunde liegende Datenmodell dasselbe.

POQL^{MM} erweitert die Anfragesprache POQL (Henrich, 1995), welche auf OQL basiert, um Multimedia-Funktionalitäten (Henrich und Robbert, 2001). Sie unterstützt die Suche in strukturierten Dokumenten und kombiniert daher Methoden des Text-Retrievals, des inhaltsbasierten Retrievals sowie des traditionellen Daten-Retrievals. POQL^{MM} wurde als allgemeine Anfragesprache entworfen, so dass diverse Methoden zur Feature-Extraktion und verschiedene Metriken für die Ähnlichkeitsberechnung unterstützt werden. Um die Zusammenhänge zwischen den einzelnen Segmenten eines strukturierten Dokumentes auszudrücken, werden reguläre Pfadausdrücke eingesetzt. Eine Gewichtung der Anfragekriterien bezüglich ihrer Relevanz ist möglich.

FOQL wurde von Nepal u. a. (1999) als Erweiterung von OQL für das Retrieval in Bilddatenbanken entworfen. FOQL ist eine Fuzzy-Anfragesprache, da sowohl die Anfragespezifkation als auch die Ergebnisse unscharf sind. Die Sprache zeichnet sich dadurch aus, dass sie neben einem inhaltsbasierten Retrieval auch ein konzeptbasiertes Retrieval unterstützt. Es wird sowohl eine Anfrageverfeinerung, eine Relevanzgewichtung als auch eine Schwellwertgewichtung angeboten. Zeitliche und räumliche Anfragen können bei entsprechender Definition der Prädikate und Funktionen unterstützt werden.

Entsprechend Paskamps Klassifikation von Multimedia-Anfragesprachen in textbasierte und visuelle Neuentwicklungen sowie in Erweiterungen klassischer Anfragesprachen, wie

SQL und QOL, können die bestehenden Sprachen, wie in Abbildung 3.9 dargestellt, eingeordnet werden.

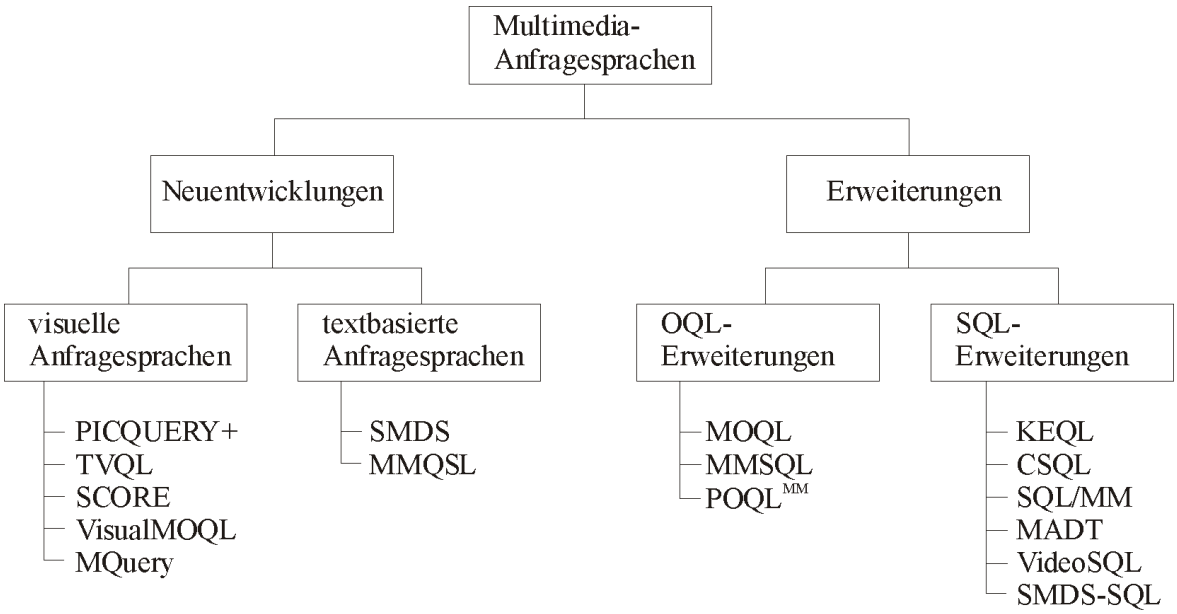


Abbildung 3.9: Einordnung der Multimedia-Anfragesprachen

Bezüglich der Erfüllung der allgemeinen Kriterien hat Paskamp festgestellt, dass bei den SQL/OQL-basierten Multimedia-Anfragesprachen die wesentlichen Eigenschaften bereits durch die zu Grunde liegende Standardanfragesprache vorgegeben sind. Jedoch bestehen Defizite insbesondere in Hinblick auf die Orthogonalität, die Eingeschränktheit und die Anwendungsunabhängigkeit (Paskamp, 1999, S. 94ff). Oftmals damit verbunden sind Auswirkungen auf die Effizienz, Optimierbarkeit und Sicherheit der Anfragesprachen.

Nur ein Teil der Anfragesprachen unterstützt die verschiedenen Medientypen, wie Audio, Video, Text und Bild, so dass die allgemeine Forderung nach Anwendungsunabhängigkeit nur von den Anfragesprachen MOQL, PQOL^{MM}, MMSQL, SMDS/SMDS-SQL, MMQSL sowie von den MADTs gewährleistet wird. Tabelle 3.2 gibt einen Überblick darüber, welche Medientypen von den einzelnen Anfragesprachen unterstützt werden. Das Symbol $\surd$ zeigt an, dass der Medientyp unterstützt wird und dementsprechend bedeutet $-$, dass der Medientyp nicht unterstützt wird.

Bei den neuentwickelten Anfragesprachen bemängelt Paskamp die fehlende formale Semantik. Mit der Semantikfestlegung einher geht die Forderung nach der Optimierbarkeit, Vollständigkeit und Sicherheit der Sprachen. Als einzige Neuentwicklung weist die Sprache SMDS eine formale Semantik auf, denn die logikbasierte Anfragesprache selbst stellt eine Art Anfragekalkül dar. Ferner kann SMDS, bis auf die Orthogonalität, alle allgemeinen Anforderungen an eine Anfragesprache erfüllen (Paskamp, 1999, S. 95f).

Bezüglich der multimediaspezifischen Kriterien zeigt sich, dass alle Anfragesprachen die Formulierung von klassischen Anfragen auf den inhaltsbeschreibenden beziehungsweise inhaltsunabhängigen Metadaten ermöglichen. Damit kann eine Suche anhand der Struktur der

Anfragesprache	Text	Bild	Audio	Video
PICQUERY+	√	√	–	–
SCORE	–	√	–	–
TVQL	–	–	–	√
SMDS/SMDS-SQL	√	√	√	√
MMQSL	√	√	√	√
KEQL/MQuery	√	√	–	–
CSQL	√	√	–	–
MADT	√	√	√	√
VideoSQL	–	–	–	√
MMSQL	√	√	√	√
MOQL	√	√	√	√
VisualMOQL	–	√	–	–
$POQL^{MM}$	√	√	√	√
FOQL	√	√	–	–

Tabelle 3.2: Überblick über die Unterstützung von Medientypen

Medien-Objekte erfolgen. Bis auf SCORE, TVSQL und VideoSQL unterstützen alle Anfragesprachen auch inhaltsbasierte Anfragen.

Neben der Formulierung von inhaltsbasierten Ähnlichkeitsanfragen auf Basis der Feature-Werte eines Objektes muss auch die Formulierung von unscharfen Anfragen durch entsprechende Fuzzy-Prädikate unterstützt werden. Dies ist jedoch nur in den Anfragesprachen PICQUERY+, SMDS/SMDS-SQL, KEQL/MQuery, FOQL sowie MMSQL möglich.

Des Weiteren werden auch räumliche und zeitliche Anfragen von fast allen Multimedia-Anfragesprachen angeboten. Die Sprachen SCORE und CSQL, die auf nicht-kontinuierliche Datentypen beschränkt sind, bieten keine Sprachkonstrukte für die Formulierung zeitlicher Anfragen an. Alle andere Sprachen bieten entsprechende zeitliche Prädikate an. Unterschiede zwischen den Anfragesprachen bestehen jedoch bezüglich der angebotenen zeitlichen Funktionen.

Für die Formulierung räumlicher Anfragen bieten alle untersuchten Anfragesprachen entsprechende Prädikate zur Abbildung von Richtungsbeziehungen an und bis auf TVSQL, CSQL und MMQSL auch von topologischen Beziehungen. Dabei ist anzumerken, dass der Umfang der angebotenen Prädikate und Operationen stark variiert. Eine besonders gute Unterstützung von räumlichen Anfragen bieten MMSQL, MOQL und die MADTs, da sie neben Richtungs- und topologischen Beziehungen auch räumliche Mengenoperationen und Aggregatfunktionen bereitstellen (Paskamp, 1999, S. 99).

Tabelle 3.3 fasst die Untersuchungen der Anfragesprachen bezüglich der Unterstützung von klassischen Anfragen auf inhaltsunabhängigen und -beschreibenden Metadaten, Ähnlichkeitsanfragen, unscharfen Prädikaten sowie von räumlichen und zeitlichen Anfragen zusammen. Die Symbole √ beziehungsweise – zeigen an, ob diese Art von Anfragen unterstützt beziehungsweise nicht unterstützt werden.

Eine wichtige Eigenschaft von Multimedia-Anfragesprachen stellt die Bereitstellung von Sprachkonstrukten zur Abbildung der Nutzerpräferenzen dar. Bezüglich dieses Kriteriums

Anfragesprache	klassische Anfragen	Ähnlichkeits-anfragen	Fuzzy-Anfragen	räumliche Anfragen	zeitliche Anfragen
PICQUERY+	√	√	√	√	√
SCORE	√	–	–	√	–
TVQL	√	–	–	√	√
SMDS/SMDS-SQL	√	√	√	√	√
MMQSL	√	–	–	√	√
KEQL/MQuery	√	√	√	√	√
CSQL	√	√	–	√	–
MADT	√	√	–	√	√
VideoSQL	√	–	–	√	√
MMSQL	√	√	√	√	√
MOQL/VisualMOQL	√	√	–	√	√
POQL^{MM}	√	√	–	–	–
FOQL	√	√	√	√	√

Tabelle 3.3: Überblick über die Unterstützung der verschiedenen Multimedia-Anfragearten

weisen die untersuchten Multimedia-Anfragesprachen erhebliche Defizite auf. So wird eine Gewichtung der Anfrageterme bezüglich der Relevanz nur von den Sprachen MMSQL, POQL^{MM} und den MADTs angeboten. Ferner erlauben nicht alle Anfragesprachen eine Verfeinerung der Anfrage. Die Anfragesprache FOQL zeichnet sich dadurch aus, dass sie neben Relevanzgewichten und einer iterativen Anfrageverfeinerung auch eine Schwellwertgewichtung integriert. Die Anfragesprachen PICQUERY+, MMQSL, VideoSQL, MOQL und VisualMOQL bieten keinerlei Möglichkeiten zur Einbindung der Nutzerpräferenzen an. Neben der Relevanz- und Schwellwertgewichtung sowie der Anfrageverfeinerung werden keine anderen Möglichkeiten zur Spezifikation von Nutzerpräferenzen bereitgestellt. Die Abbildung 3.10 zeigt, welche Anfragesprachen Nutzerpräferenzen berücksichtigen.

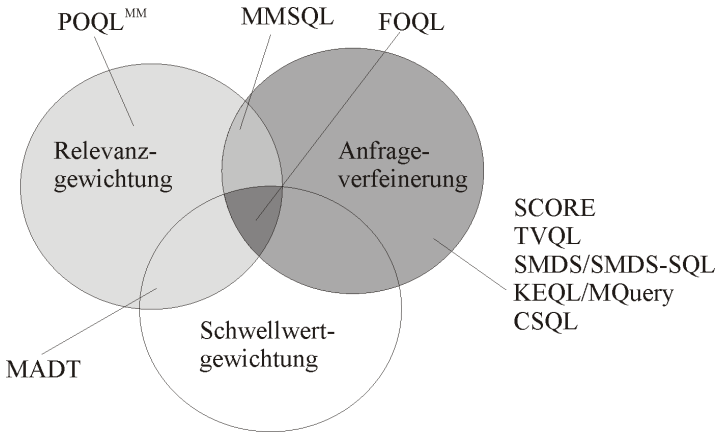


Abbildung 3.10: Überblick über die Unterstützung von Nutzerpräferenzen

Bei der Formulierung von Anfragen müssen Multimedia-Anfragesprachen Konstrukte für die Parametrisierung der Ergebnispräsentation bereitstellen. Die untersuchten Anfragesprachen unterschieden sich dabei sehr in den angebotenen Möglichkeiten. Die meisten Anfragesprachen, wie MMSQL, SMDS/SMDS-SQL, MMQSL, FOQL und POQL^{MM}, bieten keine entsprechenden Sprachmittel an. Alle anderen Multimedia-Anfragesprachen bieten zum Teil eine räumliche oder zeitliche Parametrisierung der Ausgabe und Transformationsfunktionen an (Paskamp, 1999, S. 100).

Zusammenfassend ist festzustellen, dass bereits eine Vielzahl von Multimedia-Anfragesprachen existieren, die jedoch auf Grund einer fehlenden Unterstützung grundlegender Medientypen und der mangelnden Erweiterbarkeit nur bedingt als Anfragesprachen in MIRS geeignet sind. Entsprechend ihrer anwendungsspezifischen Charakteristika können sie jedoch beispielsweise als Bild- oder Videoanfragesprachen in entsprechenden Systemen gut eingesetzt werden. Die anwendungsunabhängigen Anfragesprachen, wie MMQSL, MOQL, MMSQL, SMDS/SMDS-SQL und POQL^{MM}, sind prinzipiell als Multimedia-Anfragesprachen geeignet. Dennoch weisen sie insbesondere im Hinblick auf die Formulierung von Nutzerpräferenzen, die Parametrisierung der Ergebnispräsentation und in der Handhabung von Fuzzy-Prädikaten Mängel auf.

3.3.2 Erweiterungen des Bereichskalküls

Die grundlegende Idee von Präferenzanfragen wurde bereits in Abschnitt 2.4.1 dargelegt. In diesem Abschnitt wird die formale Semantik dieser Anfragen beschrieben.

Lacroix und Lavency (1987) entwickelten eine Anfragesprache für die Formulierung von Booleschen Präferenzenanfragen, welche auf dem klassischen Bereichskalkül (Codd, 1972) basiert. Die Anfragen werden jedoch nicht direkt im Kalkül formuliert sondern vielmehr in einer SQL-ähnliche Anfragesprache. Es werden einfache Präferenzbedingungen, wie in Beispiel 3.11 dargestellt, sowie komplexe Präferenzbedingungen unterstützt. Neben der Schachtelung von Präferenzprädikaten (Beispiel 3.12) ist auch eine gleichwertige Behandlung (Beispiel 3.13) dieser möglich. Darüber hinaus ist der Einsatz von Aggregationsfunktionen gestattet (Beispiel 3.14).

Beispiel 3.11 *Suche die Typen der Flugzeuge vom Hersteller Airbus und bevorzuge diejenigen, die weniger als 100000 Euro kosten.*

```
select the Typ of Flugzeug
  having Hersteller = 'Airbus'
from which
  prefer those having Preis < 100000
```

Beispiel 3.12 *Suche die Typen der Flugzeuge vom Hersteller Airbus und bevorzuge diejenigen, die weniger als 100000 Euro kosten. Davon wiederum bevorzuge diejenigen, die vom Baujahr 2003 sind.*

```
select the Typ of Flugzeug
  having Hersteller = 'Airbus'
from which
  prefer those having Preis < 100000
```

```
from which
 prefer those having Baujahr = 2003
```

Beispiel 3.13 *Suche die Typen der Flugzeuge vom Hersteller Airbus und bevorzuge diejenigen, die weniger als 100000 Euro kosten oder vom Baujahr 2003 sind.*

```
select the Typ of Flugzeug
 having Hersteller = 'Airbus'
from which
 prefer those having Preis < 100000
 prefer those having Baujahr = 2003
```

Beispiel 3.14 *Suche die Typen der Flugzeuge vom Hersteller Airbus und bevorzuge diejenigen, mit einem maximalen Preis.*

```
select the Typ of Flugzeug
 having Hersteller = 'Airbus'
from which
 prefer those having a maximum Preis
```

Für jeden dieser Anfragetypen geben Lacroix und Lavency (1987) an, wie die Überführung in einen Ausdruck des Bereichskalküls erfolgt. Die Präferenzanfragen in deduktiven Datenbanken, wie sie von Govindarajan u. a. (2000) vorgestellt werden, weisen dieselbe Semantik auf. Beispielsweise ergibt sich für eine einfache Präferenzanfrage der Form `select x having Q from which prefer P1` (vgl. Beispiel 3.11) im Bereichskalkül nach Lacroix und Lavency (1987) der Ausdruck

$$\{x \mid Q(x) \wedge (\exists y Q(y) \wedge P1(y) \Rightarrow P1(x))\}.$$

Eine geschachtelte Präferenzanfrage der Form `select x having Q from which prefer P1 from which prefer P2` (vgl. Beispiel 3.12) entspricht dem Bereichskalkülausdruck

$$\begin{aligned}\{x \mid\ & Q(x) \wedge \\ & (\exists y Q(y) \wedge P1(y) \wedge P2(y) \Rightarrow P1(x) \wedge P2(x)) \wedge \\ & (\neg\exists y(Q(y) \wedge P1(y) \wedge P2(y)) \wedge \exists y(Q(y) \wedge P1(y)) \Rightarrow P1(x)) \wedge \\ & (\neg\exists y(Q(y) \wedge P1(y) \wedge P2(y)) \wedge \neg\exists y(Q(y) \wedge P1(y)) \wedge \exists y(Q(y) \wedge P2(y)) \Rightarrow P2(x))\}.\end{aligned}$$

3.3.3 Erweiterungen der relationalen Algebra

Adali u. a. (1998) entwickelten eine *Multi-Similarity Algebra*, die jedoch auf einer abstrakten Ebene angesiedelt ist. Diese stellt keine operationale Multimedia-Algebra dar, die die Formulierung, Optimierung und Evaluierung von Multimedia-Anfragen unterstützt, indem

Ähnlichkeits- und klassische relationale Operatoren auf der Implementierungsebene kombiniert werden (Atnafu u. a., 2001). Im Folgenden werden daher konkrete operationale Erweiterungen der relationalen Algebra vorgestellt. Diese umfassen zum Teil die Entwicklung neuer multimediaspezifischer Algebraoperatoren und spezifizieren entsprechende Regeln für die Optimierung.

SAME^w Ansatz

In dem von Montesi und Penzo vorgestellten Ansatz einer Fuzzy-Anfragesprache für Multimedia-Daten werden die klassischen Operatoren der relationalen Algebra derart modifiziert, dass eine Evaluierung von unscharfen Anfragen unterstützt wird (Montesi und Penzo, 2000; Ciaccia u. a., 2000). So basiert die Selektion nicht mehr auf einer Booleschen Verknüpfung sondern auf einer Verknüpfung von Fuzzy-Mengen. Für diesen Zweck werden entsprechende T-Normen beziehungsweise T-Conormen aus der Fuzzy-Logik eingesetzt. Bei der Projektion ist auf Grund der Relevanzwerte eine Modifikation bei der Duplikateliminierung notwendig. So wird bei gleichen Tupeln, für die jedoch unterschiedliche Relevanzwerte berechnet wurden, das Tupel mit dem höchsten RSV gewählt. Es erfolgt also eine Disjunktion der einzelnen Relevanzwerte gleicher Tupel. Ähnlich wie bei der Selektion müssen auch bei den Mengenoperationen und dem Join-Operator Fuzzy-Mengen verknüpft werden. Auch für diese Zwecke werden T-Normen und T-Conormen eingesetzt.

Bedeutend ist, dass nicht nur unscharfe Anfragen sondern auch gewichtete Anfragen unterstützt werden. So können auch Relevanzgewichte den einzelnen Operanden zugewiesen werden. So ist es beispielsweise möglich, die einzelnen Bedingungen einer Selektion oder die zu verknüpfenden Mengen bei einer Vereinigung zu gewichten. Die gewichteten Anfragen werden durch Einsatz der Gewichtungsformel (Fagin und Wimmers, 1997), auf die im Abschnitt 4.2.1 noch näher eingegangen wird, evaluiert.

Neben der Anpassung der bestehenden Operationen der relationalen Algebra zur Handhabung von unscharfen Anfragen und Fuzzy-Mengen werden weitere Operatoren eingeführt. Dazu zählen der *top*- und der *cut*-Operator.

Der *top*-Operator τ ermittelt die ersten k Tupel t einer Relation r bezüglich einer Sortierfunktion g. Wird eine Gewichtung Θ in das Ranking mit einbezogen, so ergibt sich die Sortierfunktion g_{Θ_g}. Hat die Relation nicht mehr als k Tupel so gilt:

$$\tau^k_{g_{\Theta_g}}(r) = r.$$

Ansonsten ist der *top*-Operator wie folgt definiert:

$$\tau^k_{g_{\Theta_g}}(r) = \{t \quad | \quad t \in r \wedge R(g_{\Theta_g}, t) > 0 \wedge |\tau^k_{g_{\Theta_g}}(r)| = k \wedge \forall t \in \tau^k_{g_{\Theta_g}}(r) : \\ \neg\exists t' : t' \in r \wedge t' \notin \tau^k_{g_{\Theta_g}}(r) \wedge g_{\Theta_g}(t') > g_{\Theta_g}(t)\},$$

wobei R eine Funktion ist, die für ein Tupel t unter Beachtung der Bedingung g_{Θ_g} einen Wert in [0,1] ermittelt (Montesi und Penzo, 2000; Ciaccia u. a., 2000). Wird kein Sortierkriterium angegeben, so wird der Relevanzwert eines Tupels als Kriterium herangezogen. Es werden dann die k-ersten Tupel mit den höchsten Relevanzwerten ermittelt.

Der *cut*-Operator γ wählt die Tupel aus, die eine angegebene Formel g erfüllen. Es werden also nicht die Relevanzwerte beeinflusst, sondern lediglich die Tupel ermittelt, für die die

Formel g einen Wert größer *null* liefert. Der Relevanzwert μ für ein Tupel t entspricht dem ursprünglichen Relevanzwert μ_E des Tupels. Der *cut*-Operator ist definiert als:

$$\gamma_g(r) = \{t | t \in r \wedge R(g,t) > 0 \wedge t.\mu = t.\mu_E > 0\}.$$

Wie Montesi und Penzo feststellen, sind die klassischen Optimierungsregeln mit Einführung der neuen Operatoren *cut* und *top* und der verwendeten Semantik auf Basis von Fuzzy-Logik nicht mehr anwendbar. Beispielsweise wird die Distributivität in einer Selektion nur für ungewichtete Bedingungen, deren Evaluierung auf den klassischen Normen *min* und *max* beruht, gewährleistet. Aus diesem Grund wurden entsprechende Optimierungsregeln für die klassischen Operatoren und für *cut* und *top* erarbeitet (Montesi und Penzo, 2000).

Winnow-Operator und *Best*-Operator

Mit Hilfe des *winnow*-Operators $w_C(R)$ von Chomicki (2002) können Präferenzanfragen in der relationalen Algebra umgesetzt werden. In Abschnitt 2.4.1 wurde die Funktionsweise dieses Operators bereits kurz vorgestellt. Im Folgenden soll näher auf die formale Definition des Operators eingegangen werden. Dabei wird angenommen, dass D die Domäne der Konstanten und N die der Zahlen ist.

Eine Präferenzrelation $\succ$ über $U = (U_1, \ldots, U_n)$ mit $i = 1, \ldots, n$ und $U_i \in D \cup N$ ist eine Teilmenge von $(U_1 \times \ldots \times U_n) \times (U_1 \times \ldots \times U_n)$. Eine Präferenzformel $C(t_1, t_2)$ über die Tupel t_1 und t_2 entspricht einer prädikatenlogischen Formel, die sich in disjunktiver Normalform befindet und quantorfrei ist, so dass gilt:

$$t_1 \succ t_2 \Leftrightarrow C(t_1, t_2).$$

Als Vergleichsoperatoren sind die klassischen Operatoren $(=, \neq, <, \leq, >, \geq)$ zulässig. Die Definition hierarchischer Präferenzen durch eine entsprechende Präferenzformel und die Verwendung von Aggregationen ist, ähnlich wie im Ansatz von Lacroix und Lavency (1987), möglich.

Der *winnow*-Operator $w_C(R)$ selektiert die Tupel t einer Relation R, die entsprechend einer gegebenen Präferenzformel C am stärksten bevorzugt werden. Die formale Semantik ist daher für jede Instanz r von R wie folgt definiert:

$$w_C(r) = \{t \in r | \neg \exists t' : t' \in r \wedge t' \succ t\}.$$

Der Operator kann in die relationale Algebra aufgenommen werden. Eine Präferenzanfrage ist somit jede Anfrage, in welcher der *winnow*-Operator mindestens einmal genutzt wird. Durch die Bereitstellung eines eigenen Operators für Präferenzanfragen können spezielle Methoden für die Anfrageevaluierung entwickelt werden. Ferner erlaubt dies die Spezifikation von Optimierungsregeln (Chomicki, 2002).

Für die iterative Anwendung des Operators, wodurch ein Ranking auch für Boolesche Anfragen ermöglicht wird, kann der Operator für die k-te Iteration wie folgt angegeben werden:

$$\begin{aligned} w_C^1(r) &= w_C(r) \\ w_C^{k+1}(r) &= w_C(r - \bigcup_{1 \leq i \leq k} w_C^i(r)). \end{aligned}$$

Der von Torlone und Ciaccia (2002b,a) entwickelte *best*-Operator $\beta_{\succ}$ ist identisch mit der Definition des *winnow*-Operators. Für die Implementierung des *best*-Operators wird von den Autoren ein entsprechender Algorithmus angegeben.

Präferenzalgebra

In dem Ansatz von Kießling (2002) werden verschiedene Typen von Präferenzen P vorgestellt. Neben Präferenzen auf alphanumerischen Daten können durch Verwendung einer einfachen Distanzfunktion $distance(a, b) = |a - b|$ auch Präferenzen auf numerischen Daten verarbeitet werden. Beispielsweise wird bei der Präferenz $AROUND(A, z)$ auf dem Attribut A und dem Referenzwert z y vor x bevorzugt ($x <\text{P}\ y$), wenn der Abstand zwischen dem Wert y und dem Referenzwert z kleiner ist als der Abstand zwischen x und dem Referenzwert. Es gilt ($x <\text{P}\ y$) genau dann, wenn $distance(x, z) > distance(y, z)$. Des Weiteren wird eine SCORE-Präferenz $SCORE(A, f)$ bereitgestellt, die unter Beachtung der Funktion f die Relevanzwerte für jeden Wert des Attributs A berechnet. Als Funktion f kann beispielsweise auch eine Distanzfunktion verwendet werden.

Neben der Abbildung von einfachen Präferenzen zeichnet sich dieser Ansatz dadurch aus, dass verschiedene Präferenzkonstruktoren bereitgestellt werden, um komplexe Präferenzen zu formulieren. Beispielsweise wird neben einer Pareto-Kumulierung auch eine numerische Kumulierung $rank_S(P1, P2)$ der Präferenzen $P1$ und $P2$ unter Beachtung einer Scoring-Funktion S gewährleistet. Dies ist in der Kombination mit SCORE-Präferenzen sinnvoll. Damit ist es möglich, durch Verwendung einer gewichteten Scoring-Funktion eine Relevanzgewichtung auch in diesem Ansatz indirekt abzubilden.

Für die Umsetzung der Präferenzen in der Relationenalgebra wird die Selektion angepasst. Es sei R eine Relation und es wird angenommen, die Präferenz $P = (A <\text{P})$ auf den Attributen $A = \{A_1, A_2, \ldots, A_n\}$ sei gegeben, dann ergibt sich die Datenbankpräferenz $P^R = (R.A <\text{P})$. Ein Tupel erfüllt eine Anfrage dann perfekt, wenn gilt:

$$t.A \in max(P) \wedge t.A \in R.$$

Damit muss zwischen den idealen Objekten aus Nutzersicht $max(P)$ und den real vorhandenen besten Objekten $max(P^R)$ nicht zwingend eine Übereinstimmung vorhanden sein. Präferenzanfragen vollziehen daher einen Abgleich zwischen den Nutzerpräferenzen P und den gegebenen Möglichkeiten P^R. Die Selektion kann unter Beachtung der spezifizierten Präferenzen wie folgt angegeben werden:

$$\sigma[P](R) = \{t \in R | t.A \in max(P^R)\}.$$

Damit werden all die Tupel mit einem maximalen Wert von P^R ermittelt. Es werden also die besten Datenbankobjekte unter Berücksichtigung der Nutzerpräferenzen bestimmt.

Darauf aufbauend schlägt Kießling einen weiteren Selektionsoperator vor, der eine Gruppierung der Präferenzen unterstützt. Ferner werden verschiedene Optimierungsregeln für eine effiziente Umsetzung der Präferenzanfragen auf Basis eines Divide-and-Conquer-Prinzips vorgestellt (Kießling, 2002; Kießling und Hafenrichter, 2003).

Ähnlichkeitsoperatoren

Atnafu u. a. (2001) entwickeln in Analogie zu den traditionellen Datenbankoperationen einen formalen Framework für inhaltsbasierte Ähnlichkeitsoperationen. Dieser Framework enthält eine Menge von Ähnlichkeitsoperatoren für ein inhaltsbasierte Bild-Retrieval und beschreibt Regeln für eine Optimierung der Evaluierung von Ähnlichkeitsanfragen.

Grundlage für die Entwicklung der Ähnlichkeitsoperationen bildet ein objektrelationaler Ansatz. Demnach weist eine Multimedia-Tabelle $M(id, O, F, A, P)$ die folgenden Komponenten auf:

- id ist ein eindeutiger Identifikator für eine Instanz von M,
- O ist eine Referenz auf das Bildobjekt,
- F ist der Featurevektor des Objektes O,
- A ist eine Attributkomponente und
- P ist eine Datenstruktur, die Zeiger auf Instanzen einer anderen Multimedia-Tabelle verwaltet.

Auf Basis von NearstNeighbor-Anfragen werden die verschiedenen Ähnlichkeitsoperatoren umgesetzt. Damit wird für ein Objekt o mit seiner entsprechenden Featurerepräsentation und einer Tabelle M die ähnlichkeitsbasierte Selektion $\sigma_o^\epsilon(M)$ wie folgt angegeben:

$$\sigma_o^\epsilon(M) = \{(id, o', f, a, p) \in M | o' \in NN^\epsilon(M, o))\},$$

wobei $\epsilon \in \mathbb{R}^+$ und $NN^\epsilon(M, o)$ eine NearstNeighbor-Anfrage darstellt, welche die ähnlichsten Objekte bezüglich des Anfrageobjektes o und der vorgegebenen Distanz ϵ ermittelt. Für zwei Tabellen M_1 und M_2 ergibt sich der *Similarity-Based-Join* $M_1 \otimes^\epsilon M_2$ als:

$$M_1 \otimes^\epsilon M_2 = \{(id_1, o_1, f_1, a_1, p_1') \quad | \quad (id_1, o_1, f_1, a_1, p_1) \in M_1 \wedge p_1' = p_1 \cup (M_2, s_id^\epsilon(M_2, o_1)) \wedge p_1' \neq null\},$$

mit $s_id^\epsilon(M_2, o_1) = \pi_{id}(\sigma_{o_1}^\epsilon(M_2))$. Für jedes Objekt aus M_1 werden zunächst die ähnlichsten Objekte aus M_2 bestimmt. Der Verweis eines Objektes o_1 wird sodann aktualisiert und enthält die Identifikatoren auf die zu diesem Objekt ähnlichen Objekte in M_2.

Der *Similarity-Based-Join* stellt die Grundlage für alle anderen Ähnlichkeitsoperatoren dar. Nachteilig an diesem Operator ist, dass der Grad der Ähnlichkeit zwischen zwei Objekten verloren geht. Es werden also nur die ähnlichsten Objekte durch eine NearstNeighbor-Anfrage und unter Beachtung des Parameters ϵ ermittelt. Die genauen Ähnlichkeitswerte werden nicht gespeichert. Die weiteren Operatoren werden im Folgenden kurz aufgeführt:

- Der *Multi-Similarity-Based-Join* $M_1 \otimes^\epsilon M_2 \otimes^\epsilon \ldots \otimes^\epsilon M_n = \biguplus_{i<j;1\leq i,j\leq n} M_i \otimes^\epsilon M_j$ führt den *Similarity-Based-Join* paarweise aus und verknüpft dann die resultierenden Ergebnistabellen. Dabei ist die Reihenfolge der Operanden von Bedeutung.
- Der *symmetrische Similarity-Based-Join* $M_1 \oplus^\epsilon M_2 = (M_1 \otimes^\epsilon M_2) \uplus (M_2 \otimes^\epsilon M_1)$ ist kommutativ, wobei $M_1 \uplus M_2$ eine Tabelle liefert, die alle Instanzen enthält, die entweder in M_1 oder M_2 sind.

- Der *symmetrische Multi-Similarity-Based-Join* $M_1 \oplus^\epsilon M_2 \oplus^\epsilon \ldots \oplus^\epsilon M_n = \biguplus_{i<j;1\leq i,j\leq n} M_i \oplus^\epsilon M_j$, wobei die Reihenfolge der Operanden nicht von Bedeutung ist.

Für eine effiziente Umsetzung dieser Operatoren wird der *Mine*-Operator genutzt. Dieser baut die korrespondierende Ergebnistabelle für eine Operation auf. Beispielsweise nutzt $Mine(M_1 \otimes^\epsilon M_2)$ die Komponente P_1 von M_1 der durch $(M_1 \otimes^\epsilon M_2)$ erzeugten Ergebnistabelle, um die dazu korrespondierende Ergebnistabelle $(M_2 \otimes^\epsilon M_1)$ aufzubauen. Auf Basis des *Mine*-Operators und der Eigenschaften der Ähnlichkeitsoperatoren wurden von Atnafu u. a. Optimierungsstrategien entwickelt.

Zusammenfassend kann festgestellt werden, dass die Definition von Ähnlichkeitsoperatoren eine sinnvolle und notwendige Erweiterung der relationalen Algebra darstellt. Nachteilig an dem Ansatz von Atnafu u. a. ist, dass die Operationen auf Bildähnlichkeitsanfragen beschränkt sind. Ferner ist ein konkretes auf dem objektrelationalen Modell basierendes Datenmodell erforderlich, so dass die Tabellenstruktur in diesem Ansatz fest vorgegeben ist. Ein weiterer Nachteil des Ansatzes liegt in der Tatsache begründet, dass keine Relevanzwerte ermittelt werden. Damit wird ein Ranking, was charakteristisch für unscharfe Multimedia-Anfragen ist, nicht unterstützt.

3.3.4 Erweiterungen von SQL

Für die Anfragesprache SQL existieren verschiedene Erweiterungen, die im Rahmen von Multimedia-Anfragen von Interesse sind. Diese beziehen sich vornehmlich auf die Integration von Präferenzen oder erlauben die Spezifikation von Ausgabebedingungen. Im Folgenden wird auf diese Erweiterungen von SQL näher eingegangen.

Preference-SQL

Kießling und Köstler (2002) integrieren in Preference-SQL einen Teil der in Kießling (2002) eingeführten und bereits im vorhergehenden Abschnitt beschriebenen Möglichkeiten zur Spezifikation von Präferenzen. Nicht unterstützt werden SCORE-Präferenzen sowie komplexe Präferenzen unter Verwendung des $rank_S$-Operators. Damit werden nur die typischen Präferenzanfragen unterstützt. Eine Verknüpfung von Präferenzanfragen und Multimedia-Retrieval ist daher nicht möglich. In Preference-SQL wird die SELECT-Klausel wie folgt erweitert:

```
SELECT ... FROM ... WHERE ...
PREFERING ...
GROUPING ...
BUT ONLY ...
ORDER BY ...
```

In der PREFERING-Klausel können die verschiedenen Präferenzen durch den Nutzer spezifiziert werden. Dabei werden die Präferenzen nur auf die Tupel angewendet, die bereits die harten, in der WHERE-Klausel spezifizierten Bedingungen erfüllen. Der GROUPING-Operator erlaubt eine Gruppierung der weichen Bedingungen und hat damit die gleiche Aufgabe wie das traditionelle GROUP BY. Die BUT ONLY-Klausel bewirkt einen logischen Test, nachdem die Präferenzen ausgewertet wurden und erlaubt ein explizites Einschränken

der Ergebnismenge. Für eine ausführliche Syntaxbeschreibung sei auf das Preference-SQL-Manual (Dat, 2000) verwiesen.

Präferenzanfragen und *Top N*-Anfragen

Borzsonyi u. a. (2001) entwickelten für einen Spezialfall der Präferenzanfragen einen speziellen SQL-Operator, den sogenannten *Skyline*-Operator. Beispiel 3.15 zeigt eine solche Anfrage. Von Interesse für den Nutzer sind all die Flugzeuge, die in beiden Dimensionen nicht schlechter als ein anderes Flugzeug sind. Die ermittelte *Skyline* für dieses Beispiel ist in Abbildung 3.11 dargestellt. Als Ergebnis auf die Anfrage werden also all die Objekte geliefert, die auf der *Skyline*-Linie liegen. Dies entspricht damit nur einer Teilmenge der Pareto-optimalen Ergebnismenge, die in der Präferenzalgebra durch eine Pareto-Kumulierung ermittelt werden kann (Kießling, 2002).

Beispiel 3.15 *Suche die Flugzeuge vom Hersteller Airbus, die möglichst viele Sitzplätze haben und dennoch niedrig im Preis sind.*

Für die Formulierung solcher *Skyline*-Anfragen wird das folgende SQL-Statement vorgeschlagen (Borzsonyi u. a., 2001):

SELECT ... FROM ... WHERE ...
GROUP BY ... HAVING ...
SKYLINE OF [DISTINCT] d_1 [MIN | MAX | DIFF], ..., d_1 [MIN | MAX | DIFF]
ORDER BY ...

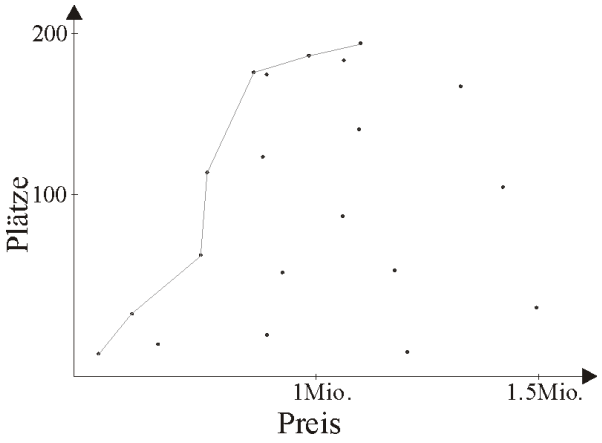


Abbildung 3.11: Sykline der Flugzeuge

Die SKYLINE-Klausel ermittelt all die Tupel, die von keinem anderen Tupel dominiert werden. Dabei dominiert ein Punkt einen anderen Punkt, wenn er in allen Dimensionen mindestens genauso gut und in mindestens einer Dimension besser ist (Borzsonyi u. a., 2001). Damit kann die Anfrage aus Beispiel 3.15, wie im folgenden Beispiel gezeigt, in SQL angegeben werden.

Beispiel 3.16
```
SELECT *
FROM Flugzeug
WHERE Hersteller = 'Airbus'
SKYLINE OF Plaetze MAX, Preis MIN
```

Die Berechnung einer Skyline basiert auf dem Maximum-Vektor-Problem. Für die Umsetzung des *Skyline*-Operators kann daher der Divide-and-Conquer Algorithmus von Kung u. a. (1975) genutzt werden. Effizientere Evaluierungsalgorithmen wurden von Borzsonyi u. a. (2001) und Tan u. a. (2001) vorgestellt.

Neben dem *Skyline*-Operator für die Formulierung spezieller Präferenzanfragen gibt es weitere SQL-Operatoren, die im Zusammenhang mit Multimedia-Anfragen nützlich sein können. Dazu zählt der *Top N*-Operator, der auch als STOP-AFTER-Operator bezeichnet wird (Carey und Kossmann, 1997, 1998). Dieser erlaubt, explizit die Kardinalität der Ergebnismenge zu reglementieren. Dies ist im Beispiel 3.17 dargestellt.

Beispiel 3.17 *Suche 21 Abbildungen von Flugzeugen, die der gegebenen Abbildung am ähnlichsten sind.*

Um eine solche Anfrage formulieren zu können, wird die SQL-Klausel STOP AFTER <value> eingeführt:

```
SELECT ... FROM ... WHERE ...
GROUP BY ... HAVING ...
ORDER BY ...
STOP AFTER <value>
```

Dabei ist <value> ein atomarer numerischer Wert, der entweder vorgegeben, berechnet oder durch eine Unterabfrage ermittelt wird. Er beschreibt die Anzahl der auszugebenden Tupel. Eine ähnliche SQL-Erweiterung wurde unter anderem von Chaudhuri und Gravano (1996) propagiert.

Die Verwendung des *Top N*-Operators innerhalb von *Skyline*-Anfragen ist dabei durchaus sinnvoll. So kann zunächst ein Gesamteindruck durch Verwendung des *Skyline*-Operators vermittelt werden, der dann durch Einsatz des *Top N*-Operators auf die wichtigsten Objekte beschränkt wird. Es wird also eine Halbordnung der N besten Objekte erstellt.

Anzumerken hierbei ist, dass eine Kombination aus *Skyline* und *Top N*-Operator andere Ergebnisse liefert als eine k-NearstNeighbor-Anfrage (kNN-Anfrage), obwohl die Kardinalität der Ergebnisse in beiden Fällen reglementiert wird.

Eine kNN-Anfrage versucht stets, k Objekte zu finden, die dem Idealobjekt am nächsten kommen. Damit werden also in Bezug auf Beispiel 3.15 die Flugzeuge mit einem möglichst kleinen Preis und vielen Sitzplätzen gesucht. Jedoch werden ganz billige Flugzeuge mit nur wenigen Plätzen nicht ermittelt. Demgegenüber liefert eine *Top N-Skyline*-Anfrage im Vergleich zu der kNN-Anfrage sowohl relevante als auch irrelevante Objekte. Dabei kann es vorkommen, dass einige relevante Objekte nicht ermittelt werden. Diese Unterschiede der beiden Anfragetypen sind in Abbildung 3.12 dargestellt.

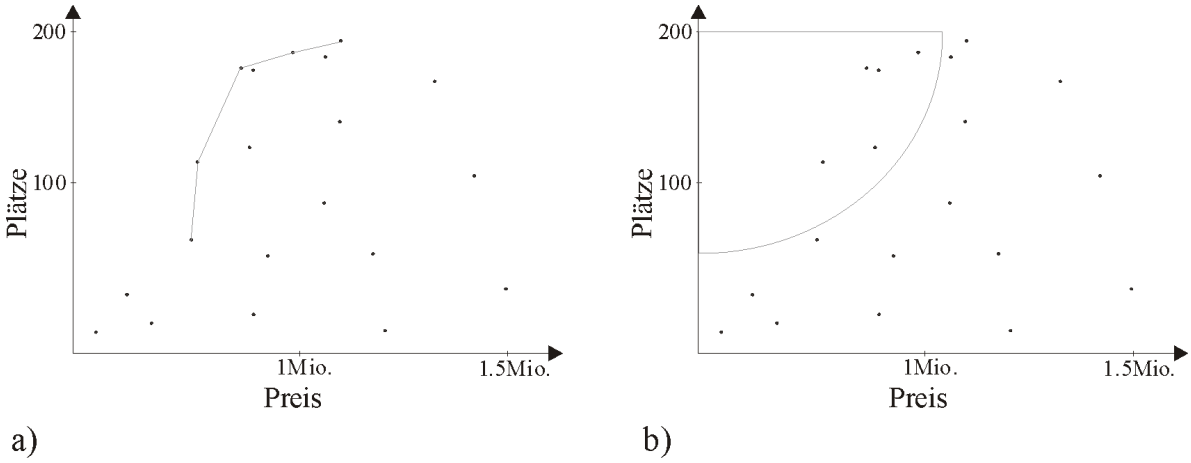


Abbildung 3.12: Ermittlung der fünf besten Flugzeuge a) durch Kombination des *Skyline*- und *Top N*-Operators (nach Sitzplätzen sortiert) und b) durch eine kNN-Anfrage mit dem Anfragepunkt (0,200)

Parametrisierung der Ergebnispräsentation mittels SQL+D

SQL+D ist eine Erweiterung von SQL, die dem Nutzer die Möglichkeit bietet, anzugeben, wie die Ergebnisse bezüglich einer Anfrage an eine Multimedia-Datenbank dargestellt werden sollen (Baral u. a., 1998b,a). Eine SQL+D Anfrage besteht aus einer normalen SQL-Anfrage und einer DISPLAY Klausel. In dieser wird die räumliche und zeitliche Darstellung der Ergebnisse spezifiziert. Darüber hinaus können auch mögliche Nutzerinteraktionen formuliert und Diagramme erstellt werden.

Die Formulierung von zeitlichen Bedingungen in der DISPLAY-Klausel wird grafisch unterstützt. So wird die Anordnung der Elemente entlang eines Zeitstrahls (vgl. Timeline-Modell in Abschnitt 3.1.3) ermöglicht. Die daraus abgeleiteten temporalen Bedingungen werden in die DISPLAY-Klausel übernommen. Eine grafisch unterstützte Spezifikation der räumlichen Bedingungen wird jedoch nicht angeboten. Dies ist als Nachteil zu werten, da der Nutzer die Vielzahl der zur Verfügung stehenden Sprachmittel kennen muss und die Spezifikation der räumlichen Anordnung der Ergebnisse aufwändig ist. Am Beispiel 3.18 (Baral u. a., 1998a) wird die Komplexität der Anfrageformulierung deutlich.

Beispiel 3.18

```
SELECT *
FROM Professors
DISPLAY PANEL main,
            PANEL A ON main.East,
            PANEL B ON A.Center,
WITH name AS list ON main.West,
     picture AS image ON A.North,
     'Education at', 'Email address' AS text ON B.West(Vertical),
     univ, email AS boxedtext ON B.East(Vertical),
     'Home Page' AS button ON B.South
     TRIGGER 'netscape('+ address +')'
```

Diese Beispielanfrage liefert das in Abbildung 3.13 dargestellte Ergebnis. Ein weiterer Nachteil von SQL+D ist, dass eine Parametrisierung der Ausgabe bezüglich der Güte der Medien-Objekte, wie beispielsweise der Größe und der Qualität eines Bildes, nicht unterstützt wird.

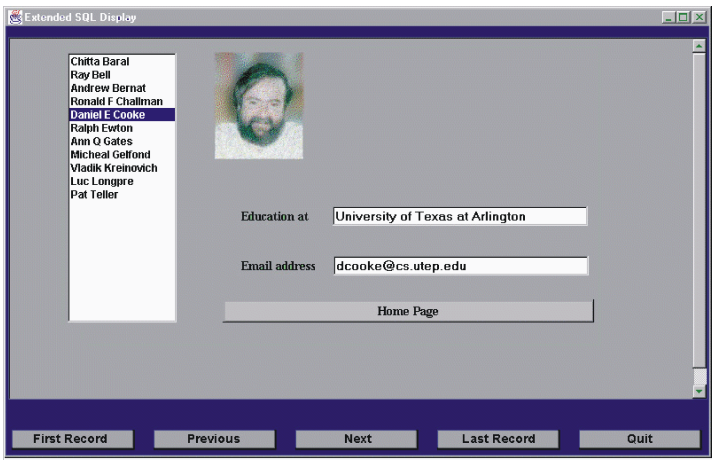


Abbildung 3.13: Ergebnis der SQL+D-Anfrage aus Beispiel 3.18 (Baral u. a., 1998a)

3.4 Zusammenfassung

In diesem Kapitel wurden zunächst die Anforderungen an eine Multimedia-Anfragesprache beschrieben. Neben den allgemeinen Anforderungen an eine Anfragesprache ergeben sich für Multimedia-Anfragesprachen auf Grund der Charakteristika des Multimedia-Retrievals zusätzliche Anforderungen. So müssen für die Ergebnispräsentation entsprechende Sprachmittel angeboten werden, die eine zeitliche, räumliche und medienspezifische Parametrisierung der Darstellung der Ergebnisobjekte unterstützen. Diese vielfältigen Möglichkeiten zur Parametrisierung der Ergebnisausgabe wurden beschrieben.

Ferner muss eine Multimedia-Anfragesprache die Formulierung von inhaltsbasierten, zeitlichen und räumlichen sowie scharfen und unscharfen Anfragen unterstützen. Des Weiteren müssen auch Präferenzen bei der Anfrageformulierung spezifiziert werden können. Diese verschiedenen Anfragetypen wurden beschrieben und es wurde dargelegt, welche speziellen Prädikate und Funktionen zur Unterstützung dieser Art von Anfragen bereitgestellt werden müssen.

Für den Entwurf einer Multimedia-Anfragesprache wurden die notwendigen Prinzipien und Anforderungen, die dabei zu berücksichtigen sind, skizziert und zusammengefasst. Auf dieser Basis wurden bestehende Multimedia-Anfragesprachen untersucht. Den Schwerpunkt dabei bildete die Evaluierung der Sprachen bezüglich der Möglichkeiten zur Abbildung von Nutzerpräferenzen. Es wurde festgestellt, dass zwar eine Vielzahl von Multimedia-Anfragesprachen existiert, die jedoch oftmals auf Grund einer fehlenden Unterstützung grundlegender Medientypen oder wegen der mangelnden Erweiterbarkeit nicht als Anfragesprachen in MIRS geeignet sind. Darüber hinaus wird sowohl eine Parametrisierung der

Ergebnispräsentation als auch die Handhabung von Fuzzy-Prädikaten nur in einem begrenzten Umfang von wenigen Multimedia-Anfragesprachen angeboten. Im Hinblick auf die Einbindung von Nutzerpräferenzen bestehen erhebliche Defizite. So wird zwar von vielen Sprachen bereits eine Anfrageverfeinerung angeboten, jedoch nur selten eine Relevanz- oder Schwellwertgewichtung. Andere Möglichkeiten der Formulierung von Nutzerpräferenzen, wie beispielsweise eine Operatorgewichtung, werden von keiner der untersuchten Multimedia-Anfragesprachen unterstützt.

Neben diesen Multimedia-Anfragesprachen existieren spezielle Erweiterungen des Bereichskalküls und der Relationenalgebra. So wurde das Bereichskalkül dahingehend modifiziert, dass Präferenzanfragen abgebildet werden können. Für die relationale Algebra gibt es diverse multimediaspezifische Erweiterungen. Beispielsweise erlaubt der *winnow*-Operator die Umsetzung von Präferenzanfragen und der *cut*-Operator ermöglicht eine Schwellwertgewichtung. Zusätzlich wurden spezielle *Multimedia-Join*-Operatoren für ein ähnlichkeitsbasiertes Bild-Retrieval eingeführt. Eine Relevanzgewichtung der Anfrageterme und die Formulierung unscharfer inhaltsbasierter Anfragen ist im SAME^w-Ansatz durch eine Modifizierung bestehender relationaler Operatoren möglich. Diesen Ansätzen ist gemein, dass jeweils sehr spezielle Anforderungen umgesetzt werden. Nachteilig ist jedoch, dass die in diesen Ansätzen vorgeschlagenen Erweiterungen nur auf Kalkül- beziehungsweise Algebraebene angeboten werden, jedoch nicht in eine nutzerfreundliche Anfragesprache integriert sind.

Eine Ausnahme stellt dabei der Vorschlag der Präferenzalgebra von Kießling (2002) dar. Diese Algebra bietet diverse Möglichkeiten zur Formulierung komplexer Präferenzanfragen an. Die Funktionalitäten der Präferenzalgebra werden jedoch nur teilweise durch Preference-SQL unterstützt. So können hier keine Präferenzen durch Scoring-Funktionen ausgedrückt werden. Damit besteht keine Möglichkeit, inhaltsbasierten Anfragen in Preference-SQL zu formulieren. Nachteilig ist weiterhin, dass die Anfragen in Preference-SQL sehr komplex werden können, so dass diese Anfragesprache daher vorwiegend für einen Experten geeignet ist.

Neben Preference-SQL existieren weiterhin verschiedene Erweiterungen von SQL, die im Zusammenhang mit Multimedia-Anfragen interessant sind. Auf diese wurden ebenfalls in diesem Kapitel eingegangen. Es wurde der *Skyline*-Operator vorgestellt. Mit diesem können spezielle Präferenzanfragen in SQL formuliert werden. Das besondere an einer *Skyline*-Anfrage ist, dass all die Objekte ermittelt werden, die von keinem anderen Objekt bezüglich der angegeben Kriterien dominiert werden. Mit Hilfe des *Top N*-Operators kann darüber hinaus die Kardinalität der Ergebnismenge beeinflusst werden. Neben diesen Möglichkeiten wurden SQL+D beschrieben. Dieser Ansatz erweitert SQL um Möglichkeiten zur Spezifikation der Ergebnisdarstellung. So können sowohl zeitliche als auch räumliche Parameter für die Anordnung der Ergebnisobjekte angegeben werden. Nachteilig an diesem Ansatz ist, dass der Aufwand des Nutzers für das Spezifizieren der räumlichen Visualisierung der Ergebnisobjekte sehr hoch ist, da dafür keine grafische Unterstützung angeboten wird.

4 Gewichtungsmodell für Multimedia-Anfragen

Wie bereits in Abschnitt 2.4 dargestellt, gibt es eine Vielzahl von Möglichkeiten, um Nutzerpräferenzen in Datenbankanfragen auszudrücken. In diesem Kapitel wird die Formulierung von Präferenzen in Form von Gewichten fokussiert. Diese Art der Spezifikation von Nutzerpräferenzen hat sich im Multimedia-Retrieval bewährt (Fagin und Wimmers, 1997; Carson u. a., 1997; Sung, 1998; Ortega u. a., 1998; Oracle, 2003). Gegenstand des Kapitels ist die Entwicklung eines adäquaten Gewichtungsmodells. Dieses soll die Abbildung und Formulierung von Präferenzen in Multimedia-Anfragen unter Verwendung verschiedener Gewichtungssemantiken erlauben.

Zunächst wird ein Anfragemodell für einfache, relevanzgewichtete Ähnlichkeitsanfragen beschrieben. Dieses Modell bildet sodann die Grundlage für die Evaluierung bestehender Systeme sowie für weitere Überlegungen bezüglich eines mächtigeren Anfragemodells.

Die Präferenzen der Nutzer werden im quantitativen Ansatz indirekt über eine *Scoring-Funktion* abgebildet. Es erfolgt daher eine Analyse der Anforderungen, die an Scoring-Funktionen gestellt werden. Der Fokus liegt dabei auf der Beschreibung geeigneter Scoring-Funktionen sowie deren Eigenschaften. Auf Basis dieser Anforderungen wird eine Evaluierung bestehender Verfahren vorgenommen.

Das Hauptaugenmerk dieses Kapitels bildet die Entwicklung eines erweiterten Gewichtungsmodells. Dieses *Multi-Level-Gewichtungsmodell* unterstützt die Relevanzgewichtung in komplexen Anfragen adäquat, indem Relevanzgewichte auf allen Ebenen in einer komplexen Anfrage spezifiziert werden können. Darüber hinaus integriert das Multi-Level-Gewichtungsmodell zusätzlich Schwellwert- und Operatorgewichte.

4.1 Basisanfragemodell

Als Ausgangsbasis für die Entwicklung eines komplexen, gewichteten Anfragemodells wird in diesem Anschnitt ein einfaches Anfragemodell vorgestellt. Hierbei wird eine Relevanzgewichtung einfacher Anfragen betrachtet. Diese Anfragen werden unter Verwendung sogenannter Scoring-Funktionen evaluiert. Auf Grund der Gewichtung werden spezielle Anforderungen an diese Funktionen gestellt. Im Folgenden wird darauf näher eingegangen.

4.1.1 Modell für einfache, gewichtete Anfragen

Eine Anfragebedingung X besteht aus n atomaren Suchbedingungen x_i, die durch die Junktoren *UND* ($\wedge$) und *ODER* ($\vee$) miteinander verknüpft sind. Weiterhin besteht die Möglichkeit, Anfrageterme zu negieren, so dass eine Anfragebedingung im Allgemeinen wie folgt aufgebaut ist:

$$X := x \mid X\ [\wedge|\vee]\ X \mid \neg X.$$

Auf Grund der Assoziativität können die Junktoren $\wedge$ und $\vee$ zu n-ären Junktoren verallgemeinert werden.

Bei der Evaluierung einer Anfrage wird für jedes Objekt der Datenbank $o_j \in DB$ und jede atomare Suchbedingung x_i ein Ähnlichkeitswert $\mu_i(o_j)$ im Intervall $[0,1]$ berechnet. Wird bei dem Vergleich eines Anfrageobjektes mit einem Datenbankobjekt keine Ähnlichkeitsfunktion sondern eine Distanzfunktion verwendet, so müssen die berechneten Distanzen in einen Ähnlichkeitswert aus dem Intervall $[0,1]$ überführt werden. Der Ähnlichkeitswert drückt aus, inwieweit das Objekt o_j die Suchbedingung x_i erfüllt. Dieser Wert wird daher auch als Relevanzwert bezeichnet. Je höher der Relevanzwert $\mu_i(o_j)$ ist, desto höher ist der Grad der Ähnlichkeit zwischen der Anfrage und dem Datenbankobjekt bezüglich der Suchbedingung x_i.

Liegen Ähnlichkeitswerte vor, so werden die Relevanzwerte $\mu_i(o_j)$ für ein Objekt o_j zu einem Gesamtrelevanzwert $\mu(o_j)$ zusammengefaßt. Dafür werden die einzelnen Fuzzy-Mengen mittels der Junktoren *UND* und *ODER* zu einer neuen Fuzzy-Menge kombiniert. Für jedes Objekt wird dabei ein neuer Relevanzwert berechnet. Die Kombination der Relevanzwerte für ein Objekt wird mittels einer Scoring-Funktion

$$S : [0,1]^n \rightarrow [0,1]$$

für eine Anfrage X realisiert.

Typischerweise basiert S auf T-Normen und T-Conormen. Eine Konjunktion von Anfragetermen wird daher evaluiert unter Verwendung einer T-Norm und eine Disjunktion von Anfragetermen dementsprechend unter Verwendung einer T-Conorm (siehe Zadeh (1965); Bellman und Giertz (1973); Dubois und Prade (1984); Zimmermann (1990)). Der Einsatz von Fuzzy-basierten Scoring-Funktionen ist weit verbreitet. Solche Funktionen finden beispielsweise in den Ansätzen von Herrera-Viedma (2001b) und Carson u. a. (1997) Anwendung. Daneben werden auch andere Scoring-Funktionen, wie probabilistische Funktionen, eingesetzt (Fuhr und Grosjohann, 2001).

In einer empirischen Studie haben Koll und Srinivasan (1990) nachgewiesen, dass Scoring-Funktionen, die auf Fuzzy-Logik basieren, für die Kombination von Anfragetermen besser geeignet sind, als solche, denen das probabilistische Modell zu Grunde liegt. Es wird argumentiert, dass die Fuzzy-Logik näher an der menschlichen Art der Anfragebeschreibung und der Entscheidungsfindung als das probabilistische Modell liegt. Soll keine Unterscheidung zwischen *UND* und *ODER* erfolgen, kann beispielsweise das arithmetische Mittel der Relevanzwerte gebildet werden. Dieser Ansatz wird in Oracle (2003) verfolgt.

Um Nutzerpräferenzen in Multimedia-Anfragen ausdrücken zu können, hat sich in den vergangenen Jahren die Relevanzgewichtung einzelner Anfragebedingungen durchgesetzt (Fagin und Wimmers, 1997; Carson u. a., 1997; Sung, 1998; Ortega u. a., 1998; Oracle, 2003). Dabei wird jedem Anfrageterm x_i ein numerisches Gewicht $\theta_i \in [0,1]$ zugeordnet, welches die relative Wichtigkeit des Anfrageterms innerhalb der Anfragebedingung festlegt. Die Gewichtung der Suchbedingungen ist in dem Anfragebaum in Abbildung 4.1 für eine n-äre Konjunktion dargestellt. Als Bedingung wird gefordert, dass die Summe der Gewichte eins ergibt: $\sum_{i=1}^{n} \theta_i = 1$.

Im Allgemeinen erfolgt eine positive Gewichtung einzelner Teilanfragen ($\theta_i \in [0,1]$). Eine Ausweitung des Intervalls von $\theta_i \in T$ und $T = [-1,1]$ mit $\sum_{i=1}^{n} |\theta_i| = 1$ ist denkbar. Dadurch werden die Möglichkeiten des Nutzers, spezielle Wünsche differenzierter auszudrücken, er-

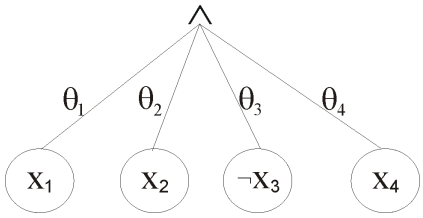


Abbildung 4.1: Anfragebaum einer einfachen, gewichteten, konjunktiven Anfrage

weitert. Durch eine negative Gewichtung ist es möglich, unerwünschte Teilanfragen entsprechend zu bewerten. Sinnvoll ist dieses, wenn der Nutzer neben speziellen Suchanforderungen auch einen 'Negativkatalog' an Forderungen hat. Eine solche, negativ bewertete Anfrage ist im folgenden Beispiel dargestellt.

Beispiel 4.1 *Suche Bilder von Flugzeugen, die viel rot aufweisen und keinen Doppeldecker enthalten. Die Bilder sollten möglichst auch nicht nach 1950 entstanden sein. Gewichtet ergibt sich beispielsweise die Anfragebedingung* $X = ((Farbe = rot)_{0.3} \wedge (Form = Doppeldecker)_{-0.4} \wedge (Jahr > 1950)_{-0.3})$.

Negativ gewichtete Anfragebedingungen lassen sich auf ein positives Problem zurückführen, indem der entsprechende Anfrageterm und die Gewichtung negiert werden. Die im Beispiel 4.1 geschilderte Anfragebedingung kann demnach wie folgt formuliert werden:

$$(Farbe = rot)_{0.3} \wedge (Form \neq Doppeldecker)_{0.4} \wedge (Jahr \leq 1950)_{0.3}.$$

Im Folgenden wird daher davon ausgegangen, dass eine negativ gewichtete Anfragebedingung in eine positiv gewichtete Bedingung überführt wird. Aus diesem Grund wird angenommen, dass $\theta_i \in [0,1]$ gilt.

Bei der Evaluierung von gewichteten Anfragen muß die Gewichtung berücksichtigt werden. Aus diesem Grund fließen die einzelnen Gewichte in die Scoring-Funktion S ein. Somit ergibt sich eine gewichtete Scoring-Funktion

$$S^{\Theta} : [0,1]^n \times [0,1]^n \rightarrow [0,1].$$

4.1.2 Anforderungen an gewichtete Scoring-Funktionen

Um die Anfragebearbeitung adäquat unterstützen zu können, werden verschiedene Anforderungen an Scoring-Funktionen gestellt. Diese erweitern die Anforderungen an eine Scoring-Funktion, die von Fagin (1999) formuliert wurden. Damit muss eine ungewichtete Scoring-Funktion $S : [0,1]^n \rightarrow [0,1]$, die Ähnlichkeitswerte auf einen Gesamtrelevanzwert im Intervall $[0,1]$ abbildet, den folgenden Forderungen gerecht werden:

Grenzbedingungen: $S(1,\ldots,1) = 1$ und $S(0,\ldots,0) = 0$

Monotonie: Wenn die Ähnlichkeitswerte nicht sinken, dann darf der berechnete Relevanz-

wert nur größer werden. Somit muss gelten:

$$\mu_{a_1}(o) \leq \mu_{b_1}(o) \wedge \ldots \wedge \mu_{a_n}(o) \leq \mu_{b_n}(o) \Rightarrow S(\mu_{a_1}(o), \ldots, \mu_{a_n}(o)) \leq S(\mu_{b_1}(o), \ldots, \mu_{b_n}(o)).$$

Strenge Monotonie: Wenn alle Ähnlichkeitswerte steigen, dann muss auch der berechnete Relevanzwert steigen. Somit muss gelten:

$$\mu_{a_1}(o) < \mu_{b_1}(o) \wedge \ldots \wedge \mu_{a_n}(o) < \mu_{b_n}(o) \Rightarrow S(\mu_{a_1}(o), \ldots, \mu_{a_n}(o)) < S(\mu_{b_1}(o), \ldots, \mu_{b_n}(o)).$$

Stetigkeit: Die Scoring-Funktion muss bezüglich der Werte $\mu_i(o)$ stetig sein.

Neben diesen geforderten Eigenschaften kann eine Scoring-Funktion weitere Eigenschaften, wie beispielsweise Idempotenz $S(\mu_1(o), \mu_1(o)) = \mu_1(o)$, aufweisen.

Eine gewichtete Scoring-Funktion $S^{\Theta} : [0,1]^n \times [0,1]^n \rightarrow [0,1]$ stellt eine um Gewichte erweiterte Scoring-Funktion dar. Deshalb gelten für eine gewichtete Scoringfunktion die selben Eigenschaften, wie für die zu Grunde liegende Scoring-Funktion. Da zusätzlich die Gewichte einen Einfluß auf das Verhalten dieser Funktion nehmen, müssen die oben aufgeführten Anforderungen unter Beachtung der Gewichte wie folgt modifiziert werden:

Grenzbedingungen: müssen unabhängig von den Gewichten erhalten bleiben.

Monotonie: muss bei gleich bleibenden Gewichten gelten.

Strenge Monotonie: muss bei gleich bleibenden Gewichten gelten.

Stetigkeit: Die gewichtete Scoring-Funktion muss sowohl bezüglich der Werte $\mu_i(o)$ als auch bezüglich der Gewichte θ_i stetig sein.

Zusätzlich dazu ist es sinnvoll, für gewichtete Scoring-Funktionen weitere Anforderungen bezüglich des Einflusses der Gewichte zu spezifizieren (Fagin und Wimmers, 1997; Sung, 1998). Demnach dürfen beispielsweise Anfrageterme, die mit *null* gewichtet werden, keinen Einfluß auf den zu berechnenden Relevanzwert haben. Zusammenfassend werden die folgenden Anforderungen an eine gewichtete Scoring-Funktion gestellt:

A1: Alle Eigenschaften der zu Grunde liegenden Scoring-Funktion S müssen von der gewichteten Scoring-Funktion S^{Θ} unter Beachtung der Gewichte aufrecht erhalten werden.

A2: Weisen alle Anfrageterme das gleiche Gewicht auf, also $\theta_1 = \ldots = \theta_n = 1/n$, so hat innerhalb der Anfrage kein Anfrageterm einen größeren Einfluß als irgendein anderer Term. Dies entspricht dem ungewichteten Fall. Damit ist die gewichtete Scoring-Funktion äquivalent zur ungewichteten Scoring-Funktion. Es gilt:

$$S^{\Theta}(\mu_1(o), \ldots, \mu_n(o), 1/n, \ldots, 1/n) = S(\mu_1(o), \ldots, \mu_n(o)).$$

A3: Wird ein Anfrageterm mit *null* gewichtet, dann ist dieser Term nicht von Interesse. Aus diesem Grund darf ein mit *null* gewichteter Anfrageterm keinen Einfluß auf die Berechnung des Relevanzwertes nehmen. Damit kann der Term von der Anfrage entfernt

werden. Es gilt:

$$S^{\Theta}(\mu_1(o),\ldots,\mu_n(o),\theta_1,\ldots,\theta_{n-1},0) = S^{\Theta}(\mu_1(o),\ldots,\mu_{n-1}(o),\theta_1,\ldots,\theta_{n-1}).$$

A4: Lokale Linearität fordert, dass die Scoring-Funktion als eine Art Balance zwischen den Gewichtungen agiert (Fagin und Wimmers, 1997). Das heißt konkret, dass bei einer Gewichtung im Mittelpunkt zweier co-monotonen Gewichtungen[1] ein RSV ermittelt wird, welcher dem Mittel aus den Relevanzwerten der beiden co-monotonen Gewichtungen entspricht.

Abbildung 4.2 verdeutlicht die Forderung nach lokaler Linearität grafisch. Für die co-monotonen Gewichtungen $\Theta = (0.6, 0.4)$ und $\Theta' = (0.8, 0.2)$ ergibt sich ein Relevanzwert μ beziehungsweise μ'. Für eine Gewichtung $\Theta^m = (0.7, 0.3)$, die im Mittelpunkt von Θ und Θ' liegt, wird ein Relevanzwert μ^m berechnet. Dabei gilt $\mu^m = \frac{\mu+\mu'}{2}$.

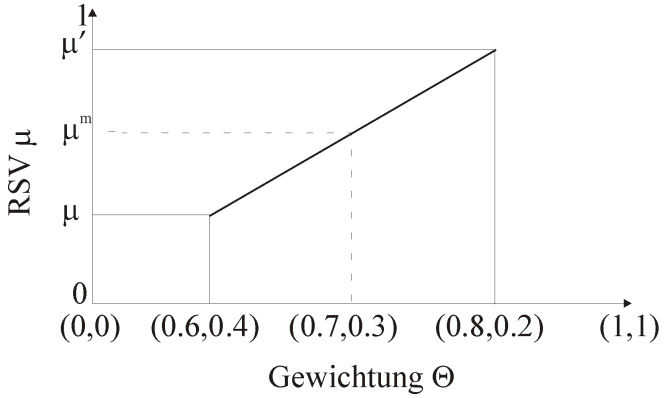


Abbildung 4.2: Anforderung **A4** (Lokale Linearität)

Neben diesen speziellen Anforderungen an eine einzelne gewichtete Scoring-Funktion ist die Forderung nach Stabilität eines allgemeinen Ansatzes zur Überführung einer Scoring-Funktion in eine gewichtete Scoring-Funktion von Bedeutung. Dabei ist die Äquivalenz von Scoring-Funktionen, also die Erzeugung gleicher Ordnungen für die Ergebnisobjekte, ausschlaggebend. Diese Meta-Anforderung **MA** wird wie folgt angegeben:

MA: Ein Ansatz zur Generierung gewichteter Scoring-Funktionen muss stabil sein. Das bedeutet, dass bei Verwendung der gewichteten Scoring-Funktionen S^{Θ} und S'^{Θ}, welche auf äquivalenten Scoring-Funktionen S und S' basieren, jeweils die gleiche Reihenfolge

[1] Zwei Gewichtungen $\Theta = \{\theta_1,\ldots,\theta_n\}$ und $\Theta' = \{\theta'_1,\ldots,\theta'_n\}$ sind co-monoton, wenn die Reihenfolge der Gewichte gleich ist, das heißt, es gibt kein i, j mit $\theta_i < \theta_j$ und $\theta'_j < \theta'_i$.

der Ergebnisobjekte bestimmt wird (Sung, 1998). Somit gilt:

$$S(\mu_1(o_1),\ldots,\mu_n(o_1)) < S(\mu_1(o_2),\ldots,\mu_n(o_2)) < \ldots < S(\mu_1(o_m),\ldots,\mu_n(o_m)) \wedge$$
$$S'(\mu_1(o_1),\ldots,\mu_n(o_1)) < S'(\mu_1(o_2),\ldots,\mu_n(o_2)) < \ldots < S'(\mu_1(o_m),\ldots,\mu_n(o_m)) \Rightarrow$$
$$S^{\Theta}(\mu_1(o_1),\ldots,\mu_n(o_1),\theta_1,\ldots,\theta_n) < S^{\Theta}(\mu_1(o_2),\ldots,\mu_n(o_2),\theta_1,\ldots,\theta_n) < \ldots <$$
$$S^{\Theta}(\mu_1(o_m),\ldots,\mu_n(o_m),\theta_1,\ldots,\theta_n) \wedge S'^{\Theta}(\mu_1(o_1),\ldots,\mu_n(o_1),\theta_1,\ldots,\theta_n) <$$
$$S'^{\Theta}(\mu_1(o_2),\ldots,\mu_n(o_2),\theta_1,\ldots,\theta_n) < \ldots < S'^{\Theta}(\mu_1(o_m),\ldots,\mu_n(o_m),\theta_1,\ldots,\theta_n).$$

4.2 Verfahren zur Relevanzgewichtung von Anfragen

In den folgenden Abschnitten werden die Ansätze von Fagin und Wimmers (1997) und der daraufhin erarbeitete Ansatz von Sung (1998) vorgestellt und auf Basis der formulierten Anforderungen evaluiert. Diese beiden Ansätze beschreiben Verfahren für die Integration von Relevanzgewichten in beliebige Funktionen. Aus diesem Grund werden die Anforderungen **A1** bis **A4** sowie die Meta-Anforderung **MA** überprüft.

Neben diesen allgemeingültigen Ansätzen werden spezielle gewichtete Scoring-Funktionen vorgestellt. Dazu zählen die Ansätze von Dubois und Prade (1986) und Bookstein (1980) sowie Scoring-Funktionen, wie sie in den Systemen Blobworld (Carson u. a., 1997), und Oracle (Hebert u. a., 1999; Oracle, 2003) eingesetzt werden. Oracle gilt hier als Sonderfall, da der gewichtete Durchschnitt der zu aggregierenden Werte gebildet wird. Ansonsten werden auf der Fuzzy-Logik basierende Funktionen verwendet. So wird für die Disjunktion das *Maximum* und für die Konjunktion das *Minimum* der zu aggregierenden Werte verwendet. Die Scoring-Funktionen *min* bildet daher auch für die Evaluierung der beiden allgemeinen Ansätze die Ausgangsbasis. Diese gewichteten Scoring-Funktionen werden hinsichtlich der Anforderungen **A1** bis **A4** untersucht. Teile dieser Evaluierung basieren auf der Veröffentlichung von Schulz und Schmitt (2002).

Für die Überprüfung der Anforderung **A1** ist es notwendig, die Eigenschaften der zu Grunde liegenden Scoring-Funktion zu kennen. Daher werden die Charakteristika der T-Norm *min* und der T-Conorm *max* zunächst kurz vorgestellt.

Definition 4.2.1. Eine Funktion $T : [0,1] \times [0,1] \rightarrow [0,1]$ ist eine T-Norm, wenn die folgenden Eigenschaften erfüllt sind (Biewer, 1997, S. 83):

(i)	$T(a,b) = T(b,a)$	(Kommutativität)
(ii)	$T(a,T(b,c)) = T(T(a,b),c)$	(Assoziativität)
(iii)	$a \leq c \wedge b \leq d \Rightarrow T(a,b) \leq T(c,d)$	(Monotonie)
(iv)	$T(a,1) = a$	(Neutrales Element).

□

Dementsprechend kann die Definition für eine T-Conorm wie folgt angegeben werden.

Definition 4.2.2. Eine Funktion $T : [0,1] \times [0,1] \rightarrow [0,1]$ ist eine T-Conorm, wenn die folgenden Eigenschaften erfüllt sind (Biewer, 1997, S. 84):

(i)-(iii)	wie in Definition 4.2.1	
(iv)	$T(a,0) = a$	(Neutrales Element).

□

Zusätzlich zu den aufgeführten Eigenschaften von T-Normen und T-Conormen gelten konkret für die T-Norm *min* und somit für die entsprechende T-Conorm *max* folgende Eigenschaften:

- Idempotenz: $T(a,a) = a$,
- Strenge Monotonie: $a < c, b < d \Rightarrow T(a,b) < T(c,d)$,
- Stetige Funktion,
- Translationserhaltung: $T(a \circ c, b \circ c) = T(a,b) \circ c$ mit $\circ \in \{+,-,*,/\}$.

Dabei ist anzumerken, dass *min* die einzige idempotente T-Norm und *max* die einzige idempotente T-Conorm sind (Klir und Yuan, 1995, S. 63ff). Somit sind *min* und *max* auch die einzigen Normen, die Distributivität gewährleisten.

4.2.1 Allgemeingültige Ansätze

Für die Integration von Relevanzgewichten in eine Scoring-Funktion existieren zwei alternative Ansätze: die *Gewichtungsformel* und das *Transformationsschema*. Beide Verfahren werden im Folgenden vorgestellt.

Gewichtungsformel

Fagin und Wimmers (1997) entwickeln eine einfache Formel, um Gewichte in jede beliebige ungewichtete Funktion zu integrieren. Die Gewichtungsformel geht von einer ungewichteten Scoringfunktion $S(\mu_1(o), \ldots, \mu_n(o))$ und einer Gewichtung $\Theta = \{\theta_1, \ldots, \theta_n\}$ aus (Fagin und Wimmers, 1997, 2000; Fagin und Maarek, 2000). Jedes Gewicht θ_i ist einem atomaren Anfrageterm x_i zugeordnet. Weiterhin gilt $\theta_i \in [0,1]$ und $\sum_{i=1}^{n} \theta_i = 1$ und $\theta_1 \geq \theta_2 \geq \ldots \geq \theta_n$. Unter Beachtung dieser Annahmen kann mittels der Formel 4.1 eine beliebige Scoring-Funktion S in eine gewichtete Scoring-Funktion S^{Θ} überführt werden.

$$\begin{aligned} S^{\Theta}(\mu_1(o), \ldots, \mu_n(o), \theta_1, \ldots, \theta_n) \quad = \quad & (\theta_1 - \theta_2) S(\mu_1(o)) + \\ & 2 * (\theta_2 - \theta_3) S(\mu_1(o), \mu_2(o)) + \\ & \ldots + \ldots \\ & n * \theta_n S(\mu_1(o), \ldots, \mu_n(o)) \end{aligned} \tag{4.1}$$

Auf Basis der Formel 4.1 kann das *gewichtete Minimum* S^{Θ}_{min} wie folgt angegeben werden:

$$\begin{aligned} S^{\Theta}_{min}(\mu_1(o), \ldots, \mu_n(o), \theta_1, \ldots, \theta_n) \quad = \quad & (\theta_1 - \theta_2) * min(\mu_1(o)) + \\ & 2 * (\theta_2 - \theta_3) * min(\mu_1(o), \mu_2(o)) + \\ & \ldots + \ldots \\ & n * \theta_n * min(\mu_1(o), \ldots, \mu_n(o)). \end{aligned}$$

Die Scoring-Funktion S^{Θ}_{min} wird beispielsweise in der Multimedia-Anfragesprache FOQL (Nepal u. a., 1999) sowie in der Ähnlichkeitsalgebra SAME^W (Montesi und Penzo, 2000; Ciaccia u. a., 2000) für die Evaluierung gewichteter Anfragen verwendet.

Die mit der Gewichtungsformel generierte gewichtete Scoring-Funktion S^{Θ}_{min} basiert auf der ungewichteten Scoring-Funktion. Es handelte sich ferner um eine stetige, streng monotone Funktion, welche die Grenzbedingungen erfüllt (Fagin und Wimmers, 2000). Darüber hinaus werden die speziellen Eigenschaften der ungewichteten Scoring-Funktion, wie beispielsweise die Idempotenz und die Translationserhaltung aufrecht erhalten (ebd.). Damit ist die Anforderung **A1** erfüllt. Des Weiteren ist die gewichtete Funktion lokal linear und erfüllt die Anforderungen bezüglich des Einflusses der Gewichte, so dass auch die Anforderungen **A2**, **A3** und **A4** erfüllt werden. Damit ist die mit der Gewichtungsformel erzeugte gewichtete Funktion auf Basis der T-Norm *min* eine gültige gewichtete Scoring-Funktion.

Der Ansatz von Fagin und Wimmers gewährleistet, dass alle gewichteten Funktionen, die mit der Gewichtungsformel erzeugt werden, den Anforderungen einer gewichteten Scoring-Funktion genügen. Auf Grund der Tatsache, dass es sich bei der Gewichtungsformel um einen allgemeingültigen Ansatz handelt, muss überprüft werden, ob auch die Meta-Anforderung gilt.

Die Gewichtungsformel ist daraufhin ausgelegt, dass jede beliebige Funktion in eine gewichtete Funktion überführt werden kann. Die Überführung von Funktionen wird also nicht auf Scoring-Funktionen, die den in Abschnitt 4.1.2 formulierten Anforderungen unterliegen, eingeschränkt. Aus diesem Grunde tritt im allgemeinen Fall das Problem der Instabilität der Gewichtungsformel auf. Dies wurde bereits sowohl von Fagin und Wimmers (2000) als auch von Sung (1998) erkannt. Der Forderung nach Stabilität kann nur dann genüge getan werden, wenn die entsprechenden äquivalenten Funktionen die Anforderungen an Scoring-Funktionen erfüllen. Im Folgenden wird anhand eines Beispiels das Problem der Stabilität verdeutlicht.

Stabilität im allgemeinen Fall Es wird angenommen, dass $\Theta = \{\theta_1, \theta_2\}$ mit $\theta_1 \geq \theta_2$ gilt. Als Funktionen für die Aggregation der Ähnlichkeitswerte werden die *Summe* und der *Durchschnitt* eingesetzt. Diese beiden Funktionen sind äquivalent. Sie liefern das gleiche Rankingergebnis für die Objekte, obwohl sie jeweils verschiedene Relevanzwerte für ein Objekt ermitteln. Unter Verwendung der Formel 4.1 ergibt sich für die *gewichtete Summe*:

$$S^{\Theta}_{sum}(\mu_1(o), \mu_2(o), \theta_1, \theta_2) = \mu_1(o) + 2\theta_2\mu_2(o)$$

und für den *gewichteten Durchschnitt*:

$$S^{\Theta}_{avg}(\mu_1(o), \mu_2(o), \theta_1, \theta_2) = \theta_1\mu_1(o) + \theta_2\mu_2(o).$$

Für das Objekt o_1 mit $\mu_1(o_1) = 0.4$ und $\mu_2(o_1) = 0.7$ und Objekt o_2 mit $\mu_1(o_2) = 0.7$ und $\mu_2(o_2) = 0.1$ und der Gewichtung $\Theta = \{0.7, 0.3\}$ ergibt sich für die *gewichtete Summe*:

$$\begin{aligned} \mu(o_1) &= S^{\Theta}_{sum}(0.4, 0.7, 0.7, 0.3) \\ &= 0.82 \end{aligned}$$

$$\begin{aligned} \mu(o_2) &= S^{\Theta}_{sum}(0.7, 0.1, 0.7, 0.3) \\ &= 0.76. \end{aligned} \tag{4.2}$$

Damit liegt Objekt o_1 vor Objekt o_2 in der Ergebnisliste. Für den *gewichteten Durchschnitt* ergibt sich:

$$\begin{aligned} \mu(o_1) &= S^{\Theta}_{avg}(0.4, 0.7, 0.7, 0.3) \\ &= 0.49 \end{aligned}$$

$$\begin{aligned} \mu(o_2) &= S^{\Theta}_{avg}(0.7, 0.1, 0.7, 0.3) \\ &= 0.52. \end{aligned}$$

Die Ergebnisliste unter Einsatz des *gewichteten Durchschnitts* ist $\langle (o_2, 0.52); (o_1, 0.49) \rangle$. Bei Verwendung der *gewichteten Summe* ergibt sich $\langle (o_1, 0.82); (o_2, 0.76) \rangle$. Die Reihenfolge der Ergebnisobjekte ist also vertauscht. Dieses Verhalten ist nicht wünschenswert und führt zu einer Instabilität. Die Abbildung 4.3 zeigt diesen Sachverhalt für die Objekte o_1 und o_2. Liegt die Gewichtung in einem bestimmten Bereich, dann führt dies zu einem instabilen Verhalten des Verfahrens. Dieser kritische Bereich ist in der Abbildung gestrichelt dargestellt.

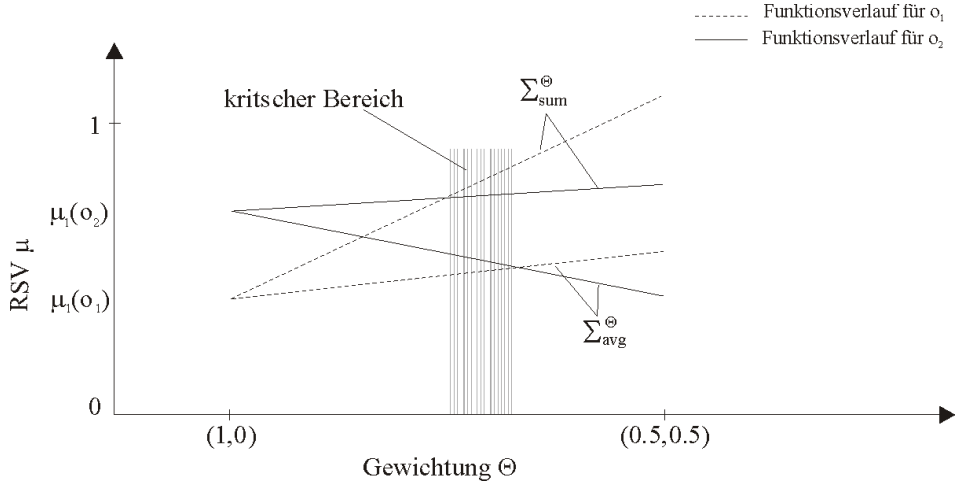


Abbildung 4.3: Problem der Stabilität der Gewichtungsformel

Das von Fagin und Wimmers entwickelte Gewichtungsverfahren ist daher für beliebige Funktionen nicht stabil. Die unterschiedliche Reihenfolge der Ergebnisobjekte rührt daher, dass bei den zu Grunde liegenden Funktionen keine Normierung auf das Intervall $[0, 1]$ stattfindet und die Grenzbedingungen verletzt werden. So ist die Summe keine Scoring-Funktion entsprechend der hier formulierten Anforderungen. Ist nur eine Gewichtung von Scoring-Funktionen zulässig, die die geforderten Eigenschaften erfüllen, so wird der Forderung nach Stabilität genüge getan.

Damit können durch Anwendung der Gewichtungsformel auf eine Scoring-Funktion alle Anforderungen (**A1** bis **A4**) an eine gewichtete Scoring-Funktion sowie die Meta-Anforde-

rung **MA** erfüllt werden.

Transformationsschema

Im Vergleich zur Gewichtungsformel von Fagin und Wimmers (1997) ist der Ansatz von Sung (1998) eher konzeptionell. Die Idee des Transformationsschemas basiert darauf, dass der Einfluss eines einzelnen Ähnlichkeitswertes sich proportional zu seinem Gewicht verhält. Daher werden die einzelnen Ähnlichkeitswerte direkt mit den Gewichten kombiniert. Es findet eine Transformation der Scoring-Funktion $S(\mu_1(o), \ldots, \mu_n(o))$ unter Einbeziehung der Gewichtung $\Theta = (\theta_1, \ldots, \theta_n)$ mit $\sum_{i=1}^{n} \theta_i = 1$ und $\theta_i \in [0, 1]$ statt. Für eine lineare Transformation der Form $\mu_i(o) \rightarrow \theta_i \mu_i(o)$ ergibt sich:

$$S^{\Theta}(\mu_1(o), \ldots, \mu_n(o), \theta_1, \ldots, \theta_n) = S(\theta_1 \mu_1(o), \ldots, \theta_n \mu_n(o)).$$

Neben der linearen Transformation von Scoring-Funktionen sind andere Arten der Transformation in einigen Fällen aussagekräftiger (Sung, 1998). Der Einsatz einer exponentiellen Transformation $\mu_i(o) \rightarrow \mu_i^{\theta_i}(o)$ ist beispielsweise bei Verwendung einer auf dem Produkt basierenden Funktion $S(\mu_1(o), \ldots, \mu_n(o)) = \mu_1(o) * \ldots * \mu_n(o)$ sinnvoll (ebd.). Der mittels des Transformationsschemas berechnete Relevanzwert $\mu(o)$ für ein Objekt o liegt im Bereich $[0, \infty)$ und kann durch eine entsprechende Normalisierung in den Bereich $[0, 1]$ überführt werden.

Das *gewichtete Minimum* kann unter Verwendung des Transformationsschemas nicht direkt hergeleitet werden, da das *Minimum* mit dem Konzept der Relevanzgewichtung im Konflikt steht. Bei der Relevanzgewichtung wird der Einfluss eines Anfrageterms durch das Gewicht ausgedrückt. Je höher das Gewicht ist, desto wichtiger ist der Anfrageterm. Damit soll auch der Einfluss des Terms innerhalb der Anfrage steigen. Der Konflikt rührt nun daher, dass das *Minimum* genau die Terme 'belohnt', die einen möglichst kleinen Ähnlichkeitswert aufweisen. Aus diesem Grund schlägt Sung (1998) vor, die folgende zum *Minimum* äquivalente jedoch auf dem *Maximum* basierende Funktion zu verwenden:

$$min(\mu_1(o), \ldots, \mu_n(o)) = 1 - max(1 - \mu_1(o), \ldots, 1 - \mu_n(o)). \tag{4.3}$$

Bei einer linearen Transformation kann durch Verwendung der Formel 4.3 das *gewichtete Minimum* wie folgt angegeben werden:

$$S_{min}^{\Theta}(\mu_1(o), \ldots, \mu_n(o), \theta_1, \ldots, \theta_n) = 1 - \frac{1}{M} max\{\theta_1 - \theta_1 \mu_1(o), \ldots, \theta_n - \theta_n \mu_n(o)\}.$$

Dabei ist $M = max\{\theta_1, \ldots, \theta_n\}$, wodurch die gewichtete Scoring-Funktion standardisiert wird, so dass $\mu(o) \in [0, 1]$ gilt.

Das *gewichtete Minimum* ist eine stetige, streng monotone Funktion, die Werte im Bereich $[0, 1]$ zurückliefert und die Grenzbedingungen erfüllt (Fagin und Wimmers, 2000; Sung, 1998). Darüber hinaus ist sie idempotent. Jedoch werden nicht alle Eigenschaften der zu Grunde liegenden T-Norm *min* aufrecht erhalten. So ist das *gewichtete Minimum* nicht mehr translationserhaltend. Damit wird die Anforderung **A1** nicht erfüllt. Des Weiteren kann die lokale Linearität (**A4**) nicht gewährleistet werden (Fagin und Wimmers, 2000). Jedoch werden die Anforderungen bezüglich des Einflusses der Gewichte erfüllt. Damit erfüllt die mit dem

Transformationsschema erzeugte gewichtete Funktion nur die Anforderungen **A2** und **A3**.

Ähnlich wie die Gewichtungsformel erhebt das Transformationsschema den Anspruch, jede beliebige Funktion in eine gewichtete Funktion zu überführen. Im Gegensatz zur Gewichtungsformel stellt das Transformationsschema jedoch für beliebige äquivalente Funktionen ein stabiles Gewichtungsverfahren dar (Sung, 1998).

4.2.2 Spezielle, gewichtete Scoring-Funktionen

Die Verknüpfung von Anfragetermen erfolgt mittels der Junktoren *UND* und *ODER*. Bei der Evaluierung der Anfrage wird ein *UND* klassischerweise als das *Minimum* und ein *ODER* als das *Maximum* der jeweiligen Ähnlichkeitswerte eines Objektes interpretiert. Aus diesem Grund gibt es eine Vielzahl von gewichteten Scoring-Funktionen, die auf diesen beiden Funktionen basieren. Im Folgenden werden verschiedene Alternativen vorgestellt und hinsichtlich der Erfüllung der Anforderungen an eine gewichtete Scoring-Funktion untersucht.

Manchmal ist es wünschenswert, keine Unterscheidung zwischen *UND* und *ODER* vorzunehmen. In einem solchen Fall wird oftmals der *gewichtete Durchschnitt* als Scoring-Funktion eingesetzt. Damit wird ein neutrale Kombination gewährleistet, so dass das Ergebnis genau zwischen dem *Minimum* und dem *Maximum* liegt. Diese Funktion wird beispielsweise in dem MADT von Oracle, dem *interMedia-Paket*, eingesetzt. Auch dieser Ansatz wird in diesem Abschnitt kurz vorgestellt.

Gewichtetes Minimum von Dubois und Prade

Sowohl Dubois und Prade (1986) als auch Yager (1987) erweitern die T-Norm *min* und die T-Conorm *max* um Gewichte. Dem zu Grunde liegen die Forderungen, dass sich bei einer Gleichgewichtung der Anfrageterme das *gewichtete Minimum* auf die Funktion *min* und das *gewichtete Maximum* auf die Funktion *max* reduziert. Dieses entspricht der Anforderung **A2**. Ferner soll ein mit *null* gewichteter Term keinen Einfluss auf die Ergebnisberechnung haben. Dies korrespondiert zur Anforderung **A3**. Auf Basis dieser Annahmen kann das *gewichtete Minimum* wie folgt angegeben werden (Dubois und Prade, 1986):

$$S^{\Theta}_{min}(\mu_1(o),\ldots,\mu_n(o),\theta_1,\ldots,\theta_n) = min_{i=1\ldots n}(max(1-\frac{\theta_i}{M},\mu_i(o))).$$

Das *gewichtete Maximum* ergibt sich als:

$$S^{\Theta}_{max}(\mu_1(o),\ldots,\mu_n(o),\theta_1,\ldots,\theta_n) = max_{i=1\ldots n}(min(\frac{\theta_i}{M},\mu_i(o))).$$

Dabei ist $M = max\{\theta_1,\ldots,\theta_n\}$, wodurch die gewichtete Scoring-Funktion standardisiert wird, so dass $\mu(o) \in [0,1]$ gilt. In den beiden Veröffentlichungen von Dubois und Prade (1996, 2001) und Yager (1987) werden die Funktionen ohne den Normierungsfaktor M eingesetzt.

Das *gewichtete Minimum* ist eine stetige, monotone und idempotente Funktion, deren ermittelte Relevanzwerte im Bereich [0,1] liegen (Sung, 1998; Fagin und Wimmers, 2000). Darüber hinaus werden die Grenzbedingungen erfüllt. Jedoch ist diese gewichtete Scoring-Funktion weder streng monoton noch translationserhaltend, so dass der Anforderung **A1**

nicht Rechnung getragen wird. Denn stehen für ein Objekt o die Ähnlichkeitswerte in folgender Relation $\mu_1(o) \geq 1 - \frac{\theta_2}{\theta_1} \geq \mu_2(o)$ zueinander, so ergibt sich unter Verwendung des *gewichteten Minimums* (Fagin und Wimmers, 2000):

$$S^{\Theta}_{min}(\mu_1(o), \mu_2(o), \theta_1, \theta_2) = 1 - \frac{\theta_2}{\theta_1}.$$

Angenommen, die Gewichte seien $\theta_1 = 2/3, \theta_2 = 1/3$ und die Ähnlichkeitswerte für das Objekt o seien $\mu_1(o) \geq 0.5 \geq \mu_2(o)$, dann gilt somit stets:

$$S^{\Theta}_{min}(\mu_1(o), \mu_2(o), \theta_1, \theta_2) = 0.5.$$

Dieses Ergebnis zeigt, dass das *gewichtete Minimum* unempfindlich gegenüber verschiedenen Argumenten in dem angegeben Bereich ist. Auf Grund dieses Verhaltens der Funktion kann die geforderte strenge Monotonie und die Translationserhaltung der zu Grunde liegenden Scoring-Funktion *min* nicht aufrecht erhalten werden. Des Weiteren kann der Forderung **A4** nach lokaler Linearität nicht nachgekommen werden (Fagin und Wimmers, 2000). Die Anforderungen **A2** und **A3** werden erfüllt. Sie bilden, wie oben aufgeführt, die Grundannahmen für die Entwicklung dieser Funktionen.

Ansatz von Bookstein

Bookstein (1980) entwickelt für die Aggregation von Relevanzwerten im Text-Retrieval eine auf Fuzzy-Logik basierende Methode, die es erlaubt, eine Gewichtung von Anfragetermen in einer Booleschen Anfrage vorzunehmen. Ein Gewicht θ_i mit $\theta_i \in [0, 1]$ wird einem Term x_i zugeordnet. Das Initialgewicht wird für jeden Term auf 1 gesetzt. Die Gewichte sind nicht normalisiert. Damit wird in diesem Ansatz die Gewichtung eines Anfrageterms unabhängig von den anderen Anfragetermen vorgenommen. Ein Gewicht $\theta_i < 1$ drückt daher nur aus, dass ein Anfrageterm weniger bedeutend für die Anfrage ist.

Für jeden Anfrageterm wird der Ähnlichkeitswert unter Berücksichtigung der Gewichte modifiziert. So ergibt sich für eine Disjunktion:

$$\begin{aligned} \mu_i(o)' &= \theta_i \mu_i(o) \\ S^{\Theta}_{max}(\mu_1, \ldots, \mu_n, \theta_1, \ldots, \theta_n) &= max(\mu'_1, \ldots, \mu'_n). \end{aligned} \tag{4.4}$$

Dementsprechend wird auch der Ähnlichkeitswert für einen Anfrageterm, der Teil einer Konjunktion ist, verändert:

$$\begin{aligned} \mu'_i(o) &= \begin{cases} \frac{1}{\theta_i}\mu_i(o) & \text{, falls } \frac{1}{\theta_i}\mu_i(o) < 1 \\ 1 & \text{, falls } \frac{1}{\theta_i}\mu_i(o) \geq 1 \text{ oder nicht definiert} \end{cases} \\ S^{\Theta}_{min}(\mu_1(o), \ldots, \mu_n(o), \theta_1, \ldots, \theta_n) &= min(\mu'_1(o), \ldots, \mu'_n(o)). \end{aligned}$$

Anhand des folgenden Beispiels soll die Idee dieses Ansatzes demonstriert werden.

Beispiel 4.2 *Die Anfrage*

Suche alle Bilder, die dem gegebenen Bild in Farbe oder Form ähnlich sind. Die Form wird

nur mit 0.5 gewichtet.

liefert für die Teilanfrage Farbe *die Ergebnisliste* $\langle(o_2, 1.0); (o_1, 0.7); (o_3, 0.4)\rangle$ *und für die Teilanfrage* Form *die Liste* $\langle(o_3, 0.9); (o_1, 0.8); (o_2, 0.4)\rangle$*. Somit ergibt sich unter Berücksichtigung der Gewichtung und unter Verwendung von Formel 4.4 die Gesamtergebnisliste:* $\langle(o_2, 1.0), (o_1, 0.7), (o_3, 0.45)\rangle$.

Es ist leicht ersichtlich, dass das *gewichtete Minimum* eine stetige, monotone Funktion ist, die Werte im Bereich [0,1] berechnet. Jedoch ist die gewichtete Funktion weder idempotent, streng monoton noch translationserhaltend. Damit wird gegen die Anforderung **A1** verstoßen. Des Weiteren kann dieser Ansatz der Forderung nach lokaler Linearität **A4** und der Anforderung **A2** (Gleichgewichtung der Terme) nicht gerecht werden. Bei einer Gewichtung eines Terms mit *null* wird dessen Relevanzwert bei einer Konjunktion auf *eins* gesetzt. Analog dazu wird dieser bei einer Disjunktion auf *null* gesetzt. Damit kann die Anforderung **A3** erfüllt werden.

Relevanzgewichtung im Retrieval-System Blobworld

Das Bild-Retrieval-System Blobworld unterstützt ebenfalls eine Relevanzgewichtung von Anfragetermen in Ähnlichkeitsanfragen (Carson u. a., 1997). In diesem Ansatz werden ähnlich wie im Transformationsschema die Gewichte direkt mit den Ähnlichkeitswerten verknüpft. Für die Disjunktion wird analog zu den bisherigen Ansätzen das *Maximum* und für die Evaluierung der Konjunktion von Anfragetermen das *Minimum* genutzt. Für jeden Anfrageterm x_i wird ein Gewicht θ_i angegeben, wobei $\sum_{i=1}^{n} \theta_i = 1$ und $\theta_i \in [0,1]$ gilt. Das *gewichtete Maximum* ergibt sich als:

$$S^{\Theta}_{max}(\mu_1(o), \ldots, \mu_n(o), \theta_1, \ldots, \theta_n) \quad = \quad max(\theta_1\mu_1(o), \ldots, \theta_n\mu_n(o))$$

und das *gewichtete Minimum* als:

$$\begin{aligned} S^{\Theta}_{min}(\mu_1(o), \ldots, \mu_n(o), \theta_1, \ldots, \theta_n) &= min(1 - \theta_1(1 - \mu_1(o)), \ldots, 1 - \theta_n(1 - \mu_n(o))) \\ &= 1 - max(\theta_1 - \theta_1\mu_1(o), \ldots, \theta_n - \theta_n\mu_n(o)). \end{aligned}$$

Bis auf eine Standardisierung entsprechen das *gewichtete Minimum* und das *gewichtete Maximum* den Funktionen, die mit dem linearen Transformationsschema von Sung (1998) ermittelt werden.

Somit gilt auch hier, dass das *gewichtete Minimum* streng monoton und stetig ist. Dennoch werden nicht alle Eigenschaften der zu Grunde liegenden Scoring-Funktion aufrechterhalten. So erfüllt das *gewichtete Minimum* auf Grund der fehlenden Standardisierung nicht die Grenzbedingungen. Darüber hinaus ist sie weder idempotent noch translationserhaltend, wodurch die Funktion der Anforderung **A1** nicht genügt. Analog zum Transformationsschema kann auch dieser Ansatz die Forderung nach lokaler Linearität nicht erfüllen (**A4**). Darüber hinaus wird auf Grund der fehlenden Standardisierung die Anforderung **A2** (Gleichgewichtung der Terme) verletzt. Lediglich die Forderung **A3** (Nullgewichtung eines Terms) wird erfüllt.

Relevanzgewichtung im MADT von Oracle

Das Datenbank-Management-System Oracle bietet mit dem *interMedia-Paket* eine Erweiterung um multimediaspezifische Datentypen und Operationen (Oracle, 2003). In diesem Zusammenhang wird auch eine Relevanzgewichtung mit $\theta_i \in [0,1]$ einzelner Anfrageterme bei der Formulierung von Anfragen unterstützt. Dabei gilt $\sum_{i=1}^{n} \theta_i = 1$. Bei der Evaluierung werden die einzelnen Bedingungen neutral mittels des *gewichteten Durchschnitts* verknüpft:

$$S^{\Theta}_{avg}(\mu_1(o), \ldots, \mu_n(o), \theta_1, \ldots, \theta_n) = \sum \mu_i(o)\theta_i. \tag{4.5}$$

Diese Funktion wird auch im Retrieval-System PREFER zur Aggregation der Relevanzwerte verwendet (Hristidis u. a., 2001). Wird die Gewichtungsformel von Fagin und Wimmers (1997) auf die Durchschnittsfunktion angewendet, so ergibt sich ebenfalls die Formel 4.5.

Das MADT von Oracle unterstützt nur die Verarbeitung der Features Farbe, Textur und Form. Auch eine lokale Betrachtung dieser Features kann vorgenommen werden. Diese Kriterien können miteinander gewichtet kombiniert werden. Eine Gewichtung von anderen Teilanfragen ist jedoch nicht möglich. Somit kann die Scoring-Funktion wie folgt angegeben werden:

```
weighted_avg = color_weight     * color_distance +
               texture_weight   * texture_distance +
               form_weight      * form_distance +
               location_weight  * location_distance.
```

Anzumerken ist, dass keine Ähnlichkeitswerte sondern Distanzwerte $d \in [0,100]$ aggregiert werden, so dass die Scoring-Funktion einen Relevanzwert $\mu \in [0,100]$ ermittelt. Bei einem berechneten Relevanzwert von *null* ist die Ähnlichkeit zwischen dem entsprechenden Datenbankobjekt und der Anfrage somit am größten. Liegen Ähnlichkeitswerte zu Grunde, erfüllt diese Funktion die geforderten Grenzbedingungen.

Die in Oracle eingesetzte gewichtete Scoring-Funktion ist stetig, streng monoton und besitzt alle Eigenschaften der zu Grunde liegenden ungewichteten Funktion. Somit erfüllt sie die Anforderung **A1**. Ferner ist die Funktion lokal linear und realisiert damit die Anforderung **A4**. Weiterhin wird die Forderung **A3** (Nullgewichtung eines Terms) erfüllt. Bei einer Gleichgewichtung der Anfrageterme, also $\theta_i = 1/n$, entspricht das Ergebnis dem Durchschnitt der zu aggregierenden Werte:

$$S^{\Theta}_{avg}(\mu_1(o), \ldots, \mu_n(o), \frac{1}{n}, \ldots, \frac{1}{n}) \quad = \quad \sum_{i=1}^{n} \mu_i(o)\frac{1}{n} = \frac{1}{n}\sum_{i=1}^{n} \mu_i(o).$$

Dadurch wird auch das Kriterium **A2** erfüllt.

4.2.3 Vergleich der Verfahren

Derzeitig gibt es eine Reihe von Verfahren zur Umsetzung einer Relevanzgewichtung von Anfragetermen. Die Tabelle 4.1 fasst die Ergebnisse der Evaluierung bezüglich der Erfüllung der einzelnen Anforderungen zusammen. Dabei zeigt das Symbol $\surd$ an, dass die Anforderung erfüllt ist und dementsprechend bedeutet $-$, dass die Anforderung nicht erfüllt ist. In der Tabelle werden die Anforderungen wie folgt dargestellt:

- **A1**: Die gewichtete Scoring-Funktion erfüllt die **G**renzbedingungen. Die Eigenschaften: **M**onotonie, **S**trenge **M**onotonie, **ST**etigkeit sowie die anderen Eigenschaften der Scoring-Funktion **S** werden von der gewichteten Scoring-Funktion aufrecht erhalten.
- **A2**: Bei gleichen Gewichten entspricht die gewichtete Scoring-Funktion der ungewichteten Scoring-Funktion.
- **A3**: Mit *null* gewichtete Anfrageterme werden bei der Aggregation nicht berücksichtigt.
- **A4**: Die gewichtete Scoring-Funktion ist lokal linear.

Die Meta-Anforderung (**MA**) ist nur für die allgemeinen Ansätze von Bedeutung und wurde daher auch nur für diese untersucht. Für einzelne Scoring-Funktionen kann über die Erfüllung von **MA** keine Aussage getroffen werden. Grundlage für die Evaluierung der allgemeinen Verfahren bildet das mit dem jeweiligen Ansatz erzeugte *gewichtete Minimum* als Scoring-Funktion.

Verfahren	**A1**					**A2**	**A3**	**A4**	**MA**
	G	**M**	**SM**	**ST**	**S**				
Gewichtungsformel	√	√	√	√	√	√	√	√	√
Transformationsschema	√	√	√	√	−	√	√	−	√
Dubois und Prade	√	√	−	√	−	√	√	−	
Bookstein	√	√	−	√	−	−	√	−	
Blobworld	−	√	√	√	−	−	√	−	
Oracle	(√)	√	√	√	√	√	√	√	

Tabelle 4.1: Zusammenfassung der Gewichtungsverfahren

Die Gewichtungsformel und das Transformationsschema sind allgemeine Ansätze, die es erlauben, eine beliebige Funktion in eine gewichtete Funktion zu überführen. Dabei ist die Gewichtungsformel nicht für beliebige Funktionen stabil. Die Anforderung der Stabilität kann jedoch für Scoring-Funktionen erfüllt werden. Darüber hinaus wird allen anderen Anforderungen an eine gewichtete Scoring-Funktion nachgekommen.

Aus Tabelle 4.1 geht hervor, dass neben der Gewichtungsformel nur noch der Ansatz von Oracle die Anforderungen an eine gewichtete Scoring-Funktion erfüllt. In den anderen Fällen werden zum Teil die speziellen Eigenschaften der zu Grunde liegenden Scoring-Funktion nicht von der gewichteten Scoring-Funktion aufrecht erhalten. So ist oftmals das *gewichtete Minimum* im Gegensatz zum *Minimum* beispielsweise nicht mehr idempotent, streng monoton oder translationserhaltend. Das *gewichtete Minimum*, welches mit dem Transformationsschema generiert wurde, ist idempotent und kann einzig die Eigenschaft der Translationserhaltung nicht erfüllen. Alle anderen untersuchten Funktionen weisen diesbezüglich mehr Defizite auf. Eng mit der Translationserhaltung verbunden ist die Erfüllung der Forderung nach lokaler Linearität. So ist kein *gewichtetes Minimum*, das nicht translationserhaltend ist, lokal linear.

Damit wird deutlich, dass die Forderung nach lokaler Linearität ein sehr strenges Kriterium für eine gewichtete Scoring-Funktion ist. Dies wird nur von der Gewichtungsfor-

mel und der Scoring-Funktion, wie sie im System Oracle eingesetzt wird, erfüllt. Letztere nimmt hier eine gesonderte Stellung ein, da eine neutrale Kombination der Ähnlichkeitswerte beziehungsweise in diesem speziellen Fall der Distanzwerte erfolgt. Liegen der Funktion Ähnlichkeitswerte zu Grunde, so werden die Grenzbedingungen erfüllt.

Zusammenfassend kann festgestellt werden, dass für die gewichtete Aggregation von Relevanzwerten gewichtete Scoring-Funktionen eingesetzt werden sollten, die mit der Gewichtungsformel erzeugt werden. Denn nur dieser Ansatz kann die Erfüllung aller Anforderungen an eine gewichtete Scoring-Funktion gewährleisten. Zu beachten ist, dass der *gewichtete Durchschnitt*, wie er im Ansatz von Oracle Anwendung findet, mit der Gewichtungsformel erzeugt werden kann.

4.3 Multi-Level-Gewichtungsmodell

In dem Basisanfragemodell wurden nur einfache Anfragen betrachtet. Daneben sind jedoch insbesondere beim Multimedia-Retrieval auch komplexe Anfragen von Bedeutung. Abbildung 4.4 zeigt eine komplexe Anfrage, welche verschiedene Ebenen E_l mit $l = 0, \ldots, m$ umfasst. Die Gesamtzahl der Ebenen entspricht $m + 1$. Jede Ebene l enthält k verschiedene Teilanfragen $s_{l,j}$ mit $j = 1, \ldots, k$. Jede Teilanfrage besteht aus mindestens zwei Operanden. Dies sind entweder wieder Teilanfragen oder atomare Anfrageterme.

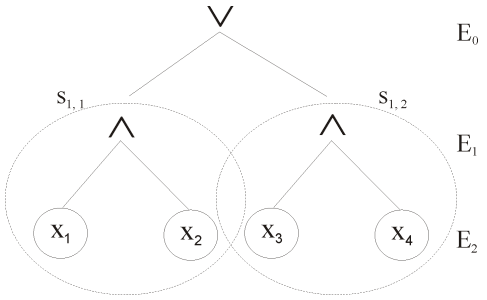


Abbildung 4.4: Anfragebaum einer komplexen Anfrage mit den Ebenen E_l und den Teilanfragen $s_{l,j}$

Insbesondere bei komplexen Multimedia-Anfragen ist neben der Spezifikation der Suchkriterien eine Abbildung der Nutzerpräferenzen durch eine Relevanzgewichtung der Anfrageterme wünschenswert. Im Basisanfragemodell für einfache Anfragen wird die Gewichtung der atomaren Anfrageterme x_i unterstützt. In komplexen Anfragen ist diese Art der Gewichtung für den Nutzer zu restriktiv, denn dieser muss in diesem Fall wissen, welche Auswirkungen die Gewichtung der Atome für die gesamte Anfrage hat. Dies ist jedoch auf Grund der Komplexität der Anfrage nicht trivial und stellt somit zu hohe Anforderungen an den Nutzer. Daher muss eine weitere Möglichkeit der Spezifikation von Relevanzgewichten in komplexen Anfragen angeboten werden, die sowohl die atomaren Anfrageterme als auch die Teilanfragen selbst in den Gewichtungsprozess einbezieht. Im Folgenden werden zwei Alternativen der Formulierung von Relevanzgewichten in komplexen Anfragen vorgestellt.

4.3.1 Alternativen der Spezifikation von Relevanzgewichten

Eine reine Gewichtung der atomaren Anfrageterme in komplexen Anfragen schränkt den Nutzer bei der Anfrageformulierung zu sehr ein. Daher muss hier, im Gegensatz zu einfachen Anfragen, zusätzlich eine Relevanzgewichtung der Teilanfragen möglich sein. Es werden die folgenden zwei Möglichkeiten unterschieden (Schmitt u. a., 2002):

Implizites Gewichten: Der Nutzer ordnet, wie auch bei einfachen Anfragen, jedem atomaren Term x_i mit $i = 1, \ldots, n$ ein Gewicht $\theta_i \in [0, 1]$ zu. Dabei gilt als globale Beschränkung $\sum_{i=1}^{n} \theta_i = 1$. Unter Berücksichtigung der gegebenen Gewichte für die atomaren Terme sind die Gewichte für die Teilanfragen $s_{l,j}$ auf einer höheren Ebene damit implizit bestimmt. Das Gewicht für eine Teilanfrage $s_{l,j}$ wird mittels einer Funktion f aus den Gewichten für die Kinder der Teilanfrage berechnet. Die Funktion $child()$ ermittelt die direkten Kinder einer Teilanfrage $s_{l,i}$. Für die Bestimmung der Gewichte von Atomen oder Teilanfragen wird die Funktionen $weight()$ verwendet. Einer Teilanfrage $s_{l,j}$ wird das Gewicht $\theta_{l,j} = f(weight(child(s_{l,j})))$ zugeordnet. Dies ist in Abbildung 4.5a dargestellt. Die implizite Gewichtung wird mit Θ bezeichnet.

Explizites Gewichten: Der Nutzer spezifiziert explizit ein Gewicht θ_i für jeden atomaren Term x_i und ein Gewicht $\theta_{l,j}$ für jede Teilanfrage $s_{l,j}$. Dies ist in Abbildung 4.5b dargestellt. Die explizite Gewichtung wird mit Θ^E bezeichnet. Im Gegensatz zum impliziten Gewichten gibt es lokale Beschränkungen. Für jede Teilanfrage $s_{l,j}$ gilt, dass sich die Gewichte ihrer Kinder zu 1 summieren:

$$\forall s_{l,j} : \sum_{\theta_{l,t} \in weight(child(s_{l,j}))} \theta_{l,t} = 1.$$

Auf Grund der lokalen Beschränkung entspricht das Gewicht für die Teilanfrage $s_{0,1}$ immer 1, so dass das Gewicht $\theta_{0,1}$ nicht mit in die Gewichtung Θ^E aufgenommen wird.

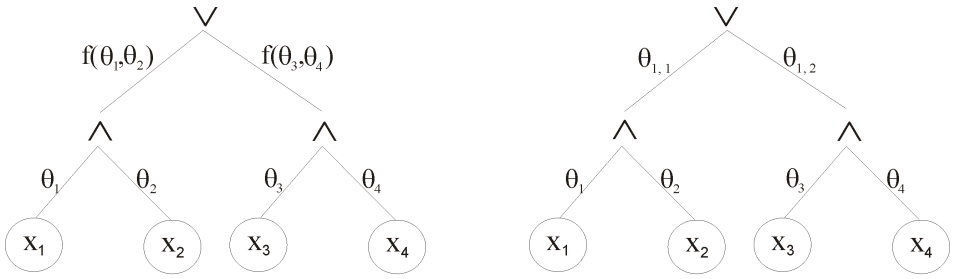


Abbildung 4.5: Vergabe von Relevanzgewichten in einer komplexen Anfrage entsprechend a) der impliziten Variante mit der Gewichtung $\Theta = \{\theta_1, \theta_2, \theta_3, \theta_4\}$ und b) der expliziten Variante mit der Gewichtung $\Theta^E = \{\theta_1, \theta_2, \theta_3, \theta_4, \theta_{1,1}, \theta_{1,2}\}$

Beide Alternativen der Formulierung von Gewichten unterscheiden sich in der Art der Spezifikation der Gewichte sowie in der Anzahl der zu Grunde liegenden Beschränkungen. Jedoch verfügen sie über die gleiche Mächtigkeit. Die Gewichtung der Anfrageterme und

Teilanfragen kann vom Anwender ohne Berücksichtigung der Bedingungen bei der Anfrageformulierung vorgenommen werden. Intern erfolgt eine Normalisierung der Gewichte, so dass die lokalen beziehungsweise globalen Beschränkungen eingehalten werden. Anhand eines kleinen Beispiels wird die Überführung der impliziten Gewichtung in die explizite Gewichtung verdeutlicht.

Beispiel 4.3 *Die Anfrage:*

Suche alle Bilder, die ein rotes Flugzeug oder ein blaues Auto enthalten.

entspricht bei der Zuordnung $x_1 : (Farbe = rot)$, $x_2 : (Form = Flugzeug)$, $x_3 : (Farbe = blau)$ *und* $x_4 : (Form = Auto)$ *dem in Abbildung 4.4 dargestellten Anfragebaum. Der Nutzer spezifiziert die implizite Gewichtung* $\Theta = \{1/6, 3/6, 1/12, 3/12\}$. *Unter Verwendung der Summenfunktion ergibt sich für die Teilanfrage* $s_{1,1}$ *das Gewicht* $\theta_{1,1} = 2/3$ *und analog dazu für* $s_{1,2}$ *das Gewicht* $\theta_{1,2} = 1/3$. *Damit ist ein rotes Flugzeug von größerer Relevanz für den Nutzer ist als ein blaues Auto. Nach der Normalisierung ergibt sich die explizite Gewichtung* $\Theta^E = \{0.25, 0.75, 0.25, 0.75, 2/3, 1/3\}$.

Eine einfache Anfrage besteht in diesem Modell aus nur einer Ebene. Damit kann nur eine Relevanzgewichtung der einzelnen Anfrageterme erfolgen. Dies korrespondiert zur impliziten Gewichtung. Damit stellen einfache Anfragen einen Sonderfall der komplexen Anfragen dar.

4.3.2 Gewichtete Multi-Level-Scoring-Funktion für komplexe Anfragen

Für die Evaluierung komplexer, gewichteter Multimedia-Anfragen werden ebenfalls gewichtete Scoring-Funktionen eingesetzt. Diese werden als *gewichtete Multi-Level-Scoring-Funktionen* bezeichnet. Sie sind eine Erweiterung der bekannten gewichteten Scoring-Funktionen.

Auf Grund einer Gewichtung auf mehreren Ebenen einer komplexen Anfrage müssen die Anforderungen, die den Einfluss der Gewichte beschreiben, neu formuliert werden. Unter Berücksichtigung dieser Anforderungen wird daher eine geeignete gewichtete Multi-Level-Scoring-Funktion vorgestellt.

Anforderungen

Die Anforderungen **A1** (S^Θ basiert auf S) und **A4** (lokale Linearität) gelten uneingeschränkt für gewichtete Multi-Level-Scoring-Funktionen. Somit wird auch für diese Funktionen die Erfüllung der Grenzbedingungen, Monotonie, strenge Monotonie und Stetigkeit gefordert. Darüber hinaus muss für die einzelnen Teilanfragen auch die lokale Linearität gewährleistet bleiben.

Die Anforderungen **A2** (Gleichgewichtung) und **A3** (Nullgewichtung eines Terms), die also den Einfluß der Gewichte berücksichtigen, können wie folgt angepasst werden:

A2: Sind für alle Teilanfragen einer komplexen Anfrage alle Gewichte innerhalb einer Teilanfrage gleich, so hat kein Anfrageterm beziehungsweise keine Teilanfrage einen größeren Einfluß innerhalb einer Teilanfrage als irgendein anderer Term beziehungsweise eine andere Teilanfrage. Dies entspricht dem ungewichteten Fall, wie in Abbildung 4.6 dargestellt. Die gewichtete Scoring-Funktion ist in diesem Fall äquivalent zur ungewichteten Scoring-Funktion.

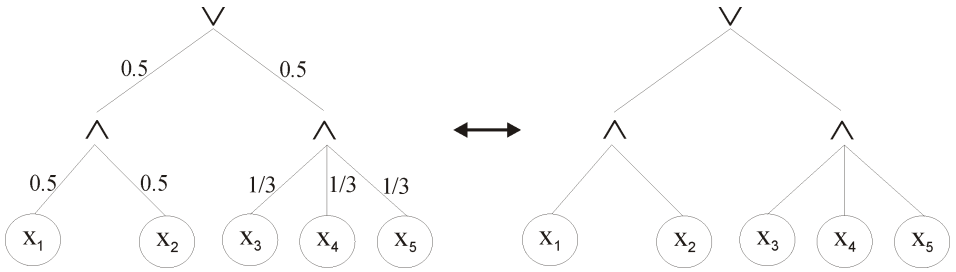


Abbildung 4.6: Eine Gleichgewichtung aller Anfrageterme innerhalb der Teilanfragen resultiert in einer ungewichteten Anfrage

A3: Wird ein atomarer Anfrageterm mit *null* gewichtet, dann ist dieser Term nicht von Interesse. Wird eine Teilanfrage mit *null* gewichtet, so werden alle in ihr vorkommenden Teilanfragen beziehungsweise Anfrageterme ebenfalls mit *null* gewichtet. Da ein mit *null* gewichteter Anfrageterm keinen Einfluß auf die Berechnung des Relevanzwertes hat, kann dieser von der Anfrage entfernt werden. Dies ist in Abbildung 4.7 dargestellt.

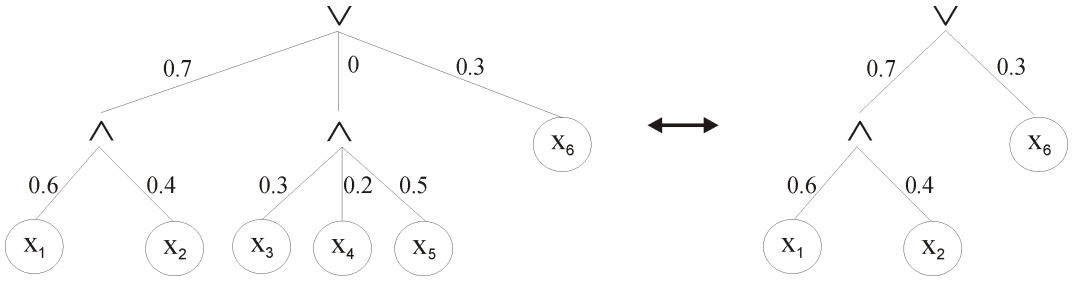


Abbildung 4.7: Eliminierung einer mit *null* gewichteten Teilanfrage

In dem Multi-Level-Gewichtungsmodell wird als Initialgewichtung für komplexe oder einfache Anfragen eine Gleichgewichtung aller Anfrageterme beziehungsweise Teilanfragen gewählt. Auf Grund der Anforderung **A2** entspricht dies einer ungewichteten Anfrage. Alternativ kann die Festlegung der Initialgewichte durch Verwendung von Nutzerprofilen erfolgen. Darüber hinaus besteht auf Grund der Normalisierung der Gewichte für den Nutzer die Möglichkeit, nur einige Anfrageterme oder Teilanfragen beliebig zu gewichten.

Gewichtete Multi-Level-Scoring-Funktion

Komplexe Anfragen setzten sich aus verschiedenen Teilanfragen zusammen. Diese wiederum entsprechen einfachen Anfragen, so dass für die Evaluierung komplexer Anfragen prinzipiell auf bestehende gewichtete Scoring-Funktionen zurückgegriffen werden kann. Wie die Untersuchung gewichteter Scoring-Funktionen in Abschnitt 4.2 zeigte, werden die mit der Gewichtungsformel erzeugten Scoring-Funktionen allen spezifizierten Anforderungen gerecht. Alle anderen untersuchten Scoring-Funktionen weisen demgegenüber gravierende

Defizite auf. Aus diesem Grund wird an dieser Stelle die Gewichtungsformel für die Generierung gewichteter Multi-Level-Scoring-Funktionen genutzt.

Eine Überführung einer Multi-Level-Scoring-Funktion für komplexe Anfragen, wie beispielsweise

$$S(\mu_1(o), \mu_2(o), \mu_3(o), \mu_4(o)) = max(min(\mu_1(o), \mu_2(o)), min(\mu_3(o), \mu_4(o)))$$

für die Evaluierung der Anfrage aus Beispiel 4.3, kann jedoch nicht direkt durch Anwendung der Gewichtungsformel erzeugt werden. Aus diesem Grund wird eine gewichtete Multi-Level-Scoring-Funktion für komplexe Anfragen durch eine *rekursive Anwendung* gewichteter Scoring-Funktionen umgesetzt. Dafür ist es notwendig, dass eine explizite Gewichtung vorliegt, beziehungsweise eine implizite Gewichtung in die explizite Form überführt wird. Für die obige Multi-Level-Scoring-Funktion ergibt sich somit die folgende gewichtete Multi-Level-Scoring-Funktion:

$$\begin{aligned} S(\mu_1(o), \ldots, \mu_4(o), \theta_1, \ldots, \theta_4, \theta_{1,1}, \theta_{1,2}) = {} & (\theta_{1,1} - \theta_{1,2})((\theta_1 - \theta_2)\mu_1(o) + 2\theta_2 min(\mu_1(o), \mu_2(o))) \\ & +2\theta_{1,2} max((\theta_1 - \theta_2)\mu_1(o) + 2\theta_2 min(\mu_1(o), \mu_2(o)), \\ & (\theta_3 - \theta_4)\mu_3(o) + 2\theta_4 min(\mu_3(o), \mu_4(o))). \end{aligned}$$

4.3.3 Transformationsregeln für komplexe, gewichtete Anfragen

Insbesondere bei komplexen Anfragen ist der Aspekt der Anfragetransformation im Rahmen der Anfragebearbeitung für die Optimierung wichtig. Durch die syntaktische Veränderung der internen Darstellung der Anfrage ist eine Minimierung des Ressourcenverbrauchs und damit eine performante Anfragebearbeitung möglich. Beispielsweise kann eine Anfrage syntaktisch so umgeformt werden, dass der Relevanzwert für eine Teilanfrage effizient in einem System und der Relevanzwert für eine andere Teilanfrage in einem anderen System ermittelt werden kann. Eine syntaktische Transformation von Anfragen setzt voraus, dass die Evaluierung sowohl der ursprünglichen als auch der transformierten Anfrage immer das selbe Ergebnis liefert.

Grundlage für die Transformation von Anfragen bilden logische Transformationsregeln. Für eine ungewichtete Formel der Booleschen beziehungsweise der Fuzzy-Logik müssen die allgemeinen Transformationsregeln – wie Kommutativität, Assoziativität, Distributivität, De Morgan, Idempotenz und Involution – gelten (Bookstein, 1978). Für Formeln der Fuzzy-Logik sind diese Regeln nur für die Normen *min* und *max* gültig.

Die Transformation ungewichteter Anfragen beruht auf diesen Transformationsregeln. Sie ist daher ohne Probleme möglich. Demgegenüber stellt die Transformation von gewichteten Anfragen bisher jedoch ein offenes Problem dar. Bei der Anfrageevaluierung muss nun zusätzlich der Einfluss der Gewichte berücksichtigt werden. Aus diesem Grund werden Nutzer in verschiedenen Retrieval-Systemen gezwungen, ihre Anfragen in der disjunktiven Normalform (DNF) beziehungsweise der konjunktiven Normalform (KNF) zu formulieren. Erst dann kann eine Gewichtung der Anfrageterme vorgenommen werden (Herrera-Viedma, 2001b; Pasi, 1999). Diese Herangehensweise ist für den Nutzer nicht praktikabel. Vielmehr sollte der Nutzer die Freiheit besitzen, seinen Informationsbedarf in einer beliebigen Anfrageform auszudrücken. Die Last der logischen Umformung sollte nicht dem Nutzer sondern dem Retrieval-System im Rahmen der Anfrageoptimierung auferlegt werden. Auf eine

Möglichkeit, wie eine solche Transformation durch das System erfolgen kann, wird hier eingegangen. Zum Teil wurden diese Transformationsregeln in (Schulz und Schmitt, 2003a,b,c) veröffentlicht.

Für die Evaluierung komplexer, gewichteter Anfragen wird auf eine gewichtete Multi-Level-Scoring-Funktion, die unter Verwendung der Gewichtungsformel erzeugt wird, zurückgegriffen. Konkret wird eine Konjunktion durch die T-Norm *min* und eine Disjunktion durch die T-Conorm *max* umgesetzt. Bei komplexen Anfragen wird davon ausgegangen, dass die Gewichtung in expliziter Form vorliegt.

Für einfache, gewichtete Anfragen gelten die Eigenschaften Kommutativität, Involution, De Morgan und Idempotenz. Die Regeln werden im Folgenden dargestellt. Die Gültigkeit dieser Transformationsregeln wird in Anhang C (Abschnitte C.1, C.2, C.3) bewiesen.

Kommutativität

Es gilt:

$$\begin{aligned}((x_1,\theta_1)\wedge(x_2,\theta_2)) &= ((x_2,\theta_2)\wedge(x_1,\theta_1)),\\ ((x_1,\theta_1)\vee(x_2,\theta_2)) &= ((x_2,\theta_2)\vee(x_1,\theta_1)).\end{aligned}$$

Bei der Berechnung des Gesamtrelevanzwertes durch eine gewichtete Scoring-Funktion werden unter Ausnutzung der Kommutativität die Anfragen so umgeformt, dass die Bedingung der Gewichtungsformel $\theta_1 \geq \ldots \geq \theta_n$ erfüllt ist. Somit gilt die Kommutativität (Fagin und Wimmers, 1997).

Idempotenz

Wenn $x_1 = x_2$, dann gilt:

$$\begin{aligned}((x_1,\theta_1)\wedge(x_2,\theta_2)) &= x_1 = x_2,\\ ((x_1,\theta_1)\vee(x_2,\theta_2)) &= x_1 = x_2.\end{aligned}$$

Involution

Es gilt:

$$\neg\neg(x_1,\theta_1) = (x_1,\theta_1).$$

De Morgan

Es gilt:

$$\begin{aligned}\neg((x_1,\theta_1)\wedge(x_2,\theta_2)) &= ((\neg x_1,\theta_1)\vee(\neg x_2,\theta_2)),\\ \neg((x_1,\theta_1)\vee(x_2,\theta_2)) &= ((\neg x_1,\theta_1)\wedge(\neg x_2,\theta_2)).\end{aligned}$$

Im Gegensatz zu einfachen, gewichteten Anfragen sind Distributivität und Assoziativität bei komplexen, gewichteten Anfragen von Bedeutung. Um bei diesen Anfragen die Distribu-

tivität und Assoziativität zu gewährleisten, ist es notwendig, die Gewichte zu modifizieren. Im Folgenden wird auf diese Transformationsregeln näher eingegangen.

Distributivität

Es gilt:

$$\begin{aligned}((((x_1,\theta_1)\wedge(x_2,\theta_2)),\theta_{1,1})\vee(x_3,\theta_3)) &= ((((x_1,\theta_{1'})\vee(x_3,\theta_{3'})),\theta_{1,1'})\wedge(((x_2,\theta_{2'})\vee(x_3,\theta_{3''})),\theta_{1,2'})),\\ ((((x_1,\theta_1)\vee(x_2,\theta_2),\theta_{1,1}))\wedge(x_3,\theta_3)) &= ((((x_1,\theta_{1'})\wedge(x_3,\theta_{3'})),\theta_{1,1'})\vee(((x_2,\theta_{2'})\wedge(x_3,\theta_{3''})),\theta_{1,2'})).\end{aligned}$$

Nun müssen die einzelnen Gewichte $\theta_{1'},\theta_{2'},\theta_{3'},\theta_{3''}$ und $\theta_{1,1'}$ sowie $\theta_{1,2'}$ ermittelt werden. Dafür wird in den nachfolgenden Betrachtungen zwischen einer *und/oder*-Anfrage ($X_{\wedge,\vee}$) sowie einer *oder/und*-Anfrage ($X_{\vee,\wedge}$) unterschieden. Wichtig bei der logischen Transformation von Anfragen ist, dass sowohl für die ursprüngliche Anfrage X als auch für die transformierte Anfrage X' gleiche Ergebnisse ermittelt werden. Aus diesem Grund wird bei der distributiven Transformation einer komplexen Anfrage die Modifikation der Gewichte erforderlich. Es wird angenommen, dass $\theta_1 \geq \theta_2$ und $\theta_{1,1} \geq \theta_3$ gilt. Auf Grund der Kommutativität ist dies ohne Einschränkung der Allgemeingültigkeit möglich.

Für die Transformation $X \Rightarrow X'$ kann die Gewichtung $\Theta^{E'}$ für die transformierte Anfrage X' wie folgt aus den Gewichten der ursprünglichen Anfrage bestimmt werden:

$$\theta_{3'} = \theta_3$$

$$\theta_{1,2'} = \begin{cases} \frac{1-(1-2\theta_3)(1-2\theta_2)}{2} & ;\text{ Bedingung 1} \\ \theta_2 & ;\text{ sonst} \end{cases}$$

$$\theta_{3''} = \begin{cases} \frac{\theta_3}{1-(1-2\theta_3)(1-2\theta_2)} & ;\text{ Bedingung1} \\ 0 & ;\text{ Bedingung 2} \\ \theta_3 & ;\text{ sonst} \end{cases}$$

Bedingung 1:
$X_{\wedge,\vee} : \mu_1 \geq \mu_3 \geq \mu_2 \wedge (1-2\theta_2)\mu_1 + 2\theta_2\mu_2 < \mu_3$
$X_{\vee,\wedge} : \mu_2 \geq \mu_3 \geq \mu_1 \wedge (1-2\theta_2)\mu_1 + 2\theta_2\mu_2 > \mu_3$
Bedingung 2:
$X_{\wedge,\vee} : \mu_1 \geq \mu_3 \geq \mu_2 \wedge (1-2\theta_2)\mu_1 + 2\theta_2\mu_2 \geq \mu_3$
$X_{\vee,\wedge} : \mu_2 \geq \mu_3 \geq \mu_1 \wedge (1-2\theta_2)\mu_1 + 2\theta_2\mu_2 \leq \mu_3$

Auf Grund der lokalen Beschränkungen können die Gewichte $\theta_{1'},\theta_{2'}$ sowie $\theta_{1,1'}$ aus den bereits ermittelten Gewichten berechnet werden, zum Beispiel $\theta_{1'} = 1-\theta_{3'}$. Die Gültigkeit dieser Transformationsregeln wird in Anhang C.4 belegt.

Analog zur Transformation von $X \Rightarrow X'$ müssen für eine Rücktransformation, also $X' \Rightarrow X$, die Gewichte modifiziert werden. Hier wird zwischen einer ursprünglichen Anfrage in

KNF ($X'_\wedge$) und einer entsprechenden Anfrage in DNF ($X'_\vee$) unterschieden. Es wird angenommen, dass $\theta_{1'} \geq \theta_{3'}$, $\theta_{2'} \geq \theta_{3''}$ und $\theta_{1,1'} \geq \theta_{1,2'}$ gilt. Bei einer Transformation der Anfrage X' müssen die ursprünglichen sechs Gewichte auf vier zurückgeführt werden. Die Gewichte θ_1 und θ_3 können, wie im Folgenden angegeben, berechnet werden. Die restlichen Gewichte θ_2 und $\theta_{1,1}$ werden auf Grund der lokalen Beschränkungen ermittelt. Die Gewichte für die transformierte Anfrage ergeben sich wie folgt:

$$\theta_3 = \begin{cases} 2\theta_{1,2'}\theta_{3''} & \text{; Bedingung 5 + 1, Bedingung 7} \\ 1 - \frac{1-2\theta_{1,2'}2\theta_{3''}}{2} & \text{; Bedingung 6 + 1, Bedingung 8} \\ (1-2\theta_{1,2'})\theta_{3'} + 2\theta_{1,2'}\theta_{3''} & \text{; Bedingung 3, Bedingung 2} \\ \theta_{3'} & \text{; sonst} \end{cases}$$

$$\theta_1 = \begin{cases} \frac{(1-2\theta_{1,2'})(1-2\theta_{3'})}{2(1-2\theta_3)} & \text{; Bedingung 2 + 9} \\ 1 & \text{; Bedingung 4} \\ 1 - \frac{\theta_{1,2'}(1-2\theta_{3''})}{(1-2\theta_3)} & \text{; Bedingung 3 + 9, Bedingung 5 + 9} \\ 1 - \frac{\theta_{1,2'}(1-2\theta_{3''})}{2(1-\theta_3)} & \text{; Bedingung 6 + 9} \\ 0.5 & \text{; Bedingung 3 + 10, Bedingung 2 +10, Bedingung 11 + 10} \\ \theta_{1,1'} & \text{; sonst} \end{cases}$$

Bedingung 1:
$2\theta_{1,2'}\theta_{3''} \neq 0.5$
Bedingung 2:
$X'_\wedge : \mu_3 \geq \mu_2 \geq \mu_1 \wedge (1-2\theta_{3''})\mu_2 + 2\theta_{3''}\mu_3 \leq (1-2\theta_{3'})\mu_1 + 2\theta_{3'}\mu_3$
$X'_\vee : \mu_1 \geq \mu_2 \geq \mu_3 \wedge (1-2\theta_{3''})\mu_2 + 2\theta_{3''}\mu_3 > (1-2\theta_{3'})\mu_1 + 2\theta_{3'}\mu_3$
Bedingung 3:
$X'_\wedge : \mu_3 \geq \mu_1 \geq \mu_2 \wedge (1-2\theta_{3''})\mu_2 + 2\theta_{3''}\mu_3 \leq (1-2\theta_{3'})\mu_1 + 2\theta_{3'}\mu_3$
$X'_\vee : \mu_2 \geq \mu_1 \geq \mu_3 \wedge (1-2\theta_{3''})\mu_2 + 2\theta_{3''}\mu_3 > (1-2\theta_{3'})\mu_1 + 2\theta_{3'}\mu_3$
Bedingung 4:
$X'_\wedge : \mu_3 \geq \mu_1 \geq \mu_2 \wedge (1-2\theta_{3''})\mu_2 + 2\theta_{3''}\mu_3 > (1-2\theta_{3'})\mu_1 + 2\theta_{3'}\mu_3$
$X'_\vee : \mu_2 \geq \mu_1 \geq \mu_3 \wedge (1-2\theta_{3''})\mu_2 + 2\theta_{3''}\mu_3 \leq (1-2\theta_{3'})\mu_1 + 2\theta_{3'}\mu_3$
Bedingung 5:
$X'_\wedge : \mu_1 \geq \mu_3 \geq \mu_2 \wedge (1-2\theta_{1,2'}(1-2\theta_{3''})/(1-4\theta_{3''}\theta_{1,2'}))\mu_1 + 2\theta_{1,2'}(1-2\theta_{3''})/(1-4\theta_{3''}\theta_{1,2'})\mu_2 < \mu_3$
$X'_\vee : \mu_2 \geq \mu_3 \geq \mu_1 \wedge (1-2\theta_{1,2'}(1-2\theta_{3''})/(1-4\theta_{3''}\theta_{1,2'}))\mu_1 + 2\theta_{1,2'}(1-2\theta_{3''})/(1-4\theta_{3''}\theta_{1,2'})\mu_2 \geq \mu_3$
Bedingung 6:
$X'_\wedge : \mu_1 \geq \mu_3 \geq \mu_2 \wedge (1-2\theta_{1,2'}(1-2\theta_{3''})/(1-4\theta_{3''}\theta_{1,2'}))\mu_1 + 2\theta_{1,2'}(1-2\theta_{3''})/(1-4\theta_{3''}\theta_{1,2'})\mu_2 \geq \mu_3$
$X'_\vee : \mu_2 \geq \mu_3 \geq \mu_1 \wedge (1-2\theta_{1,2'}(1-2\theta_{3''})/(1-4\theta_{3''}\theta_{1,2'}))\mu_1 + 2\theta_{1,2'}(1-2\theta_{3''})/(1-4\theta_{3''}\theta_{1,2'})\mu_2 < \mu_3$

Bedingung 7:
$X'_{\wedge} : \mu_1 \geq \mu_3 > \mu_2 \wedge \theta_{1,2'} = \theta_{3''} = 0.5$
$X'_{\vee} : \mu_2 \geq \mu_3 = \mu_1 \wedge \theta_{1,2'} = \theta_{3''} = 0.5$
Bedingung 8:
$X'_{\wedge} : \mu_1 \geq \mu_3 = \mu_2 \wedge \theta_{1,2'} = \theta_{3''} = 0.5$
$X'_{\vee} : \mu_2 \geq \mu_3 > \mu_1 \wedge \theta_{1,2'} = \theta_{3''} = 0.5$
Bedingung 9:
$\theta_3 \neq 0.5$
Bedingung10:
$\theta_3 = 0.5$
Bedingung 11:
$X'_{\wedge} : \mu_1 \geq \mu_3 \geq \mu_2$
$X'_{\wedge} : \mu_1 \geq \mu_3 \geq \mu_2$

Assoziativität

Es gilt:

$$\begin{aligned}((((x_1,\theta_1) \wedge (x_2,\theta_2)),\theta_{1,1}) \wedge (x_3,\theta_3)) &= ((x_1,\theta_{1'}) \wedge (((x_2,\theta_{2'}) \wedge (x_3,\theta_{3'})),\theta_{1,1'})),\\ ((((x_1,\theta_1) \vee (x_2,\theta_2)),\theta_{1,1}) \vee (x_3,\theta_3)) &= ((x_1,\theta_{1'}) \vee (((x_2,\theta_{2'}) \vee (x_3,\theta_{3'})),\theta_{1,1'})).\end{aligned}$$

Ebenso wie bei einer distributiven Transformation der Anfrage müssen auch bei der assoziativen Transformation Gewichte modifiziert werden. Zunächst wird die Transformation $X \Rightarrow X'$ mit $\theta_1 \geq \theta_2$ und $\theta_{1,1} \geq \theta_3$ durchgeführt. Die Berechnung der Gewichte für die transformierte Anfrage X' ist nachfolgend beschrieben. Die Gewichte $\theta_{1'}$ und $\theta_{2'}$ können auf Basis der lokalen Beschränkungen der expliziten Gewichtung ermittelt werden. Somit gilt:

$$\theta_{1,1'} = \begin{cases} \theta_3 & \text{; Bedingung 1} \\ \frac{1-(1-2\theta_3)(1-2\theta_2)}{2} & \text{; Bedingung 2, Bedingung 3} \\ \theta_2 & \text{; sonst} \end{cases}$$

$$\theta_{3'} = \begin{cases} 1 & \text{; Bedingung 1} \\ \frac{\theta_3}{2\theta_{1,1'}} & \text{; Bedingung 2} \\ 1 - \frac{(1-2\theta_3)\theta_2}{2\theta_{1,1'}} & \text{; Bedingung 3} \\ \theta_2 & \text{; sonst} \end{cases}$$

Bedingung 1:
$X_{\wedge} : \mu_2 \geq \mu_1 \geq \mu_3$
$X_{\vee} : \mu_3 \geq \mu_1 \geq \mu_2$

Bedingung2:
$X_\wedge : \mu_1 \geq \mu_2 \geq \mu_3$
$X_\vee : \mu_3 \geq \mu_2 \geq \mu_1$
Bedingung 3:
$X_\wedge : \mu_1 \geq \mu_3 \geq \mu_2 \wedge (1-2\theta_2)\mu_1 + 2\theta_2\mu_2 > \mu_3$
$X_\vee : \mu_2 \geq \mu_3 \geq \mu_1 \wedge (1-2\theta_2)\mu_1 + 2\theta_2\mu_2 < \mu_3$

Für die Rücktransformation $X' \Rightarrow X$ gilt $\theta_{1'} \geq \theta_{1,1'}$ und $\theta_{2'} \geq \theta_{3'}$. Die Gewichte θ_2 und θ_3 werden, wie im Folgenden angegeben, berechnet. Die Beweise der Gültigkeit dieser Transformationsregeln sind ebenfalls in Anhang C.5 aufgeführt.

$$\theta_3 = \begin{cases} 2\theta_{1,1'}\theta_{3'} & \text{; Bedingung 1, Bedingung 4} \\ 0 & \text{; Bedingung 2} \\ \theta_{1,1'} & \text{; sonst} \end{cases}$$

$$\theta_2 = \begin{cases} \frac{(1-2\theta_{3'})\theta_{1,1'}}{(1-2\theta_3)} & \text{; Bedingung 1 + 3} \\ 1 - \frac{(1-2\theta_{1,1'})}{2(1-2\theta_3)} & \text{; Bedingung 4 + 3} \\ \theta_{1,1'} & \text{; sonst} \end{cases}$$

Bedingung 1:
$X'_\wedge : \mu_1 \geq \mu_2 \geq \mu_3$
$X'_\vee : \mu_3 \geq \mu_2 \geq \mu_1$
Bedingung 2:
$X'_\wedge : \mu_1 \geq \mu_3 \geq \mu_2$
$X'_\wedge : \mu_2 \geq \mu_1 \geq \mu_3 \wedge (1-2\theta_{3'})\mu_2 + 2\theta_{3'}\mu_3 \geq \mu_1$
$X'_\vee : \mu_2 \geq \mu_3 \geq \mu_1$
$X'_\vee : \mu_3 \geq \mu_1 \geq \mu_2 \wedge (1-2\theta_{3'})\mu_2 + 2\theta_{3'}\mu_3 \leq \mu_1$
Bedingung 3:
$\theta_{3'} \neq \theta_{1,1'} \vee \theta_{3'} = \theta_{1,1'} \neq 0.5$
Bedingung 4:
$X'_\wedge : \mu_2 \geq \mu_1 \geq \mu_3 \wedge (1-2\theta_{3'})\mu_2 + 2\theta_{3'}\mu_3 > \mu_1$
$X'_\vee : \mu_3 \geq \mu_1 \geq \mu_2 \wedge (1-2\theta_{3'})\mu_2 + 2\theta_{3'}\mu_3 > \mu_1$

4.3.4 Gewichtung der Operatoren

Wie verschiedene Studien gezeigt haben, können die klassischen Junktoren $\wedge$ und $\vee$ das Nutzerempfinden nicht angemessen abbilden (Zimmermann und Zysno, 1980; Cooper, 1988; Avrahami und Kareev, 1993). Ferner setzten die Junktoren eine zu utilitaristische Aggregation der Kriterien um und erlauben keine Kompensation der verschiedenen Kriterien (Yager, 1988).

Aus diesen Gründen wird in dem Multi-Level-Gewichtungsmodell neben einer Relevanzgewichtung der Anfrageterme auch eine Gewichtung der Operatoren unterstützt. Für die-

sen Zweck wird der Universaljunktor $\oplus^z$ eingeführt, dessen Verhalten durch das Gewicht $z \in [0,1]$ parametrisiert werden kann. Die Semantik für diesen Junktor wird durch einen kompensatorischen *ODER*-Operator beschrieben (vgl. Abschnitt 2.4.2). Bei einem Operatorgewicht von $z = 1$ ergibt sich somit eine reine Konjunktion und dementsprechend bei einem z-Wert von 0 eine Disjunktion der zu aggregierenden Werte.

Um eine Kombination der Relevanz- und Operatorgewichtung zu ermöglichen, wird der kompensatorische *ODER*-Operator dahingehend erweitert, dass Relevanzgewichte integriert werden. Das Verhalten des Operators liegt somit zwischen den verwendeten gewichteten Scoring-Funktionen für die Konjunktion $S^{\Theta}_{S_\wedge}$ und Disjunktion $S^{\Theta}_{S_\vee}$. Demnach wird der Junktor $\oplus^z$ wie folgt definiert:

$$\begin{aligned}\oplus^z_{(\theta_1,\theta_2,\ldots,\theta_m)}(\mu_1(o),\ldots,\mu_n(o)) &= z * S^{\Theta}_{S_\wedge}(\mu_1(o),\ldots,\mu_n(o),\theta_1,\ldots,\theta_n) + \\ &\quad (1-z) * S^{\Theta}_{S_\vee}(\mu_1(o),\ldots,\mu_n(o),\theta_1,\ldots,\theta_n).\end{aligned}$$

Für die Funktionen $S_\wedge$ und $S_\vee$ können beliebige Scoring-Funktionen, also nicht nur T-Normen und T-Conormen, verwendet werden. Das Verhalten des Operators unter Einsatz der T-Norm *min* und der T-Conorm *max* bei unterschiedlichen Parametrisierungen ist in Abbildung 4.8 dargestellt. Dabei werden die folgenden Charakteristika, die bei Einsatz beliebiger Normen gelten, deutlich:

- Bei einem Operatorgewicht von $z = 1$ ergibt sich eine gewichtete, konjunktive Verknüpfung.
- Bei einem Operatorgewicht von $z = 0$ ergibt sich eine gewichtete, disjunktive Verknüpfung.
- Bei gleichen Relevanzgewichten ergibt sich ein ungewichteter kompensatorischer Operator.
- Bei einer Gewichtung $\Theta = \{1, 0, \ldots, 0\}$ ist das Ergebnis unabhängig vom Operatorgewicht z stets μ_1.

Der gewichtete Universaljunktor ist kommutativ. Er ist jedoch auf Grund des Operatorgewichts nicht mehr distributiv, respektive assoziativ. Basiert der Universaljunktor auf den Funktionen *min* und *max*, so ist der Operator auch idempotent und für $z = 0$ und $z = 1$ wird die Assoziativität und Distributivität, gegebenenfalls unter Verwendung der in Abschnitt 4.3.3 beschriebenen Transformationsregeln, unterstützt. Darüber hinaus gilt:

$$(\neg(\oplus^z_{\theta_1,\theta_2,\ldots,\theta_m}(F_1,F_2,\ldots,F_m))) \Leftrightarrow (\oplus^{1-z}_{\theta_1,\theta_2,\ldots,\theta_m}((\neg F_1),(\neg F_2),\ldots,(\neg F_m))).$$

Als Initialgewicht für den Universaljunktor gilt $z = 0$. Damit wird er als *ODER*-Operator initialisiert.

4.3.5 Schwellwertgewichtung

Das Multi-Level-Gewichtungsmodell unterstützt ferner eine Schwellwertgewichtung. Die Schwellwertgewichte können dabei auf allen Ebenen einer Multimedia-Anfrage spezifiziert

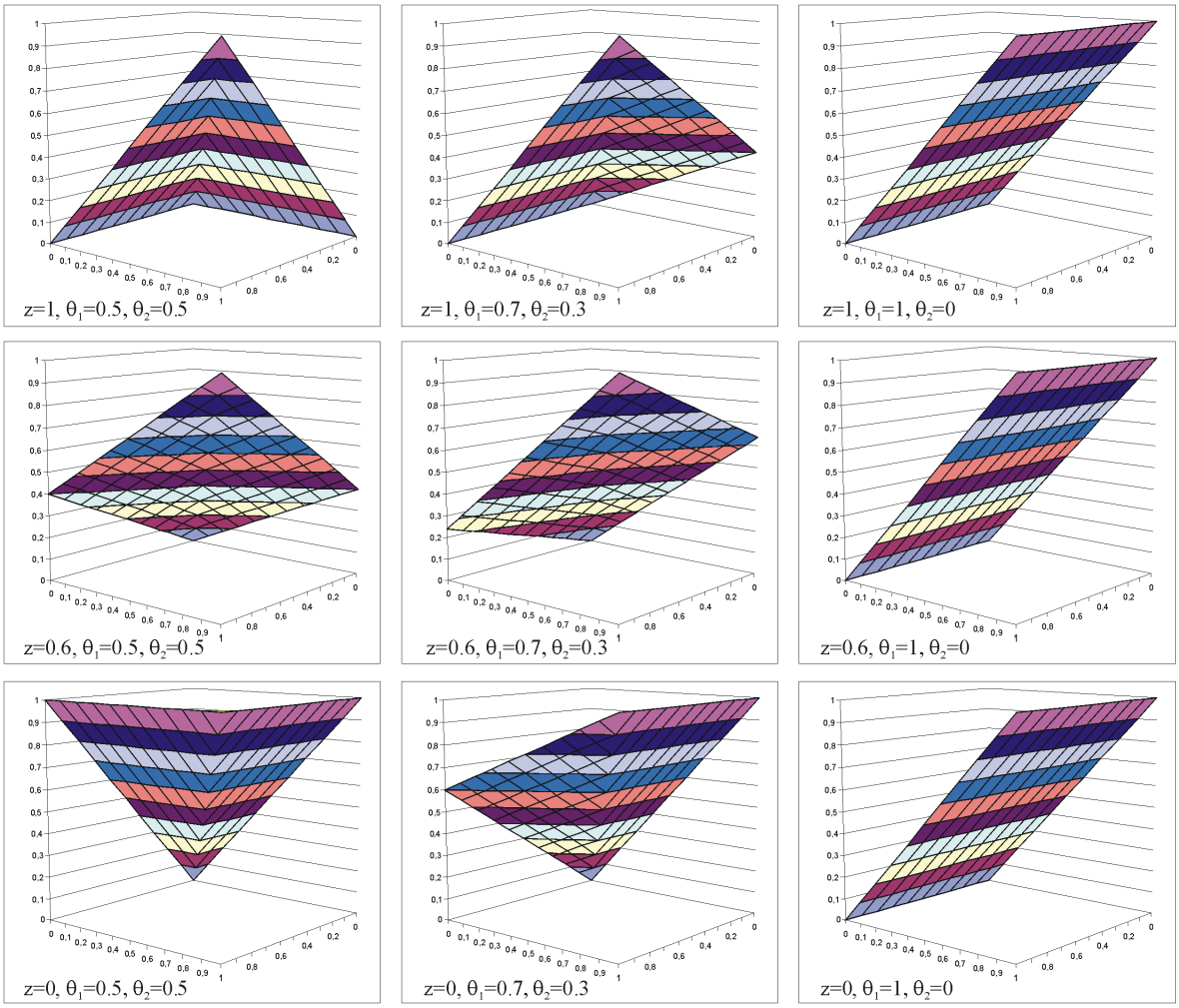


Abbildung 4.8: Aggregationscharakteristika des Universaljunktors

werden. So kann ein Schwellwert aus dem Interval $[0, 1]$ sowohl für eine atomare Anfragebedingung als auch für die gesamte Anfrage spezifiziert werden. Bei komplexen Anfragen ist es darüber hinaus möglich, auch für Teilanfragen einen Schwellwert anzugeben.

Für die Evaluierung von Schwellwertgewichten kann eine der in Abschnitt 2.4.2 beschriebenen Funktionen eingesetzt werden.

4.4 Zusammenfassung

In diesem Kapitel wurde das Basisanfragemodell beschrieben. Es stellt ein Modell für die Relevanzgewichtung einfacher Anfragen dar. Es wurde darauf eingegangen, dass gewichtete Anfragen durch den Einsatz einer Scoring-Funktion evaluiert werden. Dabei handelt es sich um Funktionen, welche die einzelnen Ähnlichkeitswerte eines Objektes zu einem Gesamtrelevanzwert für die Anfrage aggregieren. Um eine adäquate Aggregation der einzelnen

Ähnlichkeitswerte gewährleisten zu können, müssen die Scoring-Funktionen verschiedenen Anforderungen gerecht werden. Dazu zählen beispielsweise die Forderung nach strenger Monotonie und die Einhaltung der Grenzbedingungen. Des Weiteren ergeben sich Anforderungen auf Grund des Einflusses der Relevanzgewichte. So ist es beispielsweise wesentlich, dass Anfrageterme, denen ein Relevanzgewicht von *null* zugeteilt wurde, keinen Einfluss auf die Berechnung des Gesamtrelevanzwertes nehmen dürfen.

Auf Basis dieser Anforderungen an Scoring-Funktionen für einfache Anfragen wurden bestehende Ansätze zur Integration von Relevanzgewichten in Scoring-Funktionen sowie verschiedene gewichtete Scoring-Funktionen untersucht. Es hat sich herausgestellt, dass die wenigsten Ansätze den spezifizierten Anforderungen voll gerecht werden. Positiv hervorzuheben ist dabei die Gewichtungsformel von Fagin und Wimmers (1997), die die Überführung einer beliebigen Scoring-Funktion in eine gewichtete Scoring-Funktion erlaubt. Mit diesem Ansatz wird die Erfüllung aller geforderten Eigenschaften an gewichtete Scoring-Funktionen gewährleistet.

Als Erweiterung des Basisanfragemodells wurde in diesem Kapitel das Multi-Level-Gewichtungsmodell eingeführt. Es ermöglicht sowohl eine Relevanzgewichtung als auch eine Schwellwertgewichtung auf allen Ebenen einer komplexen Anfrage. Diese schließt natürlich einfache Anfragen ein. Für die Evaluierung komplexer, gewichteter Anfragen wird eine Multi-Level-Scoring-Funktion eingesetzt. Hierfür wurden die Anforderungen an die gewichtete Scoring-Funktion modifiziert. Aufbauend auf der Gewichtungsformel wird eine gewichtete Multi-Level-Scoring-Funktion propagiert.

Des Weiteren wurde eine Operatorgewichtung in das Gewichtungsmodell integriert. Für die Umsetzung werden kompensatorische Operatoren genutzt. Um jedoch die Relevanzgewichtung mit der Operatorgewichtung zu kombinieren, fließen die Relevanzgewichte mit in den kompensatorischen Operator ein. Für die Anfrageformulierung werden die klassischen Junktoren $\wedge$ und $\vee$ durch den Universaljunktor $\oplus^z$ ergänzt. Für den gewichteten Universaloperator, der die Relevanz- und Operatorgewichtung vereint, wurden die speziellen Charakteristika dargelegt. Ferner integriert das Multi-Level-Gewichtungsmodell eine Schwellwertgewichtung auf allen Ebenen einer komplexen Anfrage.

Das Multi-Level-Gewichtungsmodell erlaubt es, komplexe Anfragen zu formulieren und dabei die Nutzerpräferenzen flexibel auf allen Ebenen der Anfrage zu spezifizieren. Der Nutzer hat dabei die Möglichkeit, seine Präferenzen durch Verwendung von Relevanz-, Schwellwert- und Operatorgewichten auszudrücken. Dieses ist in dem Ansatz von Herrera-Viedma (2001a,b) nur eingeschränkt möglich. In diesem Ansatz wird auch eine Operatorgewichtung mittels eines OWA-Operators (vgl. Abschnitt 2.4.2) angeboten. Jedoch wird hier eine Schwellwertgewichtung nur für atomare Anfrageterme und eine Relevanzgewichtung nur für Teilanfragen gewährleistet. Damit wird auf jeder Ebene der Anfrage nur eine Art der Gewichtung unterstützt. Mit dem hier vorgeschlagenen Multi-Level-Gewichtungsmodell hat der Nutzer mehr Freiheiten, seine Präferenzen unter Verwendung verschiedener Gewichtungssemantiken zu beschreiben.

5 Ähnlichkeitskalkül und Ähnlichkeitsalgebra

In diesem Kapitel wird eine deklarative kalkülbasierte Multimedia-Anfragesprache vorgestellt. Das *Ähnlichkeitskalkül* (*Similarity Domain Relation Calculus* $\mathcal{SDC}$) bildet die formale Grundlage für die Entwicklung einer nutzerfreundlichen, visuellen Anfragesprache, die in Kapitel 6 eingeführt wird.

Im Folgenden wird die Syntax und die Semantik des Ähnlichkeitskalküls festgelegt. Das Ähnlichkeitskalkül ist eine Erweiterung des relationalen Bereichskalküls. Es wird ferner eine Erweiterung der Relationenalgebra, die sogenannte *Ähnlichkeitsalgebra* (*Similarity Algebra* $\mathcal{SA}$), vorgeschlagen. Damit ist eine Überführung beliebiger Ausdrücke des Kalküls in einen Ausdruck der Ähnlichkeitsalgebra möglich. Dieses wird ebenfalls in diesem Kapitel beschrieben. Das hier vorgestellt Ähnlichkeitskalkül und die entsprechende -algebra basieren auf den grundlegenden Ideen, wie sie von Schmitt und Schulz (2004a,b) beschrieben wurden. Darauf aufbauend wurden einige Erweiterungen und Veränderungen vorgenommen, um beispielsweise die Nutzerfreundlichkeit zu steigern.

Bei der Reduktion eines Kalkülausdrucks in einen entsprechenden Algebraausdruck ist der Aspekt der Sicherheit von Anfragen von Bedeutung. Aus diesem Grund wird zunächst nach einer kurzen Darstellung des Bereichskalküls auf die Problematik der sicheren Anfragen eingegangen.

5.1 Bereichskalkül

Das Bereichskalkül ist, wie auch das Tupelkalkül, ein Relationenkalkül und stellt einen logikbasierten Ansatz zur Formulierung von Datenbankanfragen dar. Im Gegensatz zu prozeduralen Anfragesprachen ist hier die Beschreibung dessen, was zum Anfrageergebnis gehört vordergründig. Die Ergebnisrelation wird beschrieben, ohne explizit eine Vorschrift zur Konstruktion des Ergebnisses vorzugeben. Dabei entsprechen die Datenbankinhalte den Belegungen von Prädikaten und die Anfragen abgeleiteten Prädikaten. Das Ergebnis einer Anfrage ist eine Menge von Tupeln, die eine angegebene Bedingung erfüllen. Das Relationenkalkül beruht auf der Prädikatenlogik erster Stufe. Es ist streng relational vollständig (Heuer und Saake, 2000, S. 324).

Im Bereichskalkül werden als Variablen nur Variablen zu den Basisdatentypen zugelassen. Diese werden als *Bereichsvariablen* bezeichnet. Ein Anfrageausdruck im Bereichskalkül hat die Form:

$$\{X_1, \ldots, X_n | F(X_1, \ldots, X_n)\},$$

wobei jede Bereichsvariable X_i jeweils einem Attribut A_i mit $i = 1, \ldots, n$ einer Relation zugeordnet ist. Die Bereichsvariablen $X_1, \ldots, X_n$ sind die einzigen freien Variablen. F ist eine Formel über diese. Die Wertebereiche der Datentypen der Attribute, welche endlich oder

unendlich sein können, enthalten die potenziellen Ergebniswerte. Die Ergebnismenge der Anfrage ist die Menge von Tupeln, die implizit aus den Wertebereichen der Variablen in der Ergebnisliste konstruiert werden und F erfüllen.

Eine Formel F setzt sich rekursiv aus *Termen* und *Atomen* zusammen. Ein Term t_i wird aus Konstanten und Variablen zu den Datentypen gebildet. Ein Atom ergibt sich aus den folgenden Konstrukten:

- Prädikatanwendung $\delta(t_1, \ldots, t_n)$ mit dem Operator $\delta \in \{<, >, \leq, \geq, =, \neq\}$ und den Termen t_i. Für zweistellige Prädikate wird typischerweise die Infixnotation verwendet.
- Prädikatanwendung für Datenbankprädikate $R(t_1, \ldots, t_n)$ mit dem Relationennamen R und den Termen t_i. Die Terme müssen vom passenden Typ sein. n entspricht der Stelligkeit der Relation.

Eine Formel F ist dann rekursiv gegeben durch:

- ein Atom sowie
- durch eine Verknüpfung von Atomen und Formeln mittels der Junktoren $\wedge$ und $\vee$ sowie durch Verwendung der Negation $\neg$ und der Quantoren $\exists$ und $\forall$. Somit sind $(F_1 \wedge F_2)$, $(F_1 \vee F_2)$, $(\neg F_1)$, $(\exists F_1)$ und $(\forall F_1)$ Formeln, sofern F_1 und F_2 Formeln sind.

Für eine vereinfachte Schreibweise wird die Klammerung oftmals weggelassen. Als Reihenfolge der Ausführung gilt dann: $\forall, \exists, \neg, \wedge, \vee$. Das Beispiel 5.1 zeigt eine Anfrage im Bereichskalkül. Als Grundlage dient die Relation *Flugzeug(Bezeichung, Hersteller, Baujahr, Fertigungsland)*.

Beispiel 5.1 *Gib die Bezeichnung und den Hersteller der Flugzeuge zurück, die vor 1990 in Deutschland produziert wurden:*

$$\{X_1, X_2 | \exists X_3 (Flugzeug(X_1, X_2, X_3, \text{'}Deutschland\text{'}) \wedge X_3 < 1990)\}.$$

5.2 Sichere Anfragen

Die Betrachtung der Sicherheit von Anfragen setzt endliche Relationen voraus. Prinzipiell können Kalkülanfragen sowohl ein endliches als auch ein unendliches Anfrageergebnis liefern. Beispielsweise bestimmt die Bereichskalkülanfrage:

$$\{X | \neg R(X)\}$$

alle möglichen Werte für X, die nicht in der Datenbank sind. Somit ist das Ergebnis natürlich unendlich, sofern der Wertebereich unendlich ist. Auch die Anfrage

$$\{X_1, X_2 | R(X_1) \vee Q(X_2)\}$$

liefert kein endliches Ergebnis, da alle möglichen X_1-Werte gültig sind, sofern $Q(X_2)$ wahr ist und umgekehrt. Ob ein Ergebnis endlich oder unendlich ist, kann vom aktuellen Stand der Datenbank abhängen. Sinnvollerweise sollte Anfragen natürlich ein endliches Ergebnis ermitteln. Anfragen, die dieses garantieren, werden als *sichere Anfragen* bezeichnet. Sie liefern

nur Ergebniswerte zurück, die aus dem Wertebereich des Ausdrucks stammen. Dazu zählen all die Werte, die in den Relationen stehen, die in dem Ausdruck referenziert werden, sowie die Konstanten aus dem Ausdruck. Diese Anfragen werden auch als *bereichsunabhängige Anfragen* bezeichnet. Die Bereichsunabhängigkeit ist im Einzelfall leicht zu zeigen, wie die obigen Beispiele für unsichere Anfragen verdeutlichen. Dies ist jedoch nicht automatisch prüfbar. Daher ist diese Art von Sicherheit im Allgemeinen nicht entscheidbar.

Nach Ullman (1982) liefern *semantisch sichere Anfragen* für jeden Datenbankzustand ein endliches Ergebnis. Semantisch sichere Anfragen entsprechen den bereichsunabhängigen Anfragen. Ullman hat gezeigt, dass zu jedem bereichsunabhängigen Kalkülausdruck ein entsprechender äquivalenter Ausdruck der Relationenalgebra existiert. Dafür wird der Einsatz der Relation $Dom(F)$ notwendig. Diese umfasst alle Konstanten der Anfrage sowie alle Konstanten, die in den entsprechenden Relationen auftreten (Ullman, 1988, S. 151).

Darauf aufbauend gibt Ullman (1988) eine engere Definition für die Sicherheit von Anfragen und führt die *syntaktisch sicheren Anfragen* ein. Diese unterliegen syntaktischen Einschränkungen, damit die semantische Sicherheit garantiert wird. Syntaktisch sichere Anfragen stellen somit eine Teilmenge der semantisch sicheren beziehungsweise bereichsunabhängigen Anfragen dar. Die syntaktische Sicherheit ist entscheidbar. Heuer und Saake fassen die Grundidee wie folgt zusammen (Heuer und Saake, 2000, S. 322):

> Jede freie Variable X_i muss in $F(X_1, \ldots, X_n)$ durch positives Auftreten $X_i = t$ oder $R(\ldots, X_i, \ldots)$ an endliche Bereiche gebunden werden. Die Bindung an endlich Bereiche muss für die ganze Bedingung, insbesondere für die Zweige einer Disjunktion, gelten.

Neben den *syntaktisch sicheren* Anfragen gibt es weitere Subklassen der bereichsunabhängigen Anfragen. Dazu zählen beispielsweise die *evaluierbaren* (Demolombe, 1982, 1992; Gelder und Topor, 1991) sowie die *zulässigen* (Topor, 1987; Gelder und Topor, 1991) Anfragen.

Unter all diesen Alternativen stellen die evaluierbaren Formeln die größte Subklasse der bereichsunabhängigen Anfragen dar (Gelder und Topor, 1991). Sie wurden ursprünglich von Demolombe (1982, 1992) entwickelt. Eine vereinfachte Definition wurde von Gelder und Topor (1991) erarbeitet. Die Autoren beschreiben darüber hinaus, wie evaluierbare Anfragen ausgewertet werden und wie die Überführung dieser in die Relationenalgebra erfolgt. Die zulässigen Anfragen sind eine echte Teilmenge der evaluierbaren Anfragen. Sie können einfacher erkannt und einfacher in die relationale Algebra überführt werden. Aus diesem Grund ist es möglich, dass evaluierbare Anfrageausdrücke zunächst in zulässige Ausdrücke überführt und im Anschluss daran in die relationale Algebra transformiert werden (ebd.).

Da die evaluierbaren Anfragen die größte, syntaktisch überprüfbare Klasse der bereichsunabhängigen Anfragen darstellen, wird die Sicherheit der Ausdrücke im Ähnlichkeitskalkül mittels des Ansatzes von Gelder und Topor festgestellt. Auf Grund der vereinfachten Definition und somit der leichteren Anwendbarkeit wird dieser Ansatz und nicht der Originalvorschlag von Demolombe gewählt. Demnach wird im Folgenden die Evaluierbarkeit einer Formel mittels der Relationen $gen()$ und $con()$ überprüft. In Abschnitt 5.3.2 wird im Zusammenhang mit der Sicherheit von Anfragen im Ähnlichkeitskalkül auf diesen Ansatz genauer eingegangen.

5.3 Ähnlichkeitskalkül

Wie bereits ausgeführt, ist eine wesentliche Anforderung einer Anfragesprache die Festlegung einer formalen Semantik für die Operationen der Sprache. Für den Entwurf einer Multimedia-Anfragesprache dient ein erweitertes, gewichtetes Bereichskalkül als Basis. Das *Ähnlichkeitskalkül* und die darauf aufsetzende *Ähnlichkeitsalgebra* wurden von Schmitt und Schulz (2004a,b) eingeführt. Sie stellen eine Erweiterung des Bereichskalküls beziehungsweise der Relationenalgebra dar. Alle Ausdrücke des Bereichskalküls können im Ähnlichkeitskalkül umgesetzt werden. Damit können neben unscharfen Anfragen weiterhin klassische scharfe Anfragen formuliert werden.

Die Definition des Ähnlichkeitskalküls erfolgt allgemeingültig, so dass es abgesehen von den Relationen und Junktoren keine Beschränkung auf bestimmte Operatoren beziehungsweise Prädikate gibt. Dadurch bildet das Ähnlichkeitskalkül einen Framework. Prinzipiell können damit zeitliche, räumliche, unscharfe sowie scharfe Anfragen leicht integriert werden.

Im Folgenden werden die Besonderheiten des Ähnlichkeitskalküls beschrieben. Es werden ferner einige Veränderungen diskutiert, die die Nutzerfreundlichkeit steigern.

5.3.1 Charakteristika

Im Ähnlichkeitskalkül von Schmitt und Schulz (2004a,b) werden verschiedene Junktoren angeboten. Neben den klassischen Junktoren, wie $\wedge$ und $\vee$, gibt es die gewichteten Junktoren:

$$\overrightarrow{\wedge}_\theta, \overleftarrow{\wedge}_\theta, \overrightarrow{\vee}_\theta \text{ sowie } \overleftarrow{\vee}_\theta$$

mit dem Relevanzgewicht $\theta \in [0,1]$. Hierbei wird zwischen linksorientierten und rechtsorientierten gewichteten Junktoren unterschieden. Die binären Junktoren ermöglichen es, einen Operanden voll zu gewichten, während der andere Operand ein Gewicht von *null* erhält. Dies führt dazu, dass der mit *null* gewichtete Operand bei der Berechnung des Relevanzwertes ignoriert wird. Die asymmetrisch gewichteten Junktoren zeigen an, für welchen Operanden das stärkere Gewicht gilt. Demnach wird bei den linksorientierten Junktoren

$$(E_1 \overleftarrow{\wedge}_\theta E_2) \text{ und } (E_1 \overleftarrow{\vee}_\theta E_2)$$

der Operand E_1 stärker gewichtet als der Operand E_2. Bei den rechtsorientierten Junktoren gilt dementsprechend, dass der Operand E_2 stärker gewichtet ist als der Operand E_1.

Die asymmetrischen Junktoren verlangen nach einer binären Gewichtung. Bei der Anfrageformulierung kommt eine binäre Gewichtung jedoch nicht dem Nutzerempfinden entgegen. Dies wird im Folgenden anhand einer Beispielanfrage demonstriert:

Beispiel 5.2 *Suche alle Bilder, die einen Hund, eine Katze und eine Maus enthalten. Dabei ist der Hund am wichtigsten, gefolgt von der Katze. Die geringste Wichtigkeit hat die Maus.*

Stehen nur binäre Junktoren zur Verfügung, liegt es an dem Nutzer, herauszufinden, welche Gewichte seine Anfrage am besten abbilden. Damit schränken binäre, gewichtete Junktoren den Nutzer bei der Anfrageformulierung zu stark ein.

Als Lösung bietet sich daher die Unterstützung von n-ären gewichteten Junktoren im Ähnlichkeitskalkül an. Damit kann die obige Beispielanfrage direkt in das Ähnlichkeitskalkül überführt werden. Eine Umrechnung der Nutzergewichtung auf einzelne binäre Gewichte ist nicht nötig. Darüber hinaus sind die Scoring-Funktionen, die einen Junktor umsetzen, bereits n-äre Funktionen.

Neben einer Relevanzgewichtung mittels gewichteter Junktoren soll ferner eine Parametrisierung des Verhaltens der Junktoren möglich sein. Dies ist wünschenswert, da verschiedene Studien gezeigt haben, dass die Junktoren $\wedge$ und $\vee$ das Nutzerempfinden nicht angemessen umsetzen können (Zimmermann und Zysno, 1980; Cooper, 1988; Avrahami und Kareev, 1993). Der Aspekt der Operatorgewichtung ist bisher nicht in dem Ähnlichkeitskalkül berücksichtigt worden. Aus diesem Grund wird, wie im Multi-Level-Gewichtungsmodell vorgeschlagen, der Universaljunktor $\oplus^z$ in das Ähnlichkeitskalkül integriert. Mittels des Parameters z wird das Verhalten des Junktors zwischen *UND* und *ODER* verändert. Der Universaljunktor basiert auf der Idee eines kompensatorischen *ODER*-Operators (siehe Abschnitt 2.4.2). Ein z-Wert von 1 bedeutet eine Konjunktion, während ein z-Wert von 0 zu einer Disjunktion führt. Zusätzlich dazu wird der Universaljunktor auch als gewichteter Junktor angeboten. Damit vereint dieser Operator eine Relevanz- und eine Operatorgewichtung. Mit Hilfe des Universaljunktors können sowohl scharfe als auch unscharfe Verknüpfungen ausgedrückt werden. Durch die Parametrisierung ist der Universaljunktor nicht assoziativ, so dass er als n-ärer Junktor realisiert wird.

Zusätzlich zu der Relevanzgewichtung der Anfrageterme, die durch Verwendung gewichteter Junktoren ermöglicht wird, wird eine Schwellwertgewichtung angeboten. Dabei werden drei Semantiken unterstützt (vgl. Abschnitt 2.4.2). Der *cut*-Operator $\succ\!\!\!\circ_\theta$ setzt eine *alles-oder-nichts*-Strategie um (siehe Abbildung 5.1a). Der $softcut$-Operator $\succ\!\!\!\sim_\theta$ verändert den Relevanzwert für ein Objekt nicht, sofern dessen RSV über dem Schwellwert θ liegt. Ist der Relevanzwert kleiner als der spezifizierte Schwellwert, so wird dieser auf *null* gesetzt. Das Verhalten dieses Operators ist in Abbildung 5.1b dargestellt. Neben diesen beiden bereits von Schmitt und Schulz (2004a) vorgestellten Operatoren wird ein nutzerdefinierter *usercut*-Operator $\succ\!\!\!\circ_{\theta,f}$ angeboten. Die Berechnung des Relevanzwertes erfolgt unter Verwendung der Funktion $f : [0,1] \times [0,1] \rightarrow [0,1]$, die das Schwellwertgewicht θ sowie den Relevanzwert auf einen Wert aus dem Interval $[0,1]$ abbildet. Der Einsatz dieses Operators ist dann sinnvoll, wenn das gewünschte Verhalten nicht durch die Operatoren $\succ\!\!\!\circ_\theta$ und $\succ\!\!\!\sim_\theta$ ausgedrückt werden kann. Als Funktion f kann beispielsweise die von Buell und Kraft (1981) vorgeschlagene Schwellwertfunktion genutzt werden (vgl. Abschnitt 2.4.2).

Damit wird im Ähnlichkeitskalkül eine Relevanz-, Schwellwert- und Operatorgewichtung unterstützt. Diese verschiedenen Möglichkeiten zur Abbildung der Nutzerpräferenzen im Ähnlichkeitskalkül erlauben eine adäquate Umsetzung des Multi-Level-Gewichtungsmodells.

Eine weitere Besonderheit des Ähnlichkeitskalküls stellen die parametrisierbaren Quantoren dar. Im klassischen Fall hängen die Quantoren von einem einzigen Wert ab. Dieses Verhalten ist in einigen Fällen jedoch zu restriktiv, so dass parametrisierbare Quantoren eingeführt werden. Der unscharfe Existenzquantor $\exists_k X$ kann auch als *einige* und der unscharfe Allquantor $\forall_k$ als *die meisten* verstanden werden (Schmitt und Schulz, 2004b). Im $\mathcal{SDC}$ benötigen diese unscharfen Quantoren mindestens k signifikante X-Werte, um das gleiche Verhalten wie die normalen Quantoren aufzuweisen. Stehen keine k Werte zur Verfügung, so wird der Relevanzwert für ein Objekt anteilig reduziert. Bei $k = 1$ entsprechen die un-

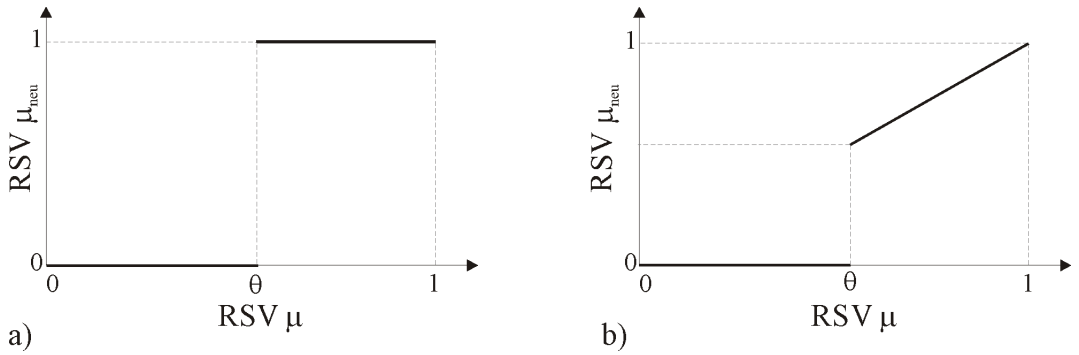


Abbildung 5.1: Das Verhalten des *cut*-Operators $\succ\!\!\!\epsilon_\theta$ (a) und des *softcut*-Operators $\succ\!\!\!\sim_\theta$ (b)

scharfen Quantoren den bekannten klassischen Quantoren.

In dem von Schmitt und Schulz (2004a,b) vorgestellten Ähnlichkeitskalkül werden die ermittelten Relevanzwerte nicht abgespeichert. Ferner ist es nicht möglich, unscharfe Relationen für die Anfrage zu verwenden. Aus diesem Grund werden in dieser Arbeit sogenannte Fuzzy-Relationen eingeführt. Diese verfügen an erster Stelle über ein Attribut für den Relevanzwert. Weist ein Tupel einen Relevanzwert von 1 auf, dann bedeutet dies, dass das Tupel ein volles Mitglied der Relation ist. Demgegenüber können Tupel auch nur zu einem bestimmten Grad zwischen 0 und 1 Mitglied der Relation sein. Diese Herangehensweise ist im Bereich der Fuzzy-Datenbanken weit verbreitet (Yazici und George, 1999, S. 144). Durch Verwendung von Fuzzy-Relationen ist das Ergebnis einer Anfrage im Ähnlichkeitskalkül wieder im relationalen Datenbankmodell darstellbar. Damit kann eine unscharfe Ergebnisrelation als Eingangsrelation für weitere Anfragen dienen, so dass der Forderung nach Abgeschlossenheit Rechnung getragen wird.

Im Folgenden wird näher auf die hier skizzierten Operatoren und Junktoren des Ähnlichkeitskalküls eingegangen. Zunächst wird die Syntax formal beschrieben. Daran anschließend erfolgt die Beschreibung der Semantik des Ähnlichkeitskalküls.

5.3.2 Syntax

In den folgenden Definitionen wird die Syntax für das Ähnlichkeitskalkül beschrieben.

Definition 5.3.1. Ein *Ähnlichkeitskalkül* $\mathcal{SDC}$ ist ein Tupel $(\mathbf{U}, A_0, \mathbf{X}, \Delta, \mathbf{C}, \mathbf{D}, Dom, \mathbf{R}, \mathbf{R}^+, \Theta, \mathbf{Z}, \mathbf{F})$, wobei $\mathbf{U} = \{A_1, A_2, \ldots\}$ das Universum der Attribute ist; A_0 das Attribut für den Relevanzwert ist; $\mathbf{X} = \{X_1, X_2, \ldots\}$ eine Menge von Variablen ist; $\Delta = \{\delta_1, \delta_2, \ldots\}$ eine Menge binärer Operationsnamen ist, wobei beide Operanden vom gleichen Typ sind; $\mathbf{C}$ eine Menge von Konstantennamen ist; $\mathbf{D}$ eine Menge von Bereichsnamen ist; *Dom* eine Abbildung von $\mathbf{U} \cup A_0 \cup \mathbf{X} \cup \Delta \cup \mathbf{C}$ nach $\mathbf{D}$ ist; $\mathbf{R}$ eine endliche Menge von Relationenschemata $R_1, R_2, \ldots, R_p$ ist und jedes Relationenschema eine Teilmenge von $\mathbf{U}$ ist; $\mathbf{R}^+$ eine endliche Menge von Fuzzy-Relationenschemata $R_1^+, R_2^+, \ldots, R_p^+$ ist und jedes Fuzzy-Relationenschema sich aus A_0 und einer Teilmenge von $\mathbf{U}$ ergibt; $\Theta = \{\theta_1, \theta_2, \ldots\}$ eine Menge von Gewichtsvariablen ist, $\mathbf{Z} = \{z_1, z_2, \ldots\}$ eine Menge von Operatorgewichtsvariablen und $\mathbf{F} = \{f_1, f_2, \ldots\}$ eine Menge binärer Funktionsnamen ist.

□

Anzumerken sei, dass bei den Operationen zwischen diskreten und kontinuierlichen Operationen unterschieden wird. Erstere umfassen beispielsweise die traditionellen Vergleichsoperationen $<, \leq, =, \neq, >, \geq$, während zu den letzteren beispielsweise der Ähnlichkeitsvergleich $\sim$ zählt.

Ein Anfragekalkülausdruck E über $\mathcal{SDC}$ ergibt sich analog zum Bereichskalkül aus *Termen*, *Atomen* und *Formeln*. Dieser Zusammenhang ist in Abbildung 5.2 dargestellt.

Abbildung 5.2: Formelaufbau im Ähnlichkeitskalkül

Definition 5.3.2. Ein *Term* Y_i wird aus Konstanten und Variablen gebildet. Damit ist:

1. jede Variable $X_i \in X$ mit dem Wertebereich $Dom(X_i)$ und

2. jede Konstante $C_i \in C$ mit dem Wertebereich $Dom(C_i)$

ein Term Y_i. □

Terme bilden die Argumente für Prädikate. Damit ist ein *Atom* eine atomare Formel über Terme. Es ist wie folgt definiert:

Definition 5.3.3. Ein *Atom* sei

1. $R(Y_1, Y_2, \ldots, Y_m)$, wobei $R \in \mathbf{R}$ ein Relationenschema $A_1, A_2, \ldots, A_m$ ist und für jedes Y_i der Wertebereich korrekt ist. Es gilt $Dom(Y_i) = Dom(A_i)$.

2. $R^+(Y_0, Y_1, Y_2, \ldots, Y_m)$, wobei $R^+ \in \mathbf{R}^+$ ein Fuzzy-Relationenschema $A_0, A_1, A_2, \ldots, A_m$ ist und für jedes Y_i der Wertebereich korrekt ist. Das Attribut A_0 verwaltet den Relevanzwert und befindet sich an erster Stelle der Relation. Es gilt $Dom(Y_i) = Dom(A_i)$.

3. $Y_1 \delta Y_2$, wobei $\delta \in \Delta$ ein Operationsname ist und Y_1, Y_2 vom gleichen Typ sind. Es gilt $Dom(Y_1) = Dom(Y_2) = Dom(\delta)$.

□

Anzumerken ist, dass bei den Operationen beide Operanden sowohl Variablen als auch Konstanten sein können. Eine Operation auf Konstanten kann beispielsweise zur Manipulation der Relevanzwerte genutzt werden.

Auf Basis der Atome kann eine *Formel* aufgebaut werden. Hierfür bietet das Ähnlichkeitskalkül eine Vielzahl von Junktoren für die Verknüpfung der Atome an. Neben den allgemein üblichen Junktoren $\wedge$ und $\vee$ steht jeweils eine gewichtete Version zur Verfügung. Des Weiteren wird ein Universaljunktor $\oplus^z$ angeboten. Der Universaljunktor wird als Sonderfall der Disjunktion verstanden. Bei einem z-Wert von *null* gilt $\oplus^z = \vee$. Auch für diesen Junktor wird eine gewichtete Variante angeboten.

Neben den Junktoren werden auch die Schwellwertoperatoren sowie die Quantoren beim Aufbau einer Formel berücksichtigt. In der folgenden Definition wird der Formelaufbau beschrieben. Die formale Semantik der einzelnen Operatoren wird im nachfolgenden Abschnitt in Definition 5.3.12 gegeben.

Definition 5.3.4. Eine *$\mathcal{SDC}$-Formel* $F(X_1, X_2, \ldots, X_n)$, kurz F, mit den involvierten freien Variablen $X_i \in \mathbf{X}$ und $i = 1, \ldots, n$ ist wie folgt rekursiv definiert:

1. Jedes Atom ist eine Formel $F(X_1, X_2, \ldots, X_n)$ mit den freien Variablen X_i.

2. $(F_1(X_{1_1}, \ldots, X_{1_k}) \phi F_2(X_{2_1}, \ldots, X_{2_l}))$ mit $\phi \in \{\wedge, \vee\}$ ist eine Formel $F(X_1, \ldots, X_n)$, wenn $F_1(X_{1_1}, \ldots, X_{1_k})$ und $F_2(X_{2_1}, \ldots, X_{2_l})$ Formeln sind. Die involvierten Variablen werden vereinigt: $\{X_1, \ldots, X_n\} = \{X_{1_1}, \ldots, X_{1_k}\} \cup \{X_{2_1}, \ldots, X_{2_l}\}$.

3. $(\oplus^z(F_1(X_{1_1}, \ldots, X_{1_k}), F_2(X_{2_1}, \ldots, X_{2_l}), \ldots, F_m(X_{m_1}, \ldots, X_{m_r})))$ mit dem Operatorgewicht $z \in \mathbf{Z}$ ist eine Formel $F(X_1, \ldots, X_n)$ genau dann, wenn alle Operanden F_j mit $j = 1, \ldots, m$ Formeln sind. Die involvierten Variablen werden vereinigt: $\{X_1, \ldots, X_n\} = \{X_{1_1}, \ldots, X_{1_k}\} \cup \{X_{2_1}, \ldots, X_{2_l}\} \cup \ldots \cup \{X_{m_1}, \ldots, X_{m_r}\}$.

   Bei nur zwei Operanden wird die Infix-Notation verwendet.

4. $(\phi(F_1(X_{1_1}, \ldots, X_{1_k}), F_2(X_{2_1}, \ldots, X_{2_l}), \ldots, F_m(X_{m_1}, \ldots, X_{m_r})))$ ist eine Formel $F(X_1, \ldots, X_n)$ genau dann, wenn alle Operanden F_j mit $j = 1, \ldots, m$ Formeln sind und für jede Teilformel F_j ein Gewicht $\theta_j \in \Theta$ existiert und $\phi \in \{\wedge_{(\theta_1, \theta_2, \ldots, \theta_m)}, \vee_{(\theta_1, \theta_2, \ldots, \theta_m)}, \oplus^z_{(\theta_1, \theta_2, \ldots, \theta_m)}\}$.

   Die Reihenfolge der Formeln ist abhängig von der Gewichtung. Es gilt $\theta_1 \geq \theta_2 \geq \ldots \geq \theta_m$. Die involvierten Variablen werden vereinigt: $\{X_1, \ldots, X_n\} = \{X_{1_1}, \ldots, X_{1_k}\} \cup \{X_{2_1}, \ldots, X_{2_l}\} \cup \ldots \cup \{X_{m_1}, \ldots, X_{m_r}\}$.

   Bei zwei Operanden wird anstelle der Präfix-Notation die Infix-Notation verwendet.

5. $(\neg F(X_1, \ldots, X_n))$ ist eine Formel, wenn $F(X_1, \ldots, X_n)$ eine Formel ist.

6. $(\exists X F(X_1, \ldots, X_n))$ ist eine Formel, wenn $F(X_1, \ldots, X_n)$ eine Formel ist und es ein i gibt, so dass gilt $X_i = X$.

7. $(\forall X F(X_1, \ldots, X_n))$ ist eine Formel, wenn $F(X_1, \ldots, X_n)$ eine Formel ist und es ein i gibt, so dass gilt $X_i = X$.

8. $(\exists_k X F(X_1, \ldots, X_n))$ ist eine Formel, wenn $F(X_1, \ldots, X_n)$ eine Formel ist, $k > 1$ eine natürliche Zahl ist und es ein i gibt, so dass gilt $X_i = X$.

9. $(\forall_k X F(X_1, \ldots, X_n))$ ist eine Formel, wenn $F(X_1, \ldots, X_n)$ eine Formel ist, $k > 1$ eine natürliche Zahl ist und es ein i gibt, so dass gilt $X_i = X$.

10. $(\succ\!\!\!\circ_\theta F(X_1, \ldots, X_n))$ ist eine Formel, wenn $F(X_1, \ldots, X_n)$ eine Formel ist und es ein Gewicht $\theta \in \Theta$ gibt.

11. $(\prec\!\!\!\circ_\theta F(X_1, \ldots, X_n))$ ist eine Formel, wenn $F(X_1, \ldots, X_n)$ eine Formel ist und es ein Gewicht $\theta \in \Theta$ gibt.

12. $(\succcurlyeq_{\theta,f} F(X_1,\ldots,X_n))$ ist eine Formel, wenn $F(X_1,\ldots,X_n)$ eine Formel ist und es ein Gewicht $\theta \in \Theta$ sowie eine Funktion $f \in \mathbf{F}$ gibt.

□

Nachdem der Aufbau einer Formel dargelegt wurde, erfolgt die Definition eines *Anfrageausdrucks* im Ähnlichkeitskalkül. Dieser ergibt sich, wie im Bereichskalkül, aus freien Variablen X_i mit $i = 1,\ldots,n$ und einer Formel F über diesen Variablen. Ein Anfrageausdruck ist demnach wie folgt definiert:

Definition 5.3.5. Ein *Anfrageausdruck* E über $\mathcal{SDC}$ hat die Form

$$\{X_1, X_2, \ldots, X_n | F(X_1, X_2, \ldots, X_n)\}$$

mit den freien Variablen $X_1,\ldots,X_n$. □

Eine Anfrage ermittelt somit zunächst die Werte für alle Variablen, die die Bedingung F erfüllen. Eine Erweiterung der Anfrage, so dass Funktionen auf den Ergebnisvariablen angewendet werden können, ist möglich, wird jedoch hier nicht weiter betrachtet.

Um ein endliches und bereichsunabhängiges Anfrageergebnis erzielen zu können, müssen die Anfragen sicher sein. Dies ist eine wichtige Anforderung an eine Anfragesprache. Zur Überprüfung der syntaktischen Sicherheit wird auf dem Ansatz von Gelder und Topor (1991) aufgebaut. Dabei ist die Evaluierbarkeit einer Anfrage das entscheidende Kriterium.

Nach Gelder und Topor werden für die Überprüfung der Evaluierbarkeit einer Anfrage die Prädikate $gen()$ und $con()$ eingesetzt. Somit wird überprüft, ob jeder Wert der Variablen X, der bei der Evaluierung einer Formel F entsteht, ein Wert aus der Datenbank ist.

Die Relation $gen()$ wird zum Testen der Sicherheit von freien Variablen verwendet, während $con()$ für den Test der gebundenen Variablen genutzt wird. Um die Sicherheit zu verifizieren, wird überprüft, an welche Relationen die jeweiligen Variablen gebunden sind. Die Unterscheidung zwischen $gen()$ und $con()$ ist notwendig, da die Werte für die freien Variablen im Gegensatz zu den gebundenen Variablen als Anfrageergebnis ausgegeben werden. Damit muss bei der Evaluierung der Quantoren mittels $con()$ nur überprüft werden, ob endliche Mengen zu Grunde liegen. Dies wird erfüllt, sofern jeder mögliche Wert, der die Teilformel erfüllt, ein Wert aus der Datenbank ist, beziehungsweise dann, wenn die Teilformel die quantifizierte Variable nicht enthält. Im Ähnlichkeitskalkül werden diese Prädikate wie folgt definiert:

Definition 5.3.6. Das Prädikat $gen(X, F, E)$, mit der Variablen $X \in \mathbf{X}$, der $\mathcal{SDC}$-Formel F und einem Algebraausdruck E ist induktiv definiert:

$$\begin{aligned}
gen(X, F, \pi_{\#_i}(R)) &\Leftrightarrow F = R(Y_1,\ldots,Y_{i-1},X,Y_{i+1},\ldots,Y_m)\\
gen(X, F, \pi_{\#_i}(R^+)) &\Leftrightarrow F = R^+(Y_0,\ldots,Y_{i-1},X,Y_{i+1},\ldots,Y_m)\\
gen(X, (F_1 \wedge F_2), E) &\Leftrightarrow gen(X, F_1, E) \text{ oder } gen(X, F_2, E)\\
gen(X, (\wedge_{(\theta_1,\theta_2,\ldots,\theta_m)}(F_1, F_2, \ldots, F_m)), E) &\Leftrightarrow gen(X, F_1, E)\\
gen(X, (F_1 \vee F_2), (E_1) \cup (E_2)) &\Leftrightarrow gen(X, F_1, E_1) \text{ und } gen(X, F_2, E_2)\\
gen(X, (\phi(F_1, F_2, \ldots, F_m)), (E_1) \cup (E_2) \cup \ldots \cup (E_m)) &\Leftrightarrow \forall j\ gen(X, F_j, E_j)
\end{aligned}$$

$$
\begin{aligned}
gen(X,(\neg F),E) &\Leftrightarrow gen(X,pushnot(\neg F),E)\\
gen(X,(\exists X_1 F),E) &\Leftrightarrow gen(X,F,E) \text{ und } X \neq X_1\\
gen(X,(\forall X_1 F),E) &\Leftrightarrow gen(X,F,E) \text{ und } X \neq X_1\\
gen(X,(\exists_k X_1 F),E) &\Leftrightarrow gen(X,F,E) \text{ und } X \neq X_1\\
gen(X,(\forall_k X_1 F),E) &\Leftrightarrow gen(X,F,E) \text{ und } X \neq X_1\\
gen(X,(\succ\!\!\!\approx_\theta F),E) &\Leftrightarrow gen(X,F,E)\\
gen(X,(\not\succ\!\!\!\approx_\theta F),E) &\Leftrightarrow gen(X,F,E)\\
gen(X,(\succ\!\!\!\approx_{\theta,f} F),E) &\Leftrightarrow gen(X,F,E).
\end{aligned}
$$

Es gilt $j = 1,\ldots,m$ und $\phi \in \{\vee_{(\theta_1,\theta_2,\ldots,\theta_m)}, \oplus^z, \oplus^z_{(\theta_1,\theta_2,\ldots,\theta_m)}\}$. Durch die *pushnot*-Operation wird die Negation in einer Formel nach Innen, also zu den Atomen, gezogen. Dafür gelten die folgenden Ersetzungsregeln:

$$
\begin{aligned}
(\neg(F_1 \wedge F_2)) &\Leftrightarrow ((\neg F_1) \vee (\neg F_2))\\
(\neg(\wedge_{(\theta_1,\theta_2,\ldots,\theta_m)}(F_1,F_2,\ldots,F_m))) &\Leftrightarrow (\vee_{(\theta_1,\theta_2,\ldots,\theta_m)}((\neg F_1),(\neg F_2),\ldots,(\neg F_m)))\\
(\neg(F_1 \vee F_2)) &\Leftrightarrow ((\neg F_1) \wedge (\neg F_2))\\
(\neg(\vee_{(\theta_1,\theta_2,\ldots,\theta_m)}(F_1,F_2,\ldots,F_m))) &\Leftrightarrow (\wedge_{(\theta_1,\theta_2,\ldots,\theta_m)}((\neg F_1),(\neg F_2),\ldots,(\neg F_m)))\\
(\neg(\oplus^z(F_1,F_2,\ldots,F_m))) &\Leftrightarrow (\oplus^{1-z}((\neg F_1),(\neg F_2),\ldots,(\neg F_m)))\\
(\neg(\oplus^z_{(\theta_1,\theta_2,\ldots,\theta_m)}(F_1,F_2,\ldots,F_m))) &\Leftrightarrow (\oplus^{1-z}_{(\theta_1,\theta_2,\ldots,\theta_m)}((\neg F_1),(\neg F_2),\ldots,(\neg F_m)))\\
(\neg(\neg F)) &\Leftrightarrow F\\
(\neg(\exists X F)) &\Leftrightarrow (\forall X(\neg F))\\
(\neg(\forall X F)) &\Leftrightarrow (\exists X(\neg F))\\
(\neg(\exists_k X F)) &\Leftrightarrow (\forall_k X(\neg F))\\
(\neg(\forall_k X F)) &\Leftrightarrow (\exists_k X(\neg F))\\
(\neg\succ\!\!\!\approx_\theta F) &\Leftrightarrow (\succ\!\!\!\approx_{1-\theta}\neg F)\\
(\neg\not\succ\!\!\!\approx_\theta F) &\Leftrightarrow (\neg((\succ\!\!\!\approx_\theta F) \wedge F))\\
(\neg\succ\!\!\!\approx_{\theta,f} F) &\Leftrightarrow (\succ\!\!\!\approx_{\theta,\neg f} F)
\end{aligned}
$$

□

Wird also mittels der Relation $gen()$ einer Variablen X und einer Formel F ein Algebraausdruck E zugeordnet, dann beinhaltet das Ergebnis der Auswertung von E alle Werte der Variablen X, die die Formel F erfüllen.

Analog dazu wird das Prädikat $con()$ zur Evaluierung gebundener Variablen wie folgt definiert:

Definition 5.3.7. Das Prädikat $con(X,F,E)$ mit der Variablen $X \in \mathbf{X}$, der $\mathcal{SDC}$-Formel F und dem Algebraausdruck E wird wie folgt induktiv definiert:

$$
\begin{aligned}
con(X,F,\pi_{\#_i}(R)) &\Leftrightarrow F = R(Y_1,\ldots,Y_{i-1},X,Y_{i+1},\ldots,Y_m)\\
con(X,F,\pi_{\#_i}(R^+)) &\Leftrightarrow F = R^+(Y_0,\ldots,Y_{i-1},X,Y_{i+1},\ldots,Y_m)\\
con(X,F(X_1,\ldots,X_m),\perp) &\Leftrightarrow X_i \neq X \text{ für } i = 1,\ldots,m
\end{aligned}
$$

$$
\begin{aligned}
con(X,(F_1 \wedge F_2),E) &\Leftrightarrow gen(X,F_1,E) \text{ oder } gen(X,F_2,E)\\
con(X,(\wedge_{(\theta_1,\theta_2,\ldots,\theta_m)}(F_1,F_2,\ldots,F_m)),E) &\Leftrightarrow gen(X,F_1,E)\\
con(X,(F_1 \wedge F_2),(E_1)\cup(E_2)) &\Leftrightarrow con(X,F_1,E_1) \text{ und } con(X,F_2,E_2)\\
con(X,(F_1 \vee F_2),(E_1)\cup(E_2)) &\Leftrightarrow con(X,F_1,E_1) \text{ und } con(X,F_2,E_2)\\
con(X,(\phi(F_1,F_2,\ldots F_m),(E_1)\cup(E_2)\cup\ldots\cup(E_m)) &\Leftrightarrow \forall j\ con(X,F_j,E_j)\\
con(X,(\neg F),E) &\Leftrightarrow con(X,pushnot(\neg F),E)\\
con(X,(\exists X_1 F),E) &\Leftrightarrow con(X,F,E) \text{ und } X \neq X_1\\
con(X,(\forall X_1 F),E) &\Leftrightarrow con(X,F,E) \text{ und } X \neq X_1\\
con(X,(\exists_k X_1 F),E) &\Leftrightarrow con(X,F,E) \text{ und } X \neq X_1\\
con(X,(\forall_k X_1 F),E) &\Leftrightarrow con(X,F,E) \text{ und } X \neq X_1\\
con(X,(\prec\!\!\!\!\cdot_\theta F),E) &\Leftrightarrow con(X,F,E)\\
con(X,(\succ\!\!\!\!\cdot_\theta F),E) &\Leftrightarrow con(X,F,E)\\
con(X,(\prec\!\!\!\!\cdot_{\theta,f} F),E) &\Leftrightarrow con(X,F,E)
\end{aligned}
$$

Es gilt $j = 1,\ldots,m$ und $\phi \in \{\oplus^z, \wedge_{(\theta_1,\theta_2,\ldots,\theta_m)}, \vee_{(\theta_1,\theta_2,\ldots,\theta_m)}, \oplus^z_{(\theta_1,\theta_2,\ldots,\theta_m)}\}$. Das Symbol '$\perp$' steht für eine 0-äre Relation, die keine Tupel enthält. Somit entspricht '$\perp$' dem neutralen Element bezüglich der Vereinigung. □

Unter Verwendung dieser Prädikate kann nun die Evaluierbarkeit eines $\mathcal{SDC}$-Anfrageausdrucks im Sinne von Gelder und Topor (1991) definiert werden.

Definition 5.3.8. Eine $\mathcal{SDC}$-Anfrage $\{X_1, X_2, \ldots, X_n | F(X_1, X_2, \ldots, X_n)\}$ ist *evaluierbar*, wenn

1. für jede Variable X_i mit $i = 1,\ldots,n$ das Prädikat $gen(X_i, F, E)$ gilt,
2. für jede Variable X, die in einer Teilformel $(\exists X F_1)$ beziehungsweise $(\exists_k X F_1)$von F gebunden ist, das Prädikat $con(X, F_1, E)$ gilt und darüber hinaus bei $(\exists_k X F_1)$ der Ausdruck E nicht das Symbol '$\perp$' enthält,
3. für jede Variable X, die in einer Teilformel $(\forall X F_1)$ beziehungsweise $(\forall_k X F_1)$ von F gebunden ist, das Prädikat $con(X, \neg F_1, E)$ gilt und darüber hinaus bei $(\forall_k X F_1)$ der Ausdruck E nicht das Symbol '$\perp$' enthält, und
4. $F(X_1, X_2, \ldots, X_n)$ rektifiziert ist, das heißt, jede Variable ist entweder frei oder einmalig gebunden.

□

Auf Basis der beschriebenen Syntax kann ein Anfrageausdruck E im Ähnlichkeitskalkül konstruiert werden. Dieser stellt eine Verallgemeinerung eines Anfrageausdrucks des Bereichskalküls dar. Neben beliebigen traditionellen Anfragen können daher auch unscharfe Anfragen formuliert werden.

Im folgenden Abschnitt wird die konkrete Semantik des Kalküls vorgestellt. Es ist zu beachten, dass die Semantik der Operatoren nicht vordefiniert ist. Vielmehr werden Anforderungen an die Operatoren gestellt, so dass das $\mathcal{SDC}$ beliebig erweiterbar ist. Damit kann das

Ähnlichkeitskalkül auf spezielle Anwendungen zugeschnitten werden. Die Gewichte werden durch Gewichtsvariablen spezifiziert, wobei die genauen Werte durch eine Interpretationsfunktion außerhalb der Anfrage festgelegt werden.

5.3.3 Semantik

Nachdem die Syntax des Ähnlichkeitskalküls beschrieben wurde, wird die Semantik eines $\mathcal{SDC}$-Ausdrucks festgelegt. Es wird zunächst in Definition 5.3.9 die Interpretation des $\mathcal{SDC}$ gegeben. Es folgen Definitionen der Variablenzuordnung und der Atomersetzung. Abschließend wird die Semantik einer $\mathcal{SDC}$-Formel sowie die Semantik eines $\mathcal{SDC}$-Anfrageausdrucks definiert.

Definition 5.3.9. Eine *Interpretation* über $\mathcal{SDC}(\mathbf{U}, A_0, \mathbf{X}, \Delta, \mathbf{C}, \mathbf{D}, Dom, \mathbf{R}, \mathbf{R}^+, \Theta, \mathbf{Z}, \mathbf{F})$ ist ein Triple $(\mathbf{d}, \mathbf{db}, I)$ und

1. $\mathbf{d}$ ist eine endliche Menge von Wertebereichen $\{d_1, d_2, \ldots, d_q\}$, wobei jeder Bereich einer Menge von Werten entspricht.
2. $\mathbf{db}$ ist eine endliche Menge von Relationen $\{r_1, r_2, \ldots, r_p\}$.
3. I ist eine Interpretationsfunktion, die
   a) jedem Bereichsnamen $D \in \mathbf{D}$ einen Wertebereich zuordnet:
   $$I(D) \in \mathbf{d}.$$
   b) jedem Relationenschema $R(A_1, A_2, \ldots, A_n) \in \mathbf{R}$ eine Relation zuordnet:
   $$I(R) \in \mathbf{db}$$
   mit $I(R) \subseteq I(Dom(A_1)) \times I(Dom(A_2)) \times \ldots \times I(Dom(A_n))$.
   c) jedem Fuzzy-Relationsschema $R^+(A_0, A_1, A_2, \ldots, A_n) \in \mathbf{R}^+$eine Relation zuordnet:
   $$I(R^+) \in \mathbf{db}$$
   mit $I(R^+) \subseteq I(Dom(A_0)) \times I(Dom(A_1)) \times I(Dom(A_2)) \times \ldots \times I(Dom(A_n))$.
   d) jedem Operationsnamen $\delta \in \Delta$ eine binäre Funktion zuordnet:
   $$I(\delta) : I(Dom(\delta)) \times I(Dom(\delta)) \rightarrow [0,1].$$
   Die Werte einer kontinuierlichen Operation liegen im Interval $[0,1]$. Hingegen sind die Werte einer diskreten Operation auf $\{0,1\}$ beschränkt, wobei der Wert 0 gleichbedeutend mit dem Booleschen Wert `falsch` und 1 gleichbedeutend mit `wahr` ist.
   Die Semantik der Operationen ist nicht vordefiniert. So können neben den klassischen Vergleichsoperatoren, wie beispielsweise $<, =, \leq$ oder $\geq$, auch Ähnlichkeitsoperationen, wie $\sim$, sowie zeitliche beziehungsweise räumliche Operationen,

wie zum Beispiel *vor*, *nach* oder *links_von*, *in*, *nah*, in die Anfragesprache integriert werden. Auf diese Art und Weise ist die Anfragesprache universell für verschiedene Medien einsetz- respektive erweiterbar und kann entsprechend den anwendungsspezifischen Gegebenheiten angepasst werden. Für alle Datentypen gibt es den Vergleichsoperator $=$. Er weist eine Gleichheitssemantik auf.

e) jedem Konstantennamen $C \in \mathbf{C}$ einen Wert zuordnet:

$$I(C) \in I(Dom(C)).$$

f) jeder Gewichtsvariablen $\theta \in \Theta$ einen Wert zuordnet:

$$I(\theta) \in [0,1].$$

g) jeder Operatorgewichtsvariablen $z \in \mathbf{Z}$ einen Wert zuordnet:

$$I(z) \in [0,1).$$

h) jedem Funktionsnamen $f \in \mathbf{F}$ eine binäre Funktion zuordnet:

$$I(f) : [0,1] \times [0,1] \rightarrow [0,1].$$

i) jedem Konjunktionssymbol '$\wedge$'eine Scoring-Funktion, die die in Abschnitt 4.1.2 spezifizierten Anforderungen erfüllt, zuordnet:

$$I(\wedge) : [0,1] \times [0,1] \rightarrow [0,1].$$

Da '$\wedge$' sowohl assoziativ als auch kommutativ ist, kann der Junktor zu einem n-ären Junktor verallgemeinert werden.

j) jedem gewichteten Konjunktionssymbol '$\wedge_{(\theta_1,\theta_2,\ldots,\theta_m)}$' eine gewichtete Scoring-Funktion zuordnet:

$$I(\wedge_{(\theta_1,\theta_2,\ldots,\theta_m)}) : [0,1]^m \times [0,1]^m \rightarrow [0,1].$$

Der erste Parameter der Funktion ist für die Gewichte $I(\theta_j)$ mit $\sum_{j=1}^{m} I(\theta_j) = 1$ reserviert. Es müssen alle Anforderungen an eine gewichtete Scoring-Funktion erfüllt werden (vgl. Abschnitt 4.3.2).

k) jedem Disjunktionssymbol '$\vee$' eine Scoring-Funktion, die die in Abschnitt 4.1.2 spezifizierten Anforderungen erfüllt, zuordnet:

$$I(\vee) : [0,1] \times [0,1] \rightarrow [0,1].$$

Da '$\vee$' sowohl assoziativ als auch kommutativ ist, kann der Junktor zu einem n-ären Junktor verallgemeinert werden.

l) jedem gewichteten Disjunktionssymbol '$\vee_{(\theta_1,\theta_2,\ldots,\theta_m)}$' eine gewichtete Scoring-Funktion zuordnet:

$$I(\vee_{(\theta_1,\theta_2,\ldots,\theta_m)}) : [0,1]^m \times [0,1]^m \rightarrow [0,1].$$

Der erste Parameter der Funktion ist für die Gewichte $I(\theta_j)$ mit $\sum_{j=1}^{m} I(\theta_j) = 1$ reserviert. Es müssen alle Anforderungen an eine gewichtete Scoring-Funktion

erfüllt werden (vgl. Abschnitt 4.3.2).

m) jedem Kombinationssymbol '$\oplus^z$' einen kompensatorischen *ODER*-Operator zuordnet:

$$I(\oplus^z) : [0,1) \times [0,1]^m \rightarrow [0,1].$$

Der erste Parameter des Operators ist für das Operatorgewicht $I(z)$ reserviert. Bei einem z-Wert von 0 entspricht das Kombinationssymbol '$\oplus^z$' der klassischen Disjunktion. Somit gilt:

$$\forall \mu_1, \mu_2, \ldots, \mu_m \in [0,1] : I(\oplus^z)(0, \mu_1, \mu_2, \ldots, \mu_m) = I(\vee)(\mu_1, \mu_2, \ldots, \mu_m).$$

Für die Disjunktion wurde hier die als n-ärer Junktor verallgemeinerte Form genutzt.

n) jedem gewichteten Kombinationssymbol '$\oplus^z_{(\theta_1,\theta_2,\ldots,\theta_m)}$' einen gewichteten kompensatorischen *ODER*-Operator zuordnet:

$$I(\oplus^z_{(\theta_1,\theta_2,\ldots,\theta_m)}) : [0,1) \times [0,1]^m \times [0,1]^m \rightarrow [0,1].$$

Der erste Parameter des Operators ist für das Operatorgewicht $I(z)$ reserviert. Bei einem z-Wert von 0 entspricht das Kombinationssymbol '$\oplus^z$' der gewichteten Disjunktion. Bei einer Gleichgewichtung der Operanden ergibt sich eine ungewichtete Kombination, während bei einem Gewicht von 1 nur der am stärksten gewichtete Operand von Bedeutung ist. Daher gelten die folgenden Bedingungen:
$\forall v_1, \ldots v_m, \mu_1, \ldots, \mu_m \in [0,1]$:
$I(\oplus^z_{(\theta_1,\theta_2,\ldots,\theta_m)})(0, v_1, \ldots v_m, \mu_1, \ldots, \mu_m) = I(\vee_{(\theta_1,\theta_2,\ldots,\theta_m)})\ (v_1, \ldots v_m, \mu_1, \ldots, \mu_m)$

sowie
$\forall v \in [0,1), \mu_1, \ldots, \mu_m \in [0,1]$:
$I(\oplus^z_{(\theta_1,\theta_2,\ldots,\theta_m)})(v, 1/m, \ldots 1/m, \mu_1, \ldots, \mu_m) = I(\oplus^z)(v, \mu_1, \ldots, \mu_m)$

und
$\forall v_2, \ldots, v_m \in [0,1), \mu_1, \ldots, \mu_m \in [0,1]$:
$I(\oplus^z_{(\theta_1,\theta_2,\ldots,\theta_m)})(v, 1, v_2, \ldots v_m, \mu_1, \ldots, \mu_m) = \mu_1.$

o) jedem Negationssymbol '$\neg$' eine Negation zuordnet:

$$I(\neg) : [0,1] \rightarrow [0,1]$$

mit $\neg\neg\mu = \mu$. Des Weiteren muss die Negation neben der Involution im Zusammenhang mit der Konjunktion und der Disjunktion die Erfüllung der De Morganschen Gesetze gewährleisten.

□

Zusammenfassend wird somit jedem *UND*-Junktor eine Scoring-Funktion für Konjunktionen und jedem *ODER*-Junktor eine Scoring-Funktion für Disjunktionen zugeordnet. Dafür

können beliebige aus der Fuzzy-Logik bekannte T-Normen und T-Conormen eingesetzt werden. Der Einsatz andere Funktionen ist ebenso möglich. Voraussetzung ist, dass die Funktionen die Anforderungen an eine Scoring-Funktion erfüllen. Für die Umsetzung der gewichteten Junktoren kann im konkreten Fall eine der in Abschnitt 4.2 vorgestellten Scoring-Funktionen eingesetzt werden.

Der Universaloperator wird durch einen kompensatorischen *ODER*-Operator umgesetzt (siehe Abschnitt 2.4.2). Auch hierfür können je nach Bedarf unterschiedliche kompensatorische *ODER*-Operatoren verwendet werden. Der gewichtete kompensatorische Operator kann, realisiert werden, wie im Multi-Level-Gewichtungsmodell vorgeschlagen (vgl. Abschnitt 4.3.4).

Nachdem die Interpretationen für die einzelnen Symbole bekannt sind, wird nun auf die Variablenzuordnung und die Atomersetzung eingegangen.

Definition 5.3.10. Sei V eine *Variablenzuordnung* von $\mathbf{X}$ nach $\bigcup_{d \in \mathbf{d}} d$, wobei gilt $\forall X \in \mathbf{X}.V(X) \in I(Dom(X))$. □

Definition 5.3.11. Eine *Atomersetzung* $V^*(F)$ eines $\mathcal{SDC}$-Atoms F bezüglich der Variablenzuordnung V und einer Interpretationsfunktion I ist wie folgt definiert:

1. Falls $F = R(Y_1, Y_2, \ldots, Y_m)$ gegeben ist, dann wird das Atom durch '1' ersetzt, wenn

   $$i \in \{1, \ldots, m\}.v_i = \begin{cases} V(Y_i) & \text{wenn } Y_i \text{ ein Variable ist} \\ I(Y_i) & \text{wenn } Y_i \text{ ein Konstantenname ist} \end{cases}$$

   und

   $$(v_1, v_2, \ldots, v_m) \in I(R).$$

   Somit gilt $V^*(F) = 1$.

   Ansonsten wird das Atom durch '0' ersetzt: $V^*(F) = 0$.

2. Falls $F = R^+(Y_0, Y_1, Y_2, \ldots, Y_m)$ gegeben ist, dann gilt $V^*(F) = v_0$, wenn

   $$i \in \{0, \ldots, m\}.v_i = \begin{cases} V(Y_i) & \text{wenn } Y_i \text{ ein Variable ist} \\ I(Y_i) & \text{wenn } Y_i \text{ ein Konstantenname ist} \end{cases}$$

   und

   $$(v_0, v_1, v_2, \ldots, v_m) \in I(R).$$

   Ansonsten wird das Atom durch '0' ersetzt: $V^*(F) = 0$.

3. Falls $F = Y_1 \delta Y_2$ gegeben ist, dann wird das Atom durch den Wert $I(\delta)(v_1, v_2)$ ersetzt. Somit gilt $V^*(F) = I(\delta)(v_1, v_2)$ und

   $$i \in \{1, 2\}.v_i = \begin{cases} V(Y_i) & \text{wenn } Y_i \text{ ein Variable ist} \\ I(Y_i) & \text{wenn } Y_i \text{ ein Konstantenname ist} \end{cases}$$

□

Definition 5.3.12. Die *Semantik* einer $\mathcal{SDC}$-Formel $F(X_1, X_2, \ldots, X_n)$ wird mit $I_V^*(F)$ notiert. Sie basiert auf einer Atomersetzung $V^*(F)$ und einer Interpretationsfunktion I. Die Semantik ist wie folgt rekursiv definiert:

1. Wenn F ein Atom ist, dann gilt:
$$I_V^*(F) = V^*(F).$$

2. Wenn F eine Konjunktion $(F_1 \wedge F_2)$ ist, dann gilt:
$$I_V^*(F) = I(\wedge)(I_V^*(F_1), I_V^*(F_2)).$$

3. Wenn F eine gewichtete Konjunktion $(\wedge_{(\theta_1,\theta_2,\ldots,\theta_m)}(F_1, F_2, \ldots F_m))$ ist, dann gilt:
$$I_V^*(F) = I(\wedge_{(\theta_1,\theta_2,\ldots,\theta_m)})(I(\theta_1), I(\theta_2), \ldots I(\theta_m), I_V^*(F_1), I_V^*(F_2), \ldots, I_V^*(F_m)).$$

4. Wenn F eine Disjunktion $(F_1 \vee F_2)$ ist, dann gilt:
$$I_V^*(F) = I(\vee)(I_V^*(F_1), I_V^*(F_2)).$$

5. Wenn F eine gewichtete Disjunktion $(\vee_{(\theta_1,\theta_2,\ldots,\theta_m)}(F_1, F_2, \ldots F_m))$ ist, dann gilt:
$$I_V^*(F) = I(\vee_{(\theta_1,\theta_2,\ldots,\theta_m)})(I(\theta_1), I(\theta_2), \ldots I(\theta_m), I_V^*(F_1), I_V^*(F_2), \ldots, I_V^*(F_m)).$$

6. Wenn F eine Kombination $(\oplus^z(F_1, F_2, \ldots F_m))$ ist, dann gilt:
$$I_V^*(F) = I(\oplus^z)(I(z), I_V^*(F_1), I_V^*(F_2), \ldots, I_V^*(F_m)).$$

7. Wenn F eine gewichtete Kombination $(\oplus^z_{(\theta_1,\theta_2,\ldots,\theta_m)}(F_1, F_2, \ldots F_m))$ ist, dann gilt:
$$I_V^*(F) = I(\oplus^z_{(\theta_1,\theta_2,\ldots,\theta_m)})(I(z), I(\theta_1), I(\theta_2), \ldots I(\theta_m), I_V^*(F_1), I_V^*(F_2), \ldots, I_V^*(F_m)).$$

8. Wenn F eine Negation $(\neg F)$ ist, dann gilt:
$$I_V^*(F) = I(\neg)(I_V^*(F)).$$

9. Wenn F eine existentiell gebundene Formel $(\exists X F_1(X_1, \ldots, X_n))$ ist, dann gilt:
$$I_V^*(F) = I(\vee)(I^*_{\mathbf{V}_X}(F))$$
mit
$\mathbf{V}_X = \{V_X$ ist eine Variablenzuordnung und $X_i \neq X$ impliziert $V_X(X_i) = V(X_i)$ für alle $X_i \in \mathbf{X}\}$. Der Ausdruck $I(\vee)(I^*_{\mathbf{V}_X}(F))$ notiert eine Disjunktion über alle auf F angewendeten Variablenzuordnungen $\mathbf{V}_X$.

10. Wenn F eine universell gebundene Formel $(\forall X F_1(X_1, \ldots, X_n))$ ist, dann gilt:

$$I_V^*(F) = I(\wedge)(I_{\mathbf{V}_X}^*(F))$$

mit

$\mathbf{V}_X = \{V_X$ ist eine Variablenzuordnung und $X_i \neq X$ impliziert $V_X(X_i) = V(X_i)$ für alle $X_i \in \mathbf{X}\}$. Der Ausdruck $I(\wedge)(I_{\mathbf{V}_X}^*(F))$ bedeutet eine Konjunktion über alle auf F angewendeten Variablenzuordnungen $\mathbf{V}_X$.

11. Wenn F eine k-existentiell gebundene Formel $(\exists_k X F_1(X_1, \ldots, X_n))$ ist, dann gilt:

$$I_V^*(F) = I(\vee)(I_{\mathbf{V}_X}^*(F_1)) * \min\left(\frac{\sum_{V_X \in \mathbf{V}_X} I_{V_X}^*(F_1)}{k * I(\vee)(I_{\mathbf{V}_X}^*(F_1))}, 1\right).$$

12. Wenn F eine k-universell gebundene Formel $(\forall_k X F_1(X_1, \ldots, X_n))$ ist, dann gilt:

$$I_V^*(F) = I_V^*((\neg(\exists_k(\neg F_1(X_1, \ldots, X_n))))).$$

13. Wenn F eine *cut*-Operation $(\succ\!\!\!\circ_\theta F_1)$ ist, dann gilt:

$$I_V^*(F) = \begin{cases} 1 & \text{wenn } I_V^*(F_1) \geq I(\theta) \\ 0 & \text{sonst} \end{cases}.$$

14. Wenn F eine *softcut*-Operation $(\tilde{\succ}\!\!\!\circ_\theta F_1)$ ist, dann gilt:

$$I_V^*(F) = I_V^*((\succ\!\!\!\circ_\theta F_1) \wedge F_1).$$

15. Wenn F eine *usercut*-Operation $(\succ\!\!\!\circ_{\theta,f} F_1)$ ist, dann gilt:

$$I_V^*(F) = I(f)(I_V^*(F_1), I(\theta)).$$

□

Aufbauend auf der Semantik der einzelnen Operationen kann die Semantik eines Anfrageausdrucks wie folgt definiert werden:

Definition 5.3.13. Die *Semantik* eines Anfrageausdrucks $E = \{X_1, X_2, \ldots, X_n | F(X_1, X_2, \ldots, X_n)\}$ wird mit $I^*(E)$ notiert und ist definiert als:

$$\{(V.I_V^*(F(X_1, X_2, \ldots, X_n)), V(X_1), V(X_2), \ldots, V(X_n)) | V.I_V^*(F(X_1, X_2, \ldots, X_n)) > 0\}.$$

□

Damit erscheinen nur diejenigen Tupel zusammen mit ihrem Relevanzwert im Ergebnis, deren RSV größer *null* ist. Es ist daher leicht ersichtlich, dass die hier definierte Semantik die Booleschen Anfragen einschließt. Eine Fuzzy-Semantik wird lediglich dann benötigt, wenn kontinuierliche Operationen, wie beispielsweise Ähnlichkeitsprädikate, oder eine Relevanzgewichtung verwendet werden.

5.4 Relationale Ähnlichkeitsalgebra

Die Ähnlichkeitsalgebra $\mathcal{SA}$ bildet als prozedurale Sprache die Basis für die Überführung eines $\mathcal{SDC}$-Ausdrucks. Sie ist jedoch nicht für die direkte Formulierung der Anfragen vorgesehen (Schmitt und Schulz, 2004b). Die Ähnlichkeitsalgebra erweitert die relationale Algebra um die Möglichkeit, unscharfe Anfragen zu formulieren sowie gewichtete Anfragen zu verarbeiten. Im Folgenden wird zunächst die Syntax der Ähnlichkeitsalgebra vorgestellt. Dem folgt die Beschreibung der Semantik der Sprache.

5.4.1 Syntax

Die Syntax der Ähnlichkeitsalgebra ist wie folgt definiert:

Definition 5.4.1. Ein Tupel $(\mathbf{U}, A_0, \Delta, \mathbf{C}, \mathbf{D}, Dom, \mathbf{R}, \mathbf{R}^+, \Theta, \mathbf{Z}, \mathbf{F})$ beschreibt die Ähnlichkeitsalgebra $\mathcal{SA}$, wobei $\mathbf{U} = \{A_1, A_2, \ldots\}$ das Universum der Attribute ist; A_0 das Attribut für den Relevanzwert ist; $\mathbf{X} = \{X_1, X_2, \ldots\}$ eine Menge von Variablen ist; $\Delta = \{\delta_1, \delta_2, \ldots\}$ eine Menge binärer Operationsnamen ist einschließlich des Vergleichsoperators '='; $\mathbf{C}$ eine Menge von Konstantennamen ist; $\mathbf{D}$ eine Menge von Bereichsnamen ist; Dom eine Abbildung von $\mathbf{U} \cup A_0 \cup \mathbf{X} \cup \Delta \cup \mathbf{C}$ nach $\mathbf{D}$ ist; $\mathbf{R}$ eine endliche Menge von Relationenschemata $R_1, R_2, \ldots, R_p$ ist und jedes Relationenschema eine Teilmenge von $\mathbf{U}$ ist; $\mathbf{R}^+$ eine endliche Menge von Fuzzy-Relationenschemata $R_1^+, R_2^+, \ldots, R_p^+$ ist und jedes Fuzzy-Relationenschema sich aus A_0 und einer Teilmenge von $\mathbf{U}$ ergibt; $\Theta = \{\theta_1, \theta_2, \ldots\}$ eine Menge von Gewichtsvariablen ist, $\mathbf{Z} = \{z_1, z_2, \ldots\}$ eine Menge von Operatorgewichtsvariablen und $\mathbf{F} = \{f_1, f_2, \ldots\}$ eine Menge binärer Funktionsnamen ist.

□

Es seien $att(E)$ alle Attribute, die in einem Algebraausdruck E vorkommen. Jedes Attribut wird mit einer Ordnungszahl $\#_i$ entsprechend des Auftretens des jeweiligen Attributs in E bezeichnet.

Die Ähnlichkeitsalgebra umfasst die klassischen Operationen der relationalen Algebra. Zusätzlich dazu werden weitere Operationen, wie beispielsweise der *cut*-Operator, bereitgestellt, um eine adäquate Abbildung des Ähnlichkeitskalküls in die -algebra zu ermöglichen. Die Definition eines Algebraausdrucks ist nachfolgend angegeben.

Definition 5.4.2. Ein *Ausdruck E* der Ähnlichkeitsalgebra ist wie folgt definiert:

1. **0**, wobei die 0-Relation genau ein Tupel mit einem Attributwert enthält. Diese künstliche Relation wird benötigt, um die Bereichsunabhängigkeit zu gewährleisten.

2. **1**, wobei die 1-Relation das neutrale Element bezüglich des Joins und des kartesischen Produktes ist.

3. $R \in \mathbf{R}$,

4. $R^+ \in \mathbf{R}^+$,

5. Dom_D with $D \in \mathbf{D}$,

6. $\pi_{\#_{p_1},\#_{p_2},\ldots,\#_{p_n}}(E)$, wobei E ein Ausdruck der Ähnlichkeitsalgebra ist und $\{\#_{p_1},\#_{p_2},\ldots,\#_{p_n}\} \subseteq att(E)$.

7. $\pi^k_{\#_{p_1},\#_{p_2},\ldots,\#_{p_n}}(E)$, wobei E ein Ausdruck der Ähnlichkeitsalgebra und $k > 1$ eine natürliche Zahl ist. Es gilt $\{\#_{p_1},\#_{p_2},\ldots,\#_{p_n}\} \subseteq att(E)$. Diese Operation ist das Gegenstück zum k-parametrisierten Existenzquantor des $\mathcal{SDC}$.

8. $\sigma_{y_i \delta y_j}(E)$, mit den binären Operationen $\delta \in \Delta$ und $Dom(y_i) = Dom(y_j) = Dom(\delta)$, $\{y_i, y_j\} \subseteq att(E) \cup \mathbf{C}$, wobei E ein Ausdruck der Ähnlichkeitsalgebra ist und δ eine diskrete Operation oder eine kontinuierliche Operation ist. Analog zum $\mathcal{SDC}$ können beide Operanden Konstanten sein, so dass der Relevanzwert direkt manipuliert werden kann.

9. Durchschnitt: Seien $E_1, E_2, \ldots, E_m$ unionkompatible Ausdrücke der Ähnlichkeitsalgebra, dann

   - ist der Durchschnitt als $(E_1 \cap E_2)$ und
   - der gewichtete Durchschnitt als $(\cap_{(\theta_1,\theta_2,\ldots,\theta_m)} E_1, E_2, \ldots, E_m)$ definiert. Die Reihenfolge der Algebraausdrücke E_i ist abhängig von der Gewichtung. Es gilt $\theta_1 \geq \theta_2 \geq \ldots \geq \theta_m$. Bei lediglich zwei Operanden wird anstelle der Präfix-Notation die Infix-Notation verwendet.

10. Vereinigung: Seien $E_1, E_2, \ldots, E_m$ unionkompatible Ausdrücke der Ähnlichkeitsalgebra, dann

    - ist die Vereinigung $(E_1 \cup E_2)$ und
    - die gewichtete Vereinigung $(\cup_{(\theta_1,\theta_2,\ldots,\theta_m)} E_1, E_2, \ldots, E_m)$. Die Reihenfolge der Algebraausdrücke E_i ist abhängig von der Gewichtung. Es gilt $\theta_1 \geq \theta_2 \geq \ldots \geq \theta_m$. Bei nur zwei Operanden kann anstelle der Präfix-Notation die Infix-Notation verwendet werden.

11. Kombination: Seien $E_1, E_2, \ldots, E_m$ unionkompatible Ausdrücke der Ähnlichkeitsalgebra, dann

    - ist die Kombination als $(\uplus^z E_1, E_2, \ldots, E_m)$ und
    - die gewichtete Kombination als $(\uplus^z_{(\theta_1,\theta_2,\ldots,\theta_m)} E_1, E_2, \ldots, E_m)$ definiert. Die Reihenfolge der Algebraausdrücke E_i ist abhängig von der Gewichtung. Es gilt $\theta_1 \geq \theta_2 \geq \ldots \geq \theta_m$.
      Bei einer Kombination mit zwei Operanden kann anstelle der Präfix-Notation die Infix-Notation verwendet werden.

12. $(E_1 \setminus E_2)$, wobei E_1 und E_2 unionkompatible Ausdrücke der Ähnlichkeitsalgebra sind.

13. $(E_1 \times E_2)$, wobei E_1 und E_2 Ausdrücke der Ähnlichkeitsalgebra sind. Da '$\times$' sowohl assoziativ als auch kommutativ ist, kann der Operator zu einem n-ären kartesischen Produkt verallgemeinert werden.

14. $(E_1 \bowtie_{\#_{i_1}=\#_{j_1},\ldots,\#_{i_m}=\#_{j_m}} E_2)$, wobei E_1 und E_2 Ausdrücke der Ähnlichkeitsalgebra sind und $\{\#_{i_1},\ldots,\#_{i_m}\} \subseteq att(E_1), \{\#_{j_1},\ldots,\#_{j_m}\} \subseteq att(E_2)$.

15. $\succ\!\!\!\kappa_\theta(E)$, wobei E ein Ausdruck der Ähnlichkeitsalgebra ist und $\theta \in \Theta$.

16. $\prec\!\!\!\kappa_\theta(E)$, wobei E ein Ausdruck der Ähnlichkeitsalgebra ist und $\theta \in \Theta$.

17. $\succ\!\!\!\kappa_{\theta,f}(E)$, wobei E ein Ausdruck der Ähnlichkeitsalgebra ist und $\theta \in \Theta$ sowie $f \in \mathbf{F}$.

□

5.4.2 Semantik

Die *Interpretation* einer Ähnlichkeitsalgebra $SA(\mathbf{U}, A_0, \Delta, \mathbf{C}, \mathbf{D}, Dom, \mathbf{R}, \mathbf{R}^+, \Theta, \mathbf{Z}, \mathbf{F})$ entspricht der Interpretation des $\mathcal{SDC}$. Es ist somit ein Triple $(\mathbf{d,db}, I)$. Daher wird an dieser Stelle auf die Definition 5.3.9 verwiesen.

Im Folgenden wird die Semantik eines Algebraausdrucks angegeben. Dabei ist anzumerken, dass die Relationen mit einem künstlichen RSV-Attribut $\#_0$ an der ersten Position der Relation ausgestattet sind. Dieses Attribut erlaubt die Verwaltung des Relevanzwerte eines Tupels. Fuzzy-Relationen verfügen bereits über ein solches Attribut für den Relevanzwert, so dass hier kein künstliches Attribut eingeführt werden muss.

Definition 5.4.3. Die *Semantik* eines Algebraausdrucks E ist mittels einer Interpretationsfunktion I^* wie folgt induktiv definiert:

1. *0-Relation* $E = \mathbf{0}$:
$$I^*(\mathbf{0}) = \{(1, 0)\}$$
ist eine Relation mit genau einem Tupel, dessen Relevanzwert 1 beträgt, und einem beliebigen Attributwert, vorzugsweise 0.

2. *1-Relation* $E = \mathbf{1}$:
$$I^*(\mathbf{1}) = \{(1)\}$$
ist eine Relation mit genau einem Tupel. Der RSV des Tupels beträgt 1. Ansonsten besitzt die Relation keine weiteren Attribute.

3. *Relationenname* $E = R \in \boldsymbol{R}$:
$$I^*(R) = \{(1, v_1, v_2, \ldots, v_n) | (v_1, v_2, \ldots, v_n) \in I(R)\}$$
mit den Attributen $A_1, A_2, \ldots, A_n$ von R. Alle Tupel werden mit einem RSV von 1 ausgestattet. Das bedeutet, dass sie zunächst als wahre Fakten betrachtet werden. Damit wird eine Relation R in eine Fuzzy-Relation R^+ überführt, wobei jedem Tupel der maximale Relevanzwert von 1 zugeteilt wird.

4. *Relationenname* $E = R^+ \in \mathbf{R}^+$:
$$I^*(R^+) = \{(v_0, v_1, v_2, \ldots, v_n) | (v_0, v_1, v_2, \ldots, v_n) \in I(R^+)\}$$
mit den Attributen $A_0, A_1, A_2, \ldots, A_n$ von R^+. Der Relevanzwert für ein Tupel entspricht v_0.

5. *Bereichsname* $E = Dom_D$:

$$I^*(Dom_D) = \{(1, v) | v \in I(D)\}.$$

Jedem Wertebereiche wird ebenfalls ein Relevanzwert von 1 zugeordnet.

6. *Projektion* $E = \pi_{\#_{p_1},\#_{p_2},\ldots,\#_{p_n}}(E_1)$: $v_{0_1}, \ldots, v_{0_l}$ sind die Relevanzwerte für eine Werteliste $v_{p_1}, \ldots, v_{p_n}$, wobei $(v_{0_i}, v_1, \ldots, v_m) \in I^*(E_1)$ gilt und die entsprechenden Werte identisch sind: $p_i = j \rightarrow v_{p_i} = v_j$ mit $i = 1, \ldots, n$. Damit ergibt sich

$$I^*(\pi_{\#_{p_1},\#_{p_2},\ldots,\#_{p_n}}(E_1)) = \left\{(u_0, v_{p_1}, v_{p_2}, \ldots, v_{p_n}) | (v_{0_i}, v_1, \ldots, v_m) \in I^*(E_1)\right\}$$

mit

$$u_0 = \begin{cases} I(\vee)(v_{0_1}, \ldots, v_{0_l}) & \text{if } l > 1 \\ v_{0_1} & \text{if } l = 1 \end{cases}.$$

Bei der Projektion findet eine Duplikateliminierung statt. Diese basiert auf der Disjunktion der Relevanzwerte ansonsten gleicher Tupel. Für diesen Zweck kann ein n-ärer *ODER*-Operator eingesetzt werden.

7. *Parametrisierbare Projektion* $E = \pi^k_{\#_{p_1},\#_{p_2},\ldots,\#_{p_n}}(E_1)$: $v_{0_1}, \ldots, v_{0_l}$ sind die Relevanzwerte für eine Werteliste $v_{p_1}, \ldots, v_{p_n}$, wobei $(v_{0_i}, v_1, \ldots, v_m) \in I^*(E_1)$ gilt und die entsprechenden Werte identisch sind: $p_i = j \rightarrow v_{p_i} = v_j$ mit $i = 1, \ldots, n$. Damit ergibt sich

$$I^*(\pi_{\#_{p_1},\#_{p_2},\ldots,\#_{p_n}}(E_1)) = \{(u_0, v_{p_1}, v_{p_2}, \ldots, v_{p_n}) | (v_{0_i}, v_1, \ldots, v_m) \in I^*(E_1)\}$$

mit

$$u_0 = I(\vee)(v_{0_1}, \ldots, v_{0_l}) * \min\left(\frac{v_{0_1} + \ldots + v_{0_l}}{k * I(\vee)(v_{0_1}, \ldots, v_{0_l})}, 1\right).$$

8. *Selektion* $E = \sigma_{y_i \delta y_j}(E_1)$:

$$I^*(\sigma_{y_i \delta y_j}(E_1)) = \{(u_0, v_1, \ldots, v_n) | (v_0, v_1, \ldots, v_n) \in I^*(E_1) \wedge u_0 = I(\wedge)(v_0, I(\delta)(\hat{y}_i, \hat{y}_j))\}$$

mit

$$\hat{y}_i = \begin{cases} v_i & \text{wenn } y_i \text{ ein Attribut ist} \\ I(y_i) & \text{wenn } y_i \text{ ein Konstantenname ist} \end{cases}$$

und

$$\hat{y}_j = \begin{cases} v_j & \text{wenn } y_j \text{ ein Attribut ist} \\ I(y_j) & \text{wenn } y_j \text{ ein Konstantenname ist.} \end{cases}$$

Bei einer Selektion ohne eine Bedingung findet keine Einschränkung statt, so dass gilt:

$$I^*(\sigma(E_1)) = I^*(E_1).$$

9. *Durchschnitt* $E = (E_1 \cap E_2)$:

$$I^*((E_1 \cap E_2)) =$$
$$\{(I(\wedge)(v_0, w_0), v_1, \ldots, v_k) | (v_0, v_1, \ldots, v_k) \in I^*(E_1) \wedge (w_0, v_1, \ldots, v_k) \in I^*(E_2)\}.$$

10. *Gewichteter Durchschnitt* $E = (\cap_{(\theta_1,\theta_2,\ldots,\theta_m)} E_1, E_2, \ldots, E_m)$:

Aus Gründen der Übersichtlichkeit wird die Semantik des gewichteten Durchschnitts exemplarisch für $E = (E_1 \cap_{(\theta_1,\theta_2)} E_2)$ angegeben. Die allgemeine Definition für den gewichteten Durchschnitt ist im Anhang A (Definition A.1) aufgeführt. Es gilt:

$$I^*((E_1 \cap_{(\theta_1,\theta_2)} E_2)) =$$
$$\{(\star_1, v_1, \ldots, v_k) | (v_0, v_1, \ldots, v_k) \in I^*(E_1) \wedge (w_0, v_1, \ldots, v_k) \in I^*(E_2)\} \cup$$
$$\{(\star_2, v_1, \ldots, v_k) | (v_0, v_1, \ldots, v_k) \in I^*(E_1) \wedge \forall w_0.(w_0, v_1, \ldots, v_k) \notin I^*(E_2)\}$$

mit

$$\begin{aligned} \star_1 &= I(\wedge_{(\theta_1,\theta_2)})(I(\theta_1)I(\theta_2), v_0, w_0) \\ \star_2 &= I(\wedge_{(\theta_1,\theta_2)})(I(\theta_1)I(\theta_2), v_0, 0). \end{aligned}$$

Der gewichtete Durchschnitt entspricht somit nicht dem normalen Durchschnitt. Denn auf Grund der Gewichtung erzeugt eine Konjunktion mit einem weniger stark gewichteten Operanden (E_2), der einen RSV von 0 besitzt, einen neuen Relevanzwert, welcher ungleich 0 ist. Daher werden all die Tupel, die in E_1 und E_2 sowie die Tupel, die nur in E_1, also der am stärksten gewichteten Menge, enthalten sind, zum Ergebnis gerechnet.

Analog dazu verhält es sich mit $(\cap_{(\theta_1,\theta_2,\ldots,\theta_m)} E_1, E_2, \ldots, E_m)$. Die Ergebnismenge enthält zum einen all die Tupel, die in jeder Menge $E_1, E_2, \ldots, E_m$ erscheinen und zusätzlich dazu die Tupel von E_1. Der Relevanzwert ergibt sich für jedes Tupel unter Verwendung des RSV-Attributs $\#_0$ an der ersten Stelle der Relation beziehungsweise des Wertes 0, falls das Tupel nicht in der entsprechenden Menge enthalten ist. Dieser Unterschied zwischen dem Durchschnitt und dem gewichteten Durchschnitt ist in Abbildung 5.3 dargestellt.

Abbildung 5.3: Unterschied zwischen a) dem Durchschnitt $(E_1 \cap E_2)$ und b) dem gewichteten Durchschnitt $(E_1 \cap_{(\theta_1,\theta_2)} E_2)$

11. *Vereinigung* $E = (E_1 \cup E_2)$:

$$
\begin{aligned}
&I^*((E_1 \cup E_2)) = \\
&\{(I(\vee)(v_0, w_0), v_1, \ldots, v_k) | (v_0, v_1, \ldots, v_k) \in I^*(E_1) \wedge (w_0, v_1, \ldots, v_k) \in I^*(E_2)\} \cup \\
&\{(I(\vee)(v_0, 0), v_1, \ldots, v_k) | (v_0, v_1, \ldots, v_k) \in I^*(E_1) \wedge \forall w_0.(w_0, v_1, \ldots, v_k) \notin I^*(E_2)\} \cup \\
&\{(I(\vee)(0, w_0), v_1, \ldots, v_k) | (w_0, v_1, \ldots, v_k) \in I^*(E_2) \wedge \forall v_0.(v_0, v_1, \ldots, v_k) \notin I^*(E_1)\}.
\end{aligned}
$$

12. *Gewichtete Vereinigung* $E = (\cup_{(\theta_1, \theta_2, \ldots, \theta_m)} E_1, E_2, \ldots E_m)$:

Aus Gründen der Übersichtlichkeit wird die Semantik der gewichteten Vereinigung exemplarisch für $E = (E_1 \cup_{(\theta_1, \theta_2)} E_2)$ angegeben. Die allgemeine Definition ist im Anhang A (Definition A.2) aufgeführt. Für zwei Operanden gilt:

$$
\begin{aligned}
&I^*(E_1 \cup_{(\theta_1, \theta_2)} E_2) = \\
&\{(\star_1, v_1, \ldots, v_k) | (v_0, v_1, \ldots, v_k) \in I^*(E_1) \wedge (w_0, v_1, \ldots, v_k) \in I^*(E_2)\} \cup \\
&\{(\star_2, v_1, \ldots, v_k) | (v_0, v_1, \ldots, v_k) \in I^*(E_1) \wedge \forall w_0.(w_0, v_1, \ldots, v_k) \notin I^*(E_2)\} \cup \\
&\{(\star_3, v_1, \ldots, v_k) | (w_0, v_1, \ldots, v_k) \in I^*(E_2) \wedge \forall v_0.(v_0, v_1, \ldots, v_k) \notin I^*(E_1)\}
\end{aligned}
$$

mit

$$
\begin{aligned}
\star_1 &= I(\vee_{(\theta_1, \theta_2)})(I(\theta_1), I(\theta_2), v_0, w_0) \\
\star_2 &= I(\vee_{(\theta_1, \theta_2)})(I(\theta_1), I(\theta_2), v_0, 0) \\
\star_3 &= I(\vee_{(\theta_1, \theta_2)})(I(\theta_1), I(\theta_2), 0, w_0).
\end{aligned}
$$

Bei der gewichteten Disjunktion werden die Fuzzy-Mengen vereinigt. Befindet sich ein Tupel in einer Menge E_j, so wird dessen Relevanzwert, andernfalls der Wert 0 zur Berechnung des neuen Relevanzwertes verwendet. Der Unterschied zwischen der Vereinigung und der gewichteten Vereinigung besteht demnach nur in der Art und Weise, wie die Berechnung des Relevanzwertes für die Ergebnistupel erfolgt.

13. *Kombination* $E = (\uplus^z E_1, E_2, \ldots, E_m)$:

Aus Gründen der Übersichtlichkeit wird die Semantik der gewichteten Vereinigung exemplarisch für $E = (E_1 \uplus^z E_2)$ angegeben. Die allgemeine Definition ist im Anhang A (Definition A.3) aufgeführt. Bei zwei Operanden ergibt sich:

$$
\begin{aligned}
&I^*((E_1 \uplus^z E_2)) = \\
&\{(I(\uplus^z)(I(z), v_0, w_0), v_1, \ldots, v_k) | (v_0, v_1, \ldots, v_k) \in I^*(E_1) \wedge (w_0, v_1, \ldots, v_k) \in I^*(E_2)\} \cup \\
&\{(I(\uplus^z)(I(z), v_0, 0), v_1, \ldots, v_k) | (v_0, v_1, \ldots, v_k) \in I^*(E_1) \wedge \forall w_0.(w_0, v_1, \ldots, v_k) \notin I^*(E_2)\} \cup \\
&\{(I(\uplus^z)(I(z), 0, w_0), v_1, \ldots, v_k) | (w_0, v_1, \ldots, v_k) \in I^*(E_2) \wedge \forall v_0.(v_0, v_1, \ldots, v_k) \notin I^*(E_1)\}.
\end{aligned}
$$

14. *Gewichtete Kombination* $E = (\uplus^z_{(\theta_1, \theta_2, \ldots, \theta_m)} E_1, E_2, \ldots, E_m)$:

Auch hier wird aus Gründen der Übersichtlichkeit die Semantik der gewichteten Kombination beispielhaft für $E = (\uplus^z_{(\theta_1, \theta_2)} E_1, E_2)$ angegeben. Die allgemeine Definition ist

im Anhang A (Definition A.4) aufgeführt. Für zwei Operanden gilt:

$$
\begin{aligned}
&I^*(\uplus^z_{(\theta_1,\theta_2)} E_1, E_2) = \\
&\{(\star_1, v_1, \ldots, v_k) | (v_0, v_1, \ldots, v_k) \in I^*(E_1) \wedge (w_0, v_1, \ldots, v_k) \in I^*(E_2)\} \cup \\
&\{(\star_2, v_1, \ldots, v_k) | (v_0, v_1, \ldots, v_k) \in I^*(E_1) \wedge \forall w_0.(w_0, v_1, \ldots, v_k) \notin I^*(E_2)\} \cup \\
&\{(\star_3, v_1, \ldots, v_k) | (w_0, v_1, \ldots, v_k) \in I^*(E_2) \wedge \forall v_0.(v_0, v_1, \ldots, v_k) \notin I^*(E_1)\}
\end{aligned}
$$

mit

$$
\begin{aligned}
\star_1 &= I(\uplus^z_{(\theta_1,\theta_2)})(I(z), I(\theta_1), I(\theta_2), v_0, w_0) \\
\star_2 &= I(\uplus^z_{(\theta_1,\theta_2)})(I(z), I(\theta_1), I(\theta_2), v_0, 0) \\
\star_3 &= I(\uplus^z_{(\theta_1,\theta_2)})(I(z), I(\theta_1), I(\theta_2), 0, w_0).
\end{aligned}
$$

15. *Differenz* $E = (E_1 \setminus E_2)$:

$$
\begin{aligned}
&I^*((E_1 \setminus E_2)) = \\
&\{(I(\wedge)(v_0, I(\neg)(0)), v_1, \ldots, v_k) | (v_0, v_1, \ldots, v_k) \in I^*(E_1) \wedge \forall w_0.(w_0, v_1, \ldots, v_k) \notin I^*(E_2)\} \\
&\cup \{(I(\wedge)(v_0, I(\neg)(w_0)), v_1, \ldots, v_k) | (v_0, v_1, \ldots, v_k) \in I^*(E_1) \wedge (w_0, v_1, \ldots, v_k) \in I^*(E_2)\}.
\end{aligned}
$$

16. *Kartesisches Produkt* $E = E_1 \times E_2$:

$$
\begin{aligned}
&I^*(E_1 \times E_2) = \\
&\{(u_0, v_1, \ldots, v_k, w_1, \ldots, w_l) | (v_0, v_1, \ldots, v_k) \in I^*(E_a) \wedge (w_0, w_1, \ldots, w_l) \in I^*(E_b) \wedge \\
&u_0 = I(\wedge)(v_0, w_0)\}.
\end{aligned}
$$

17. *Join* $E = E_1 \bowtie_{\#_{1_1}=\#_{2_1},\ldots,\#_{1_n}=\#_{2_n}} E_2$:

$$
\begin{aligned}
&I^*(E_1 \bowtie_{\#_{1_1}=\#_{2_1},\ldots,\#_{1_n}=\#_{1_n}} E_2) = \\
&\{(u_0, v_1, \ldots, v_k, w_1, \ldots, w_l) | (v_0, v_1, \ldots, v_k) \in I^*(E_1) \wedge (w_0, w_1, \ldots, w_l) \in I^*(E_2) \wedge \\
&\forall i \in \{1, \ldots, n\}.v_{1_i} = w_{2_i} \wedge u_0 = I(\wedge)(v_0, w_0)\}.
\end{aligned}
$$

Ein Join mit einer leeren Bedingung resultiert im kartesischen Produkt.

18. *cut-Operation* $E = \succ\!\!\!\circ_\theta(E_1)$:

$$
\begin{aligned}
&I^*(\succ\!\!\!\circ_\theta(E_1)) = \\
&\left\{(u_0, v_1, \ldots, v_m) | (v_0, v_1, \ldots, v_m) \in I^*(E_1) \wedge u_0 = \left\{ \begin{array}{ll} 1 & \text{wenn } v_0 \geq I(\theta) \\ 0 & \text{sonst} \end{array} \right\}\right.
\end{aligned}
$$

19. *softcut-Operation* $E = \tilde{\succ}\!\!\!\circ_\theta(E_1)$:

$$
\begin{aligned}
&I^*(\tilde{\succ}\!\!\!\circ_\theta(E_1)) = \\
&\left\{(u_0, v_1, \ldots, v_m) | (v_0, v_1, \ldots, v_m) \in I^*(E_1) \wedge u_0 = \left\{ \begin{array}{ll} v_0 & \text{wenn } v_0 \geq I(\theta) \\ 0 & \text{sonst} \end{array} \right\}\right.
\end{aligned}
$$

20. *usercut-Operation* $E = \succ\!\!\!\cdot_{\theta,f}(E_1)$:

$$I^*(\succ\!\!\!\cdot_{\theta,f}(E_1)) = \{(u_0, v_1, \ldots, v_m) | (v_0, v_1, \ldots, v_m) \in I^*(E_1) \wedge u_0 = I(f)(v_0, I(\theta))\}\,.$$

□

Mit den hier beschriebenen Operatoren und der diesbezüglich definierten Semantik weist die Ähnlichkeitsalgebra die gleiche Anfragemächtigkeit wie das Ähnlichkeitskalkül auf. Damit können die Ausdrücke des Kalküls in entsprechende Ausdrücke der Algebra überführt werden. Darauf wird im folgenden Abschnitt näher eingegangen.

5.5 Überführung eines Kalkülausdrucks in einen Algebraausdruck

In diesem Abschnitt steht die Transformation eines Ausdrucks des Ähnlichkeitskalküls in einen Ausdruck der Ähnlichkeitsalgebra im Vordergrund. Dafür wird zunächst die Vorgehensweise bei der Reduktion beschrieben. Im Anschluss daran wird dies an einem Beispiel verdeutlicht. Dabei wird dargestellt, wie Anfragen im $\mathcal{SDC}$ formuliert werden, in einen $\mathcal{SA}$-Ausdruck überführt und letztendlich mittels der Verwendung konkreter Daten, Operationen und Scoring-Funktionen evaluiert werden.

5.5.1 Reduktion

Die Reduktion eines Kalkülausdrucks in einen äquivalenten Ausdruck der Ähnlichkeitsalgebra wurde bereits von Schmitt und Schulz (2004a,b) veröffentlicht. Auf Grund der in dieser Arbeit vorgenommenen Erweiterungen beziehungsweise Modifizierungen des Ähnlichkeitskalküls und damit verbunden auch der Ähnlichkeitsalgebra müssen diese Aspekte bei der Reduktion Berücksichtigung finden.

Grundlage für diese Überführung bildet die Vorgehensweise bei der Reduktion eines Ausdrucks des Bereichskalküls in einen entsprechenden Ausdruck der Relationenalgebra. Hierfür sei auf die ausführlichen Darstellungen von Maier (1983) sowie Ullman (1988) verwiesen.

Definition 5.5.1. Die *Abbildung* φ einer $\mathcal{SDC}$ Formel F in die Ähnlichkeitsalgebra ist wie folgt induktiv definiert:

1. Wenn $F = R(Y_1, Y_2, \ldots, Y_m) : \varphi(F) = \pi_{\#_{v_1},\ldots,\#_{v_n}}(\sigma_{cond_c,cond_v}(R))$ mit
   - $cond_c := \#_{i_1} = C_{i_1}, \ldots, \#_{i_l} = C_{i_l}$, wobei $Y_{i_1}, \ldots, Y_{i_l}$ Konstantennamen $C_{i_1}, \ldots, C_{i_l}$ sind,
   - $cond_v := \#_{j_1} = \#_{k_1}, \ldots, \#_{j_o} = \#_{k_o}$, wobei $Y_{j_1} = Y_{k_1}, \ldots, Y_{j_o} = Y_{k_o}$ unter Beachtung der Reflexivität, Symmetrie und Transitivität der Attributpositionen jeweils Paare von gleich benannten Variablen sind.
   - $\#_{v_1}, \ldots, \#_{v_n}$ führt eindeutig die entsprechenden Attribute für alle Variablen Y_{v_i} in der richtigen Reihenfolge auf.

   Dies gilt analog für eine Fuzzy-Relation $F = R(Y_0, Y_1, Y_2, \ldots, Y_m)$.
2. Wenn $F = Y_1 \delta Y_2$:

- und $Y_1, Y_2 \in \mathbf{X} \wedge Y_1 \neq Y_2$, dann gilt $\varphi(F) = \sigma_{\#_1 \delta \#_2}(Dom_D \times Dom_D)$.
- und $Y_1, Y_2 \in \mathbf{X} \wedge Y_1 = Y_2$, dann gilt $\varphi(F) = \pi_{\#_1}(\sigma_{\#_1 \delta \#_2}(\pi_{\#_1,\#_1}(Dom_D)))$.
- und $Y_1 \in \mathbf{X}$ und Y_2 eine Konstante $C \in \mathbf{C}$ ist, dann gilt $\varphi(F) = \sigma_{\#_1 \delta C}(Dom_D)$.
- und $Y_2 \in \mathbf{X}$ und Y_1 eine Konstante $C \in \mathbf{C}$ ist, dann gilt $\varphi(F) = \sigma_{C \delta \#_1}(Dom_D)$.
- und Y_1 und Y_2 eine Konstante $C_1, C_2 \in \mathbf{C}$ ist, dann gilt $\varphi(F) = \sigma_{C_1 \delta C_2}(\mathbf{1})$.

3. Wenn $F = (F_1(X_{1_1}, \ldots, X_{1_k}) \wedge F_2(X_{2_1}, \ldots, X_{2_l})) : \varphi(F) = \pi_{\#_{p_1},\ldots,\#_{p_n}}(\varphi(F_1) \bowtie_{cond_c} \varphi(F_2))$, wobei
   - $cond_c := \#_{v_1} = \#_{w_1}, \ldots, \#_{v_n} = \#_{w_n}$ mit $X_{1_{v_i}} = X_{2_{w_i}}$ und $i = 1, \ldots, n$.
   - $\#_{p_1}, \ldots, \#_{p_n}$ führt eindeutig die entsprechenden Attribute für alle Variablen X_{1_1}, ..., X_{1_k},X_{2_1},...,X_{2_l} in der richtigen Reihenfolge auf.
4. Wenn $F = (F_1(X_{1_1}, \ldots, X_{1_k}) \vee F_2(X_{2_1}, \ldots, X_{2_l})) : \varphi(F) = E_1 \cup E_2$
   mit
   $$E_1 = \pi_{\#_{i_1},\ldots,\#_{i_m}}(\varphi(F_1) \times Dom_{D_{2_{o_1}}} \times \ldots \times Dom_{D_{2_{o_{t-k}}}})$$
   $$E_2 = \pi_{\#_{j_1},\ldots,\#_{j_m}}(\varphi(F_2) \times Dom_{D_{1_{p_1}}} \times \ldots \times Dom_{D_{1_{p_{t-l}}}})$$
   unter den Bedingungen:
   - $\{X_{1_1}, \ldots, X_{1_k}, X_{2_1}, \ldots, X_{2_l}\}$ korrespondiert bijektiv zu $att(E_1)$ und dementsprechend auch zu $att(E_2)$,
   - die Attribute von E_1 und E_2 erscheinen in der gleichen Reihenfolge,
   - $t = |\{X_{1_1}, \ldots, X_{1_k}, X_{2_1}, \ldots, X_{2_l}\}|$ und
   - $D_{1_{p_i}} = Dom(X_{1_{p_i}})$ and $D_{2_{o_i}} = Dom(X_{2_{o_i}})$.
5. Gewichtete Konjunktion, gewichtete Disjunktion und Kombinationen
   - wenn $F = (\wedge_{(\theta_1,\theta_2,\ldots,\theta_m)} F_1(X_{1_1}, \ldots, X_{1_k}), F_2(X_{2_1}, \ldots, X_{2_l}), \ldots, F_m(X_{m_1}, \ldots, X_{m_r})) : \varphi(F) = \cap_{(\theta_1,\theta_2,\ldots,\theta_m)} E_1, E_2, \ldots, E_m$
   - wenn $F = (\vee_{(\theta_1,\theta_2,\ldots,\theta_m)} F_1(X_{1_1}, \ldots, X_{1_k}), F_2(X_{2_1}, \ldots, X_{2_l}), \ldots, F_m(X_{m_1}, \ldots, X_{m_r})) : \varphi(F) = \cup_{(\theta_1,\theta_2,\ldots,\theta_m)} E_1, E_2, \ldots, E_m$
   - wenn $F = (\oplus^z F_1(X_{1_1}, \ldots, X_{1_k}), F_2(X_{2_1}, \ldots, X_{2_l}), \ldots, F_m(X_{m_1}, \ldots, X_{m_r})) : \varphi(F) = \uplus^z E_1, E_2, \ldots, E_m$
   - wenn $F = (\oplus^z_{(\theta_1,\theta_2,\ldots,\theta_m)} F_1(X_{1_1}, \ldots, X_{1_k}), F_2(X_{2_1}, \ldots, X_{2_l}), \ldots, F_m(X_{m_1}, \ldots, X_{m_r})) : \varphi(F) = \uplus^z_{(\theta_1,\theta_2,\ldots,\theta_m)} E_1, E_2, \ldots, E_m$

   mit

   $E_1 = \pi_{\#_{i_1},\ldots,\#_{i_m}}(\varphi(F_1) \times Dom_{D_{2_{o_1}}} \times \ldots \times Dom_{D_{2_{o_{t-k}}}} \times \ldots \times Dom_{D_{m_{q_1}}} \times \ldots \times Dom_{D_{m_{q_{t-r}}}})$
   $E_2 = \pi_{\#_{j_1},\ldots,\#_{j_m}}(\varphi(F_2) \times Dom_{D_{1_{p_1}}} \times \ldots \times Dom_{D_{1_{p_{t-l}}}} \times \ldots \times Dom_{D_{m_{q_1}}} \times \ldots \times Dom_{D_{m_{q_{t-r}}}})$
   ...
   $E_m = \pi_{\#_{g_1},\ldots,\#_{g_m}}(\varphi(F_m) \times Dom_{D_{2_{o_1}}} \times \ldots \times Dom_{D_{2_{o_{t-k}}}} \times \ldots \times Dom_{D_{1_{p_1}}} \times \ldots \times Dom_{D_{1_{p_{t-l}}}})$

   unter den Bedingungen:

- $\{X_{1_1}, \ldots, X_{1_k}, X_{2_1}, \ldots, X_{2_l}, \ldots, X_{m_1}, \ldots, X_{m_r}\}$ korrespondiert bijektiv zu $att(E_1)$, zu $att(E_2)$ beziehungsweise zu $att(E_m)$,
- die Attribute von E_1, $E_2 \ldots E_m$ erscheinen in der gleichen Reihenfolge,
- $t = |\{X_{1_1}, \ldots, X_{1_k}, X_{2_1}, \ldots, X_{2_l}, \ldots, X_{m_1}, \ldots, X_{m_r}\}|$und
- $D_{1_{p_i}} = Dom(X_{1_{p_i}})$, $D_{2_{o_i}} = Dom(X_{2_{o_i}})$ und $D_{m_{q_i}} = Dom(X_{m_{q_i}})$.

6. Wenn $F = (\neg F_1(X_1, \ldots, X_n)) : \varphi(F) = (Dom_{D_1} \times \ldots \times Dom_{D_m}) \setminus \varphi(F_1)$, wobei $D_i = Dom(X_i)$.
7. Wenn $F = (\exists X F_1(X, X_1, \ldots, X_n)) : \varphi(F) = \pi_{\#_{p_1}, \ldots, \#_{p_n}}(\varphi(F_1(X, X_1, \ldots, X_n)))$, wobei $\#_{p_1}, \ldots, \#_{p_n}$ eindeutig die entsprechenden Attribute für alle Variablen $X_1, \ldots, X_n$ in der richtigen Reihenfolge aufführt.
8. Wenn $F = (\forall X F_1) : \varphi(F) = \varphi((\neg(\exists X(\neg F_1))))$.
9. Wenn $F = (\exists_k X F_1(X, X_1, \ldots, X_n)) : \varphi(F) = \pi^k_{\#_{p_1}, \ldots, \#_{p_n}}(\varphi(F_1(X, X_1, \ldots, X_n)))$, wobei $\#_{p_1}, \ldots, \#_{p_n}$ eindeutig die entsprechenden Attribute für alle Variablen $X_1, \ldots, X_n$ in der richtigen Reihenfolge aufführt.
10. Wenn $F = (\forall_k X F_1) : \varphi(F) = \varphi((\neg(\exists_k X(\neg F_1))))$.
11. Wenn $F = (\succ\!\!\!\circ_\theta F_1) : \varphi(F) = \succ\!\!\!\circ_\theta(\varphi(F_1))$.
12. Wenn $F = (\prec\!\!\!\circ_\theta F_1) : \varphi(F) = \prec\!\!\!\circ_\theta(\varphi(F_1))$.
13. Wenn $F = (\succ\!\!\!\circ_{\theta,f} F_1) : \varphi(F) = \succ\!\!\!\circ_{\theta,f}(\varphi(F_1))$.

□

Unter Berücksichtigung dieser Ersetzungsregeln kann die Reduktion eines Kalkülausdrucks in einen entsprechenden Algebraausdruck wie im Folgenden beschrieben stattfinden.

Dafür ist $E = \{X_1, X_2, \ldots, X_n | F(X_1, X_2, \ldots, X_n)\}$ eine evaluierbare $\mathcal{SDC}$-Anfrage und

$$rel(X) = (E_1) \cap \ldots \cap (E_m)$$

ist eine Funktion auf den in F genutzten Variablen, welche die entsprechenden Datenbankwerte ermittelt. Ein $\mathcal{SA}$-Ausdruck E_i mit $i = 1, \ldots, m$ bezüglich X kann, wenn X frei in F ist, unter Verwendung der *gen*-Relation $\{E_i | gen(X, F, E_i)\}$ erzeugt werden. Andernfalls wird der Ausdruck aus der *con*-Relation $\{E_i | con(X, F, E_i)\}$ erzeugt. Da das Symbol '$\perp$' in der *con*-Relation für eine leere Relation steht, wird es bei einer Disjunktion innerhalb von $rel(X)$ entfernt.

Definition 5.5.2. Die *Abbildung* $\varphi^*(F)$ einer evaluierbaren $\mathcal{SDC}$-Formel $F(X_1, X_2, \ldots, X_n)$ basiert auf einer Modifikation im Hinblick auf die Abbildung $\varphi(F)$. Daher gilt:

1. Für jedes Atom $Y_1 \delta Y_2$ innerhalb von F wird der entsprechende Algebraterm

   a) $\sigma_{\#_1 \delta \#_2}(Dom_D \times Dom_D)$ in $\varphi(F)$ durch $\sigma_{\#_1 \delta \#_2}(rel'(Y_1) \times rel(Y_2))$, wenn $Y_1 \neq Y_2$ und $Y_1, Y_2 \in \mathbf{X}$,

b) $\pi_{\#_1}(\sigma_{\#_1\delta\#_2}(\pi_{\#_1,\#_1}(Dom_D)))$ in $\varphi(F)$ durch $\pi_{\#_1}(\sigma_{\#_1\delta\#_2}(\pi_{\#_1,\#_1}(rel'(Y_1))))$, wenn $Y_1 = Y_2$ und $Y_1, Y_2 \in \mathbf{X}$,

c) $\sigma_{\#_1\delta C}(Dom_D)$ in $\varphi(F)$ durch $\sigma_{\#_1\delta Y_2}(rel(Y_1))$, wenn $Y_1 \in \mathbf{X}, Y_2 \in \mathbf{C}$

d) $\sigma_{C\delta\#_1}(Dom_D)$ in $\varphi(F)$ durch $\sigma_{Y_1\delta\#_1}(rel(Y_2))$, wenn $Y_1 \in \mathbf{C}, Y_2 \in \mathbf{X}$

ersetzt.

2. Disjunktion:

   Für jede Disjunktion $(F_1(X_{1_1}, \ldots, X_{1_k}) \vee F_2(X_{2_1}, \ldots, X_{2_l}))$ in F wird der entsprechende Algebraterm $E_1 \cup E_2$ in $\varphi(F)$ durch $E_1' \cup E_2'$ ersetzt, mit

$$E_1' = \pi_{\#_{i_1},\ldots,\#_{i_m}}(\varphi(F_1) \times rel'(X_{2_{o_1}}) \times \ldots \times rel'(X_{2_{o_{t-k}}}))$$
$$E_2' = \pi_{\#_{j_1},\ldots,\#_{j_m}}(\varphi(F_2) \times rel'(X_{1_{p_1}}) \times \ldots \times rel'(X_{1_{p_{t-l}}}))$$

   unter den Bedingungen:

   - $\{X_{1_1}, \ldots, X_{1_k}, X_{2_1}, \ldots, X_{2_l}\}$ korrespondiert bijektiv zu $att(E_1')$ und dementsprechend auch zu $att(E_2')$,
   - die Attribute von E_1' und E_2' erscheinen in der gleichen Reihenfolge,
   - $t = |\{X_{1_1}, \ldots, X_{1_k}, X_{2_1}, \ldots, X_{2_l}\}|$ und
   - $rel'(X) = \begin{cases} (rel(X) \cup \mathbf{0}) & \text{wenn } X \text{ an } \exists \text{ or } \forall \text{ gebunden ist} \\ rel(X) & \text{sonst} \end{cases}$.

3. Gewichtete Konjunktion, gewichtete Disjunktion und Kombinationen.

   Für jede gewichtete Konjunktion $F = (\wedge_{(\theta_1,\theta_2,\ldots,\theta_m)} F_1(X_{1_1}, \ldots, X_{1_k}), F_2(X_{2_1}, \ldots, X_{2_l}), \ldots, F_m(X_{m_1}, \ldots, X_{m_r}))$ in F wird der entsprechende Algebraterm $\cap_{(\theta_1,\theta_2,\ldots,\theta_m)} E_1, E_2, \ldots E_m$ in $\varphi(F)$ durch $\cap_{(\theta_1,\theta_2,\ldots,\theta_m)} E_1', E_2', \ldots E_n'$ ersetzt, mit

$$\begin{aligned}
E_1' &= \pi_{\#_{i_1},\ldots,\#_{i_m}}(\varphi(F_1) \times rel'(X_{2_{o_1}}) \times \ldots \times rel'(X_{2_{o_{t-k}}}) \times \ldots \times rel'(X_{m_{q_1}}) \times \ldots \times rel'(X_{m_{q_{t-r}}})) \\
E_2' &= \pi_{\#_{j_1},\ldots,\#_{j_m}}(\varphi(F_2) \times rel'(X_{1_{p_1}}) \times \ldots \times rel'(X_{1_{p_{t-l}}}) \times \ldots \times rel'(X_{m_{q_1}}) \times \ldots \times rel'(X_{m_{q_{t-r}}})) \\
&\ldots \\
E_m' &= \pi_{\#_{k_1},\ldots,\#_{k_m}}(\varphi(F_m) \times rel'(X_{2_{o_1}}) \times \ldots \times rel'(X_{2_{o_{t-k}}}) \times \ldots \times rel'(X_{1_{p_1}}) \times \ldots \times rel'(X_{1_{p_{t-l}}}))
\end{aligned}$$

   unter den Bedingungen:

   - $\{X_{1_1}, \ldots, X_{1_k}, X_{2_1}, \ldots, X_{2_l}, \ldots, X_{m_1}, \ldots, X_{m_r}\}$ korrespondiert bijektiv zu $att(E_1)$, zu $att(E_2)$ beziehungsweise zu $att(E_m)$,
   - die Attribute von E_1, $E_2 \ldots E_m$ erscheinen in der gleichen Reihenfolge,
   - $t = |\{X_{1_1}, \ldots, X_{1_k}, X_{2_1}, \ldots, X_{2_l}, \ldots, X_{m_1}, \ldots, X_{m_r}\}|$ und
   - $rel'(X) = \begin{cases} (rel(X) \cup \mathbf{0}) & \text{wenn } X \text{ an } \exists \text{ oder } \forall \text{ gebunden ist} \\ rel(X) & \text{sonst} \end{cases}$.

   Analog dazu sind die Ersetzung für $\vee_{(\theta_1,\theta_2,\ldots,\theta_m)}$, $\oplus^z_{(\theta_1,\theta_2,\ldots,\theta_m)}$ sowie für $\oplus^z$ definiert.

4. Für jede Negation $(\neg F_1(X_1, \ldots, X_m))$ in F wird der Algebraterm $(Dom_{D_1} \times \ldots \times Dom_{D_m}) \setminus \varphi(F_1)$ in $\varphi(F)$ durch $(rel(X_1) \times \ldots \times rel(X_m)) \setminus \varphi(F_1)$ ersetzt.

□

Unter Beachtung dieser Ersetzungsregeln kann die Semantik eines in die Ähnlichkeitsalgebra überführten $\mathcal{SDC}$-Ausdrucks angegeben werden.

Definition 5.5.3. Die *Semantik* eines $\mathcal{SDC}$-Anfrageausdrucks $E = \{X_1, X_2, \ldots, X_n | F(X_1, X_2, \ldots, X_n)\}$, welcher in einen $\mathcal{SA}$-Ausdruck überführt wird, ist:

$$\{(v_0, v_1, \ldots, v_n) | (v_0, v_1, \ldots, v_n) \in I^*(\varphi(F(X_1, X_2, \ldots, X_n))) \wedge v_0 > 0\}.$$

□

Das Potenzial des Ähnlichkeitskalküls und die Überführung eine $\mathcal{SDC}$-Ausdrucks in die Ähnlichkeitsalgebra wird im Folgenden an einem Beispiel mit Hilfe einer Stoffdatenbank demonstriert.

5.5.2 Beispiel

Der Stoffhersteller speichert die Informationen zu den verschiedenen Stoffen in der Relation *Stoff*. Zu jedem Stoff existieren verschiedene Abbildungen, beispielsweise unterschiedliche Fotografien eines Musters. Diese werden in der Tabelle *Muster* gespeichert. Auszüge aus diesen beiden Relationen sind in den nachfolgenden Tabellen dargestellt. Als ein ähnliches Szenario wäre die Suche nach ähnlichen, gerenderten Strickmustern (Chen u. a., 2003) denkbar.

Stoff

SId	Name	Qualitaet
S001	tartan_0815	hoch
S002	tartan_0816	mittel
S003	blue_stripes	hoch
...	...	...

Muster

BId	SId	Bild
B001	S001	tartan_0815_1.jpg
B002	S001	tartan_0815_2.jpg
B003	S002	tartan_0816_1.jpg
B004	S003	blue_stripes_1.jpg
...	...	...

Anhand der folgenden komplexen Multimedia-Anfrage soll die Überführung des entsprechenden Kalkülausdrucks in einen äquivalenten Algebraausdruck verdeutlicht werden:

> *Ermittle den Namen und die Qualität der Stoffe, die eine hohe Qualität haben oder dem gegeben Bild (C_{Bild}) in Farbe und Textur ähnlich sind. Die Verknüpfung dieser Bedingungen wird mit z=0.2 reglementiert. Die Farbähnlichkeit ist wichtiger als die Ähnlichkeit bezüglich der Textur (Gewicht von 0.9 versus 0.1). Ferner ist die Qualität eines Stoffes weniger wichtig als die Ähnlichkeit eines Stoffes zum vorgegebenen Stoffmuster (Gewicht von 0.3 versus 0.7). Letztlich sollen nur die Stoffe ermittelt werden, die einen Relevanzwert größer 0.7 aufweisen.*

Das verwendete Retrieval-System bietet eine Vielzahl verschiedener Ähnlichkeitsprädikate an. So können beispielsweise die Ähnlichkeitsoperatoren $\sim_{Farbe}$ und $\sim_{Textur}$ genutzt werden,

um die Ähnlichkeit bezüglich des Features Farbe beziehungsweise Textur zu bestimmen. Unter Beachtung dieser Gegebenheiten kann die obige natürlichsprachliche Anfrage als folgender $\mathcal{SDC}$-Ausdruck dargestellt werden:

$$\{X_N, X_Q \mid (\not\sim_{\theta_S}(\exists X_{SId}(\exists X_{BId}(\exists X_B((Stoff(X_{SId}, X_N, X_Q) \wedge Muster(X_{BId}, X_{SId}, X_B)) \wedge (((X_B \sim_{Farbe} C_{Bild}) \wedge_{(\theta_F,\theta_T)} (X_B \sim_{Textur} C_{Bild})) \oplus^z_{(\theta_\sim,\theta_Q)} (X_Q = C_{hoch}))))))\}$$

Die Formel $F(X_N, X_Q)$ ist als Baum in Abbildung 5.4 dargestellt. Aus Gründen der Übersichtlichkeit wird die Formel F in die folgenden Teilformeln zerlegt:

$$F(X_N, X_Q) := (\not\sim_{\theta_S} F_a(X_N, X_Q))$$
$$F_a(X_N, X_Q) := (\exists X_{SId} F_b(X_{SId}, X_N, X_Q))$$
$$F_b(X_{SId}, X_N, X_Q) := (\exists X_{BId} F_c(X_{SId}, X_N, X_Q, X_{BId}))$$
$$F_c(X_{SId}, X_N, X_Q, X_{BId}) := (\exists X_B F_1(X_{SId}, X_N, X_Q, X_{BId}, X_B))$$
$$F_1(X_{SId}, X_N, X_Q, X_{BId}, X_B) := (F_2(X_{SId}, X_N, X_Q, X_{BId}, X_B) \wedge F_3(X_Q, X_B))$$
$$F_2(X_{SId}, X_N, X_Q, X_{BId}, X_B) := (F_{21}(X_{SId}, X_N, X_Q) \wedge F_{22}(X_{BId}, X_{SId}, X_B))$$
$$F_{21}(X_{SId}, X_N, X_Q) := Stoff(X_{SId}, X_N, X_Q)$$
$$F_{22}(X_{BId}, X_{SId}, X_B) := Muster(X_{BId}, X_{SId}, X_B)$$
$$F_3(X_Q, X_B) := (F_5(X_B) \oplus^z_{(\theta_\sim,\theta_Q)} F_4(X_Q))$$
$$F_4(X_Q) := X_Q = C_{hoch}$$
$$F_5(X_B) := (F_{51}(X_B) \wedge_{(\theta_F,\theta_T)} F_{52}(X_B))$$
$$F_{51}(X_B) := X_B \sim_{Farbe} C_{Bild}$$
$$F_{52}(X_B) := X_B \sim_{Textur} C_{Bild}$$

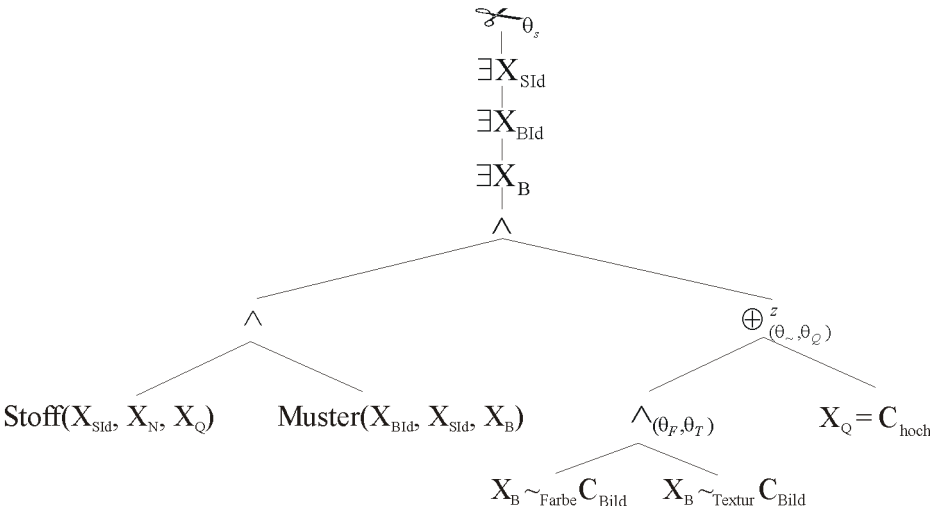


Abbildung 5.4: Anfragebaum des Kalkülausdrucks

Die *gen*-Relation gilt für die freien Variablen X_N und X_Q, beispielsweise gilt $gen(X_N, F, \pi_{\#2}(Stoff))$. Da auch die *con*-Relation für die gebundenen Variablen X_{SId}, X_{BId} und X_B gilt, ist die Anfrage evaluierbar. Der $\mathcal{SDC}$-Ausdruck kann daher in einen $\mathcal{SA}$-Ausdruck überführt werden. Dies wird auch hier aus Gründen der Übersichtlichkeit für die einzelnen Teilformeln dargestellt. Somit ergibt sich:

$$\varphi^*(F_{52}(X_B)) := \sigma_{\#1 \sim_{Textur} C_{Bild}}((\pi_{\#3}(Muster)) \cup 0)$$
$$\varphi^*(F_{51}(X_B)) := \sigma_{\#1 \sim_{Farbe} C_{Bild}}((\pi_{\#3}(Muster)) \cup 0)$$
$$\varphi^*(F_5(X_B)) := (\pi_{\#1}(\varphi^*(F_{51}(X_B)))) \wedge_{(\theta_F,\theta_T)} (\pi_{\#1}(\varphi^*(F_{52}(X_B))))$$
$$\varphi^*(F_4(X_Q)) := \sigma_{\#1=C_{hoch}}(\pi_{\#3}(Stoff))$$
$$\varphi^*(F_3(X_Q, X_B)) := (\pi_{\#2,\#1}((\varphi^*(F_5(X_B))) \times \pi_{\#3}(Stoff))) \uplus^z_{(\theta_\sim,\theta_Q)} (\pi_{\#1,\#2}((\varphi^*(F_4(X_Q))) \times ((\pi_{\#3}(Muster)) \cup 0)))$$
$$\varphi^*(F_{22}(X_{BId}, X_{SId}, X_B)) := \pi_{\#1,\#2,\#3}(\sigma(Muster))$$
$$\varphi^*(F_{21}(X_{SId}, X_N, X_Q)) := \pi_{\#1,\#2,\#3}(\sigma(Stoff))$$
$$\varphi^*(F_2(X_{SId}, X_N, X_Q, X_{BId}, X_B)) := \pi_{\#1,\#2,\#3,\#4,\#6}((\varphi^*(F_{21}(X_{SId}, X_N, X_Q))) \bowtie_{\#1=\#2} (\varphi^*(F_{22}(X_{BId}, X_{SId}, X_B))))$$
$$\varphi^*(F_1(X_{SId}, X_N, X_Q, X_{BId}, X_B)) := \pi_{\#1,\#2,\#3,\#4,\#5}(\varphi^*(F_2(X_{SId}, X_N, X_Q, X_{BId}, X_B)) \bowtie_{\#3=\#1,\#5=\#2} \varphi^*(F_3(X_Q, X_B)))$$
$$\varphi^*(F_c(X_{SId}, X_N, X_Q, X_{BId})) := \pi_{\#1,\#2,\#3,\#4}(\varphi^*(F_1(X_{SId}, X_N, X_Q, X_{BId}, X_B)))$$
$$\varphi^*(F_b(X_{SId}, X_N, X_Q)) := \pi_{\#1,\#2,\#3}(\varphi^*(F_c(X_{SId}, X_N, X_Q, X_{BId})))$$
$$\varphi^*(F_a(X_N, X_Q)) := \pi_{\#2,\#3}(\varphi^*(F_b(X_{SId}, X_N, X_Q)))$$
$$\varphi^*(F(X_N, X_Q)) := \rightsquigarrow_{\theta_S}(\varphi^*(F_a(X_N, X_Q)))$$

Dieser komplexe, generierte $\mathcal{SA}$-Ausdruck entspricht dem in Abbildung 5.5 dargestellten Anfragebaum. Wie hier bereits deutlich wird, besteht ein Optimierungspotenzial. Eine Optimierung kann sowohl während der Überführung eines Kalkülausdrucks in einen entsprechenden Algebraausdruck als auch im Anschluss an die Überführung erfolgen. Im Ausblick dieser Arbeit wird in Abschnitt 8.2.1 auf diesen Aspekt näher eingegangen und es wird eine vereinfachte Variante dieses Algebraausdrucks vorgestellt.

Für die gewichtete Kombination der einzelnen Relevanzwerte für ein Objekt wird die Faginsche Formel (siehe Abschnitt 4.2.1) eingesetzt. Als zu Grunde liegende Scoringfunktion wird für die Konjunktion die Funktion *min* und für die Disjunktion die Funktion *max* eingesetzt. Der Universaljunktor verwendet ebenfalls *min* und *max* als Basisfunktionen. Die unter Verwendung der Gewichte $I(\theta_F) = 0.9$ und $I(\theta_T) = 0.1$, $I(\theta_\sim) = 0.7$ und $I(\theta_Q) = 0.3$ sowie $I(z) = 0.2$ berechneten Ähnlichkeitswerte sind in der Tabelle *RSV* zusammengefasst.

RSV

BId	$\sim_{Farbe}$	$\sim_{Textur}$	$\wedge_{(\theta_F,\theta_T)}$	$\oplus^z_{\theta_{(\sim,\theta_Q)}}$
B001	0.7	0.2	0.6	0.792
B002	0.8	0.9	0.8	0.896
B003	0.9	0.4	0.8	0.704
B004	0.1	0.8	0.1	0.532
...	...	...	...	...

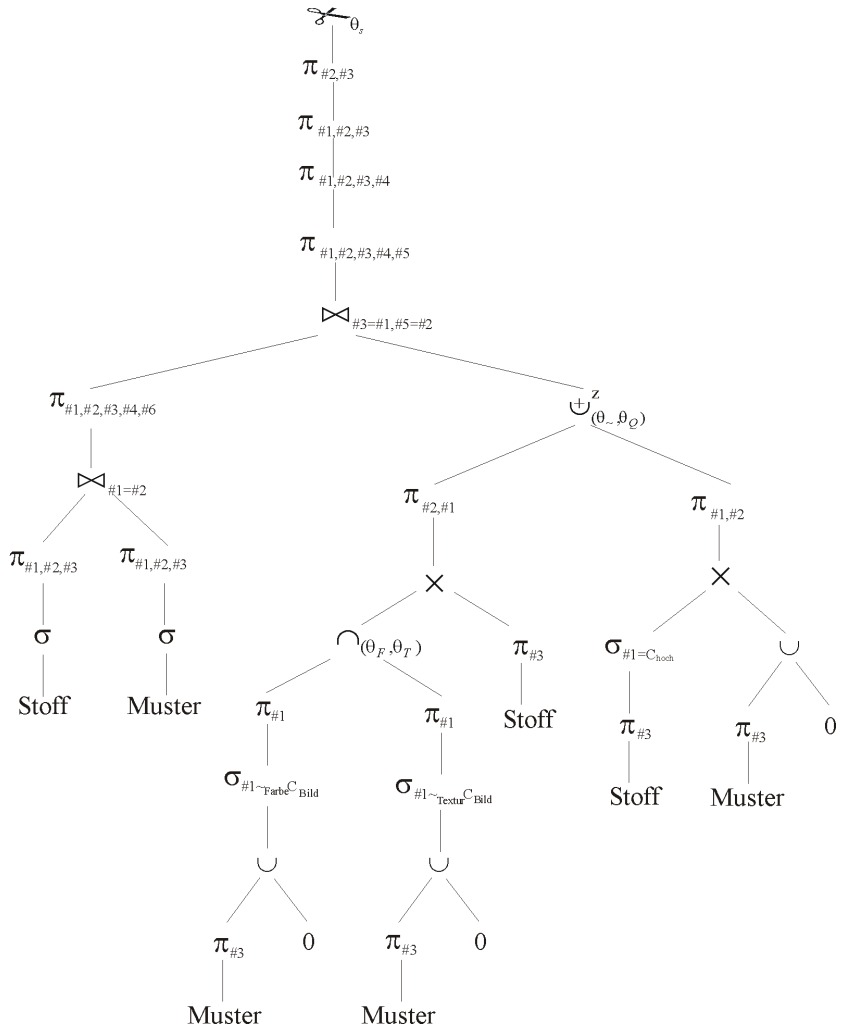


Abbildung 5.5: Anfragebaum des generierten Algebraausdrucks

Das Zwischenergebnis ist in der Tabelle *Zwischenergebnis* dargestellt. Es zeigt die Tupel mit den berechneten Relevanzwerten. Das letztendliche Anfrageergebnis ist in der Tabelle *Ergebnis* aufgeführt. Bei der Projektion werden die Duplikate eliminiert, wobei davon nur das Tupel mit dem höchsten Relevanzwert beibehalten wird. Das Tupel (0.792; tartan_0815; hoch) wird daher von dem Ergebnis entfernt. Bei Anwendung des $softcut$-Operators und unter Berücksichtigung des Schwellwertgewichts $I(\theta_S) = 0.7$ wird der Relevanzwert des Tupels (0.532; blue_stripes; hoch) auf 0 gesetzt. Damit wird dieses Tupel entsprechend Definition 5.5.3 nicht ausgegeben. Letztlich werden also nur die Tupel angezeigt, deren Relevanzwert über 0.7 liegen.

Zwischenergebnis

RSV	BId	SId	Bild	Name	Qualitaet
0.792	B001	S001	tartan_0815_1.jpg	tartan_0815	hoch
0.896	B002	S001	tartan_0815_2.jpg	tartan_0815	hoch
0.704	B003	S002	tartan_0816_1.jpg	tartan_0816	mittel
0.532	B004	S003	blue_stripes_1.jpg	blue_stripes	hoch
...	...	...	...	...	...

Ergebnis

RSV	Name	Qualitaet
0.896	tartan_0815	hoch
0.704	tartan_0816	mittel
...	...	...

Anhand dieses Beispiel wird deutlich, dass die Formulierung einer Anfrage in einer deklarativen Sprache (Ähnlichkeitskalkül) einfacher ist, als in einer prozeduralen Sprache (Ähnlichkeitsalgebra). Der Nutzer spezifiziert demnach nur, was gesucht ist und nicht, wie das Gesuchte ermittelt werden soll. Damit kann die Anfrageformulierung unabhängig von der internen Anfrageauswertung erfolgen. Ein Vergleich der in den Abbildungen 5.4 und 5.5 dargestellten Anfrageausdrücke zeigt, dass der Kalkülausdruck weniger komplex und leichter zu verstehen ist. Ein Teil der Komplexität des Algebraausdrucks entsteht zwar auf Grund der Generierung, aber dennoch ist der Kalkülausdruck wesentlich verständlicher.

5.6 Zusammenfassung

Dieses Kapitel bildet die formale Basis für die Entwicklung einer nutzerfreundlichen, deklarativen Multimedia-Anfragesprache. Da dafür das Bereichskalkül erweitert wurde, wurde zunächst eine Einführung in dieses Kalkül gegeben. Dem schloß sich eine kurze Diskussion bezüglich der Sicherheit von Anfragen an.

Den Schwerpunkt des Kapitels bildete die Vorstellung und Erweiterung eines Ähnlichkeitskalküls. In dieser Arbeit wurde das auf dem Bereichskalkül basierende Ähnlichkeitskalkül von Schmitt und Schulz (2004a,b) erweitert und modifiziert. Dieses Kalkül unterstützt sowohl diskrete als auch kontinuierliche Operationen sowie verschiedene Arten von Gewichten. Die Semantik der Operationen ist nicht vordefiniert. Das $\mathcal{SDC}$ bildet somit einen Framework, der es, je nach Anforderungen und Anwendungsgebiet erlaubt, die Semantik der Operationen zu definieren. Damit können durch das Ähnlichkeitskalkül sowohl zeitliche, räumliche, scharfe und unscharfe sowie inhaltsbasierte Multimedia-Anfragen unterstützt werden. Um das Multi-Level-Gewichtungsmodell adäquat umsetzen zu können, wurde zusätzlich der Universaljunktor eingeführt. Ferner wurden die gewichteten Junktoren derart modifiziert, dass sie als n-äre Operatoren eingesetzt werden können. Des Weiteren wurden Fuzzy-Relationen eingeführt. Das Ergebnis einer Anfrage ist somit eine Fuzzy-Relation, die wiederum als Eingabe für die nächste Anfrage verwendet werden kann.

Neben dem Ähnlichkeitskalkül wurde die Ähnlichkeitsalgebra vorgestellt. Die Erweiterungen des Ähnlichkeitskalküls wurden hier berücksichtigt. Diese Algebra bildet die Ziel-

sprache für eine Überführung eines $\mathcal{SDC}$-Ausdrucks. Es wurde sowohl die Syntax als auch die Semantik der Ähnlichkeitsalgebra definiert.

In einem weiteren Abschnitt wurde die Reduktion eines Ausdrucks des Ähnlichkeitskalküls in einen Ausdruck der Ähnlichkeitsalgebra beschrieben. An einem Beispiel wurde die Formulierung einer natürlichsprachlichen Anfrage im Ähnlichkeitskalkül sowie die Überführung der Anfrage in einen Ausdruck der Ähnlichkeitsalgebra demonstriert. Weiterhin wurden anhand konkreter Beispieldaten und der Festlegung einer konkreten Semantik für die Junktoren die Evaluierung der Anfrage veranschaulicht.

An diesem Beispiel wurden die Vorteile einer deklarativen Anfragesprache deutlich. Das Beispiel zeigte aber auch, dass die Formulierung von Anfragen im Ähnlichkeitskalkül für den Nutzer immer noch zu kompliziert ist. Damit kann der Entwurf einer neuen deklarativen Multimedia-Anfragesprache motiviert werden. Die Entwicklung dieser Sprache ist Gegenstand des folgenden Kapitels.

6 Die Anfragesprache WS-QBE

In diesem Kapitel wird der Entwurf der Multimedia-Anfragesprache *WS-QBE (Weighted-Similarity-QBE)* vorgestellt. Dies erfolgt unter Beachtung der in Abschnitt 3.2 beschriebenen Prinzipien für die Entwicklung von Multimedia-Anfragesprachen.

Eine wichtige Aufgabe von Multimedia-Anfragesprachen ist die adäquate Unterstützung des Nutzers bei der Formulierung seiner Suchbedingungen. So muss es beispielsweise möglich sein, eine gewünschte Farbe oder Textur als Anfragekriterium zu wählen. Häufig werden daher Query-By-Example-Anfragen formuliert. Dabei gibt der Nutzer Medien-Objekte vor, um zu beschreiben, welche Eigenschaften das zu suchende Objekt aufweisen muss. Für diese Aufgabe sind visuelle Anfragesprachen besonders gut geeignet. In diesem Kapitel wird eine Multimedia-Anfragesprache entwickelt, die auf dem Kern der Anfragesprache QBE aufsetzt und diese um Multimedia-Funktionalitäten erweitert.

Zunächst steht die Motivation der Entwicklung einer visuellen QBE-ähnlichen Multimedia-Anfragesprache im Vordergrund. Daraufhin erfolgt eine Beschreibung der Grundlagen von QBE. Den Kern des Kapitels bildet die Beschreibung der Multimedia-Anfragesprache WS-QBE. Dafür werden zunächst die grundlegenden Prinzipien der Sprache vorgestellt. Diese werden anhand von Beispielen veranschaulicht. Dann wird die formale Syntax der Sprache beschrieben. Die Semantik von WS-QBE wird durch die Überführung auf das gewichtete Ähnlichkeitskalkül festgelegt. Hierfür wird ein Überführungsalgorithmus entwickelt. Demonstriert wird dieser abschließend an einem Beispiel.

6.1 Motivation

Bei der Entwicklung von Anfragesprachen wird prinzipiell zwischen *visuellen* und *textbasierten* Sprachen unterschieden. Das Ziel visueller Anfragesprachen besteht darin, den interaktiven Umgang mit Datenbankanfragen zu vereinfachen. Dazu zählen vordergründig das Anbieten einer nutzerfreundlichen Mensch-Maschine-Interaktion, die Unterstützung nichttechnisch versierter Nutzer bei der Formulierung von Suchanfragen in Datenbanken sowie die Einführung von Mechanismen für eine komfortable Navigation durch die Daten auch dann, wenn die Anfragen unvollständig und unklar formuliert werden (Batini u. a., 1992). Damit liegen die Nachteile textbasierter Anfragesprachen klar auf der Hand: Zum einen erfolgt die Interaktion zwischen System und Nutzer nur auf Basis von Text. Zum anderen weisen die Operatoren eine komplexe Syntax auf (ebd.). Daher sind textuelle Sprachen zwar gut geeignet für einen Experten, jedoch kaum für einen nicht-technisch versierten Nutzer. Visuelle Anfragesprachen orientieren sich hingegen an einem weiten Spektrum von Nutzern. Dazu zählen sowohl Experten als auch Nutzer, die nur über wenig technisches Wissen und Können im Umgang mit Datenbanken verfügen.

Batini u. a. (1992) klassifizieren die visuellen Anfragesprachen in Form-, Diagramm- und Icon-basierte Sprachen sowie Hybride. Die Sprache QBE zählt zu den Form-basierten Spra-

chen. Sie unterstützt auf Grund der natürlichen Tendenz der Menschen, Daten in Tabellen, also in festen Strukturen, zu organisieren, insbesondere unversierte Nutzer bei der Formulierung von Anfragen (Catarci u. a., 1997). Verschiedene Studien belegen, dass die Nutzer bei QBE-ähnlichen Anfragen im Vergleich zu textuellen Anfragen weniger Fehler machen und die Anfragen einfacher und schneller formulieren können (Thomas und Gould, 1975; Greenblatt und Waxman, 1978; Reisner, 1988; Yen und Scamell, 1993). Dies gilt nicht nur für den unversierten Nutzer. Nach Catarci und Santucci (1995) können auch Experten von visuellen Anfragesprachen profitieren, denn so können typische Syntaxfehler oftmals vermieden werden.

Das Prinzip des Query-By-Examples ist im Rahmen des Multimedia-Retrievals etabliert. Darüber hinaus weisen visuelle Anfragesprachen gegenüber textbasierten Sprachen bezüglich der Nutzerfreundlichkeit entscheidene Vorteile auf. Aus diesen Gründen bietet sich die Entwicklung einer visuellen, QBE-basierten Multimedia-Anfragesprache an. Es ist somit möglich, eine komfortable, leicht zu bedienenden Schnittstelle für die Ad-hoc-Anfrageformulierung anzubieten. Dies gibt beispielsweise dem Nutzer zu Hause die Möglichkeit, seine Sammlung digitaler Bilder in einer Datenbank zu organisieren und darin nach ähnlichen Bildern zu suchen. Aber auch in dem Anwendungsszenario LostArt würde eine visuelle Anfragesprache den Mitarbeitern die Möglichkeit geben, eigenständig Anfragen an die Datenbank zu formulieren. Dies ist auf weitere Anwendungsgebiete, in denen Datenbank auch von unversierten Nutzern genutzt werden, übertragbar.

6.2 Query-By-Example

Die Anfragesprache Query-By-Example ist eine visuelle Anfragesprache mit minimaler Syntax und wurde von Zloof (1975, 1977) entwickelt. Sie bildet die Basis für den Entwurf der Multimedia-Anfragesprache WS-QBE.

Ein wesentlicher Unterschied von QBE zu anderen Anfragesprachen, wie beispielsweise SQL, besteht darin, dass der Nutzer nicht ausdrücklich eine strukturierte Anfrage formulieren muss. Vielmehr wird die Anfrage durch Ausfüllen von *Tabellengerüsten*, die auch als *Skelette* bezeichnet werden, spezifiziert. Der intensionale Teil der Relationen wird angezeigt und der extensionale Teil wird vom Nutzer ausgefüllt. In die Tabellenskelette werden *Beispielelemente* eingetragen. Sie dienen als Platzhalter für Datenbankwerte. Beispielelemente entsprechen somit *Variablen* und beginnen stets mit dem Zeichen '_'. Neben den Beispielelementen gibt es *Konstanten* und *Vergleichswerte*. Des Weiteren sind *Kontrolleinträge*, wie etwa **P.**, von Bedeutung. Der Eintrag **P.** zeigt an, welche Werte ausgegeben werden sollen.

QBE basiert auf dem Bereichskalkül. Die Semantik wurde von Zloof anhand von Beispielen definiert (Zloof, 1975, 1977). In den folgenden Abschnitten werden die Grundlagen von QBE sowie eine orthogonale Erweiterung, *SubQuery-By-Example (SQBE)*, vorgestellt.

6.2.1 Grundlagen von QBE

Eine QBE-Anfrage besteht aus einem oder mehreren Skeletten. Jedes Skelett entspricht einer Datenbankrelation und weist die folgende Struktur auf:

Relationenname	Attribut	...	Attribut
Tupelbefehl	Selektionsbedingung	...	Selektionsbedingung
⋮	⋮	...	⋮

wobei

- der *Relationenname* den Namen der relevanten Datenbankrelation enthält.
- ein *Attribut* ein aus der gewählten Relation gewünschtes Attribut bezeichnet.
- ein *Tupelbefehl* jene Befehle beinhaltet, die für ein ganzes Tupel gelten sollen. Neben den Kontrolleinträgen, wie beispielsweise **P.** für Print, kann auch das Negationszeichen $\neg$ auftreten. Die Negation einer Zeile bedeutet dann, dass es kein Tupel gibt, welches die spezifizierten Bedingungen erfüllt.
- die *Selektionsbedingungen* atomare Bedingungen beherbergen. Ein Tupel, das selektiert wird, muss die Bedingungen erfüllen. Neben Konstanten und Vergleichswerten können auch Beispielelemente angegeben werden. Vergleiche werden unter Verwendung der typischen Vergleichsoperatoren ($<, >, \leq, \geq, \neq, =$) gebildet.

  Einer Variablen wird der Wert eines Attributs zugeordnet, so dass sie beliebig oft verwendet werden kann. Durch die Verwendung von Variablen können mehrere Tabellen miteinander verknüpft werden.

  Auch das Voranstellen eines Kontrolleintrages ist möglich. Wird beispielsweise **P.** nicht als Tupelbefehl sondern nur für ein Attribut angegeben, so werden auch nur die Werte dieses Attributs ausgegeben.

Beispielanfragen

Im Folgenden wird kurz die Semantik von QBE durch einige Beispiele verdeutlicht. Dabei wird jede Anfrage in QBE formuliert und dazu der entsprechende Ausdruck im Bereichskalkül angegeben. Grundlage für die Anfragen bildet das Szenario einer Flugzeugbestellung mit den Relationen *Kunde(KName, Land, Ort)* und *Bestellung(BNummer, KName, FTyp, Anzahl)*. Für eine ausführliche Darstellung sei auf die einschlägige Literatur, wie beispielsweise Zloof (1975, 1977); Heuer und Saake (2000) sowie Ullman (1988), verwiesen.

1. Einfache Anfragen: *Welcher Kunde aus Deutschland hat eine Bestellung mit mehr als 3 Flugzeugen vom Typ Ultraleicht200C aufgegeben?*

Kunde	KName	Land	Ort
P.	_FlugvereinMD	Deutschland	

Bestellung	BNummer	KName	FTyp	Anzahl
		_FlugvereinMD	Ultraleicht200C	> 3

Diese einfache Anfrage setzt einen Verbund über zwei Relationen um. Um anzuzeigen, dass in beiden Relationen der gleiche Kunde gemeint ist, wird die Variable *_FlugvereinMD* als Beispielelement verwendet. Für die Sicherheit von Anfragen müssen Beispielelemente mindestens einmal in einer nicht negierten Zeile einer Relation auftreten. Ein äquivalenter Ausdruck

im Bereichskalkül ist:

$$\{X_K, X_L, X_O \mid \exists X_B \exists X_F \exists X_A Kunde(X_K, X_L, X_O) \wedge Bestellung(X_B, X_K, X_F, X_A) \\ \wedge X_A > 3 \wedge X_L = \text{'}Deutschland\text{'} \wedge X_F = \text{'}Ultraleicht200C\text{'}\}.$$

Bei Anfragen, die Ausgaben aus mehreren Tabellen umfassen, ist der Einsatz einer *temporären Ausgabetabelle* notwendig. In diesem Beispiel muss eine Ausgabetabelle dann eingesetzt werden, wenn neben den Kundeninformationen auch Information bezüglich der Bestellung ausgegeben werden sollen. Dabei kann die Struktur der Ausgabetabelle, also die Spaltennamen und die Anordnung der Spalten, vom Nutzer definiert werden.

2. Disjunktive Bedingungen: *Welche Bestellung enthält ein Flugzeug vom Typ Ultraleicht200C oder Doppeldecker3000?*

Bestellung	BNummer	KName	FTyp	Anzahl
	P. _111		Doppeldecker3000	
	P. _222		Ultraleicht200C	

Eine solche Disjunktion von Bedingungen kann durch Markierung mehrerer unterschiedlicher Variablen in der selben Spalte erfolgen. Alternativ kann die gleiche Anfrage auch mit Hilfe einer *Condition-Box* formuliert werden. Diese ermöglicht die Spezifikation von komplexen Suchbedingungen:

Bestellung	BNummer	KName	FTyp	Anzahl
	P.		_Flugzeug	

Condition-Box	Condition
	_Flugzeug = Doppeldecker3000 or _Flugzeug = Ultraleicht200C

Ein äquivalenter Ausdruck im Bereichskalkül ist:

$$\{X_B \mid \exists X_K \exists X_F \exists X_A Bestellung(X_B, X_K, X_F, X_A) \\ \wedge (X_F = \text{'}Doppeldecker3000\text{'} \vee X_F = \text{'}Ultraleicht200C\text{'})\}.$$

3. Selbstverbund: *Welche Bestellungen wurden für den selben Flugzeugtyp mit der gleichen Anzahl aufgegeben?*

Bestellung	BNummer	KName	FTyp	Anzahl
	P.		**P.** _Ultraleicht	**P.** _10
	P.		_Ultraleicht	_10

Analog zur disjunktiven Verknüpfung von Bedingungen in einer Relationen kann auch eine konjunktive Verknüpfung erfolgen. Ein Selbstverbund erfolgt durch Markierung gleicher Variablen in der selben Spalte. Mehrfachzeilen einer Relation entsprechen einer konjunktiven Verknüpfung der jeweiligen Bedingungen. Ein dazu äquivalenter Ausdruck im

Bereichskalkül ist:

$$\{X_B, X_{B'}, X_F, X_A \quad | \quad \exists X_K \exists X_{K'} Bestellung(X_B, X_K, X_F, X_A) \wedge Bestellung(X_{B'}, X_{K'}, X_F, X_A)\}.$$

4. Negation: *Welche Kunden haben nichts bestellt?*

Kunde	KName	Land	Ort
P.	¬FlugvereinMD		

Bestellung	BNummer	KName	FTyp	Anzahl
¬		¬FlugvereinMD		

Durch die Negation von Zeilen können all die Tupel vom Ergebnis ausgeschlossen werden, welche die in der negierten Zeile angegebenen Bedingungen erfüllen. Somit entspricht dies einem *not exists*. Um die Sicherheit von Anfragen zu gewährleisten, dürfen negierte Zeilen keine Ausgabeanforderung **P.** enthalten. Ein äquivalenter Ausdruck im Bereichskalkül ist:

$$\{X_K, X_L, X_O \quad | \quad Kunde(X_K, X_L, X_O) \wedge \neg(\exists X_B, \exists X_{K'}, \exists X_F, \exists X_A Bestellung(X_B, X_{K'}, X_F, X_A) \wedge X_{K'} = X_K)\}.$$

5. Funktionen, Sortierung und Aggregatfunktionen: Die bisher beschriebenen Sprachmittel bilden den QBE-Kern. Darüber hinaus unterstützt QBE die Verwendung von Funktionen, Aggregatfunktionen und eine Sortierung der Ausgabe.

In QBE stehen die fünf typischen Aggregatfunktionen **SUM.**, **AVG.**, **MIN.**, **MAX.** und **CNT.** zur Verfügung. Im Zusammenhang mit diesen Funktionen werden **G.**, **ALL.** und **UN.** verwendet. Mit **G.** können Mengen nach den Werten eines Attributs gruppiert werden. **ALL.** bezeichnet die Menge aller Attributwerte, wobei ein vorangestelltes **UN.** die Duplikate eliminiert.

Des Weiteren sind einfache arithmetische Funktionen in einer Anfrage zulässig. Ähnlich wie in SQL unterstützt auch QBE eine Sortierung in der Ausgabe. Mit **AO(n).** wird aufsteigend und dementsprechend mit **DO(n)**. absteigend sortiert. Die Priorität der Sortierung wird durch den Wert **n** ausgedrückt.

Formale Semantik von QBE

Die Semantik einer QBE-Kern-Anfrage ist formal durch die Transformation der Anfrage in einen entsprechenden Ausdruck des Bereichskalküls der Form $\{x_1, \ldots, x_n | F(x_1, \ldots, x_n)\}$ definiert. Dabei müssen disjunktive Bedingungen in der Condition-Box spezifiziert werden. Ferner darf, um die Sicherheit zu gewährleisten, in negierten Zeilen keine Ausgabeanforderung durch **P.** spezifiziert werden. Die Transformation einer QBE-Anfrage in das Bereichskalkül findet folgendermaßen statt:

1. Für alle Attribute A_i aller vorkommenden Anfragezeilen der Anfrage wird eine Bereichsvariable X_i erzeugt.

2. Die Variablen $_Var$ der Ausgabetabelle werden zu den freien Variablen geschrieben: $(\{\ldots, _Var, \ldots \mid \ldots\})$.

3. Für alle positiven Anfragezeilen gilt:

   a) Steht in der Anfragezeile mit $A_l, \ldots, A_m$ der Tupelbefehl **P.**, dann werden alle X_j mit $j = l, \ldots m$ zu den freien Variablen der Ausgabe geschrieben: $\{\ldots, X_l, \ldots, X_m, \ldots \mid \ldots\}$.

   b) Steht bei A_i das Kommando **P.**, dann wird X_i zu den freien Variablen geschrieben: $\{\ldots, X_i, \ldots \mid \ldots\}$.

   c) Für jede Relation R wird unter Verwendung der jeweiligen Bereichsvariablen der Anfragezeile ein Atom $R(X_i, X_{i+1}, \ldots)$ erzeugt. Diese bilden konjunktiv miteinander verknüpft die Formel F.

   d) Steht bei A_i eine Konstante *Const* beziehungsweise ein Vergleichswert der Form $\delta Const$ mit $\delta \in \{<, >, \leq, \geq\}$, dann wird $X_i = Const$ beziehungsweise $X_i \delta Const$ mit $\wedge$ an F angefügt.

   e) Dies gilt auch für die Variablen $_Var$. Demnach wird $X_i = _Var$ beziehungsweise $X_i \delta _Var$ mit $\delta \in \{<, >, \leq, \geq\}$ mit $\wedge$ an F angefügt.

4. Für alle negierten Anfragezeilen gilt:

   a) Für jede Relation R wird unter Verwendung der jeweiligen Bereichsvariablen der Anfragezeile ein Atom $R(X_i, X_{i+1}, \ldots)$ erzeugt. Diese Atome werden konjunktiv zu der Teilformel F_{neg} verknüpft.

   b) Steht bei A_i eine Konstante *Const* beziehungsweise ein Vergleichswert der Form $\delta Const$ mit $\delta \in \{<, >, \leq, \geq\}$, dann wird $X_i = Const$ beziehungsweise $X_i \delta Const$ mit $\wedge$ an F_{neg} angefügt.

   c) Dies gilt auch für die Variablen $_Var$. Demnach wird $X_i = _Var$ beziehungsweise $X_i \delta _Var$ mit $\delta \in \{<, >, \leq, \geq\}$ mit $\wedge$ an F_{neg} gehängt.

   d) Jede Variable X_i der Anfragezeile wird mit $\exists$ in F_{neg} gebunden: $\{\ldots \mid \ldots (\exists X_i(F_{neg}))\}$.

   e) $\neg F_{neg}$ wird mit $\wedge$ an F angefügt.

5. Der Inhalt der Condition-Box wird mit $\wedge$ an F angefügt.

6. Alle in einer positiven Anfragezeile auftretenden Variablen X_i und $_Var$, die nicht zu den freien Variablen gehören, werden mit $\exists$ in F gebunden: $\{\ldots \mid \ldots (\exists X_i(F))\}$ beziehungsweise $\{\ldots \mid \ldots (\exists _Var(F))\}$.

7. Zuletzt findet eine Vereinfachung der Formel F statt.

Damit entspricht eine QBE-Anfrage, bestehend aus positiven und negierten Zeilen sowie Einträgen in der Condition-Box dem folgenden Bereichskalkülausdruck (Heuer und Saake, 2000, S. 397f):

$$\{X_1, \ldots, X_m \quad | \quad \exists Y_1 \ldots \exists Y_n \bigwedge_i \langle \text{i-te positive Zeile} \rangle$$
$$\wedge \bigwedge_j \neg [\exists Z_{j_1} \ldots \exists Z_{j_p} \langle \text{j-te negierte Zeile} \rangle]$$
$$\wedge \bigwedge_k \langle \text{k-te Bedingung in Condition-Box} \rangle \}$$

wobei $X_1 \ldots X_m$ den mit **P.** markierten Variablen entspricht. $Y_1 \ldots Y_n$ sind alle restlichen Variablen in den positiven Zeilen und $Z_{j_1} \ldots Z_{j_p}$ sind die restlichen, impliziten Variablen in der j-ten negierten Zeile.

6.2.2 SubQuery-By-Example

Die visuelle Anfragesprache *SubQuery-By-Example (SQBE)* erweitert den Kern von QBE um parametrisierbare Unteranfragen sowie um Mengen und Aggregationsoperationen (Scharnofske, 1996; Scharnofske u. a., 1997). Dadurch ist SQBE streng relational vollständig. Damit wird ein wesentlicher Nachteil von QBE – die eingeschränkte Ausdrucksfähigkeit – ausgeräumt. Ein ähnlicher Ansatz wurde bereits von Özsoyoglu u. a. für statistische Datenbanken vorgeschlagen (Özsoyoglu und Wang, 1989; Özsoyoglu u. a., 1989).

Die formale Grundlage der Semantik von SQBE bildet, anders als bei QBE, ein erweitertes Tupelkalkül. Dieses erweiterte Tupelkalkül basiert auf dem klassischen Tupelkalkül und wird um die Verarbeitung von Multimengen sowie Möglichkeiten zur Aggregation und Gruppierung angereichert. Daher entspricht das erweiterte Tupelkalkül im wesentlichen dem auf Tupel und Relationen eingeschränkten Teil des EER-Kalküls (Hohenstein und Gogolla, 1988; Gogolla und Hohenstein, 1991; Hohenstein, 1993; Gogolla, 1994).

Ebenso wie in QBE werden Anfragen durch das Ausfüllen von Tabellengerüsten formuliert. Eine SQBE-Anfrage setzt sich aus verschiedenen *Fenstern*, die jeweils eine Unteranfrage darstellen, zusammen. Ein Fenster beinhaltet eine beliebige Anzahl von Anfragetabellen, eine Ausgabetabelle (**OUTPUT**) und optional eine Condition-Box. Es wird durch einen Namen eindeutig identifiziert. Demnach entspricht jedes Fenster genau einer QBE-Anfrage. Ein wichtiges Element der Fenster sind die Parameter. Die erste Anfrage wird mit **ROOT** bezeichnet und besitzt keine Parameter.

Eine Anfragetabelle basiert auf Tabellen. Dazu zählen Relationen, Fensteraufrufe oder ein mittels einer (Multi-)Mengenoperation zusammengesetzter (Multi-)Mengenterm von Tabellen. Bei einem Fensteraufruf wird das Ergebnis einer Unteranfrage unter Beachtung der formalen Parameter bestimmt. Abbildung 6.1 verdeutlicht den Aufbau einer SQBE-Anfrage anhand der Formulierung der Anfrage aus Beispiel 6.1.

Beispiel 6.1 *Welche Paare von Lieferanten (Namen) bieten genau die gleichen Waren an? (Scharnofske u. a., 1997)*

Die Anfrage wird mit Hilfe der Unteranfrage **WAREN**(**lname**) umgesetzt. Diese berechnet zu einem Lieferanten alle Waren, die dieser liefert. Dafür wird der Unteranfrage der Parameter **lname** übergeben. Liefern zwei unterschiedliche Lieferanten ($lname1 \neq lname2$), die

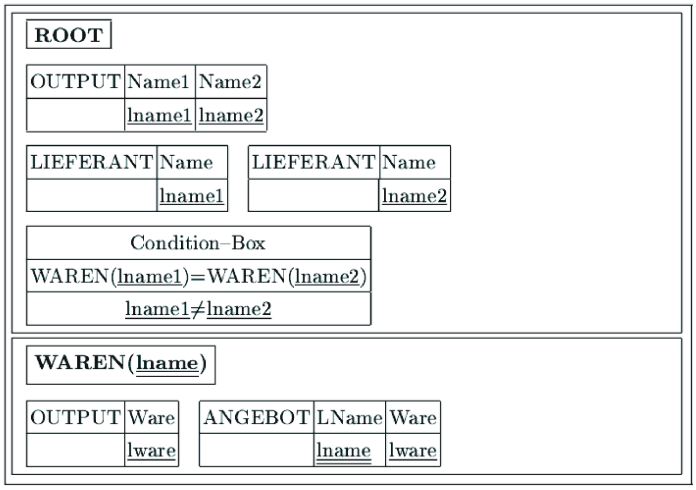


Abbildung 6.1: Anfrageformulierung in SQBE (Scharnofske u. a., 1997)

gleichen Waren ($WAREN(lname1) = WAREN(lname2)$), werden diese im **ROOT**-Fenster ausgegeben.

Die in SQBE formulierten Anfragen werden auf das erweiterte Tupelkalkül abgebildet. Dieses stellt jedoch nur einen Zwischensprache für die eindeutige Definition der Semantik von SQBE-Anfragen dar. Für die Umsetzung von SQBE werden daher alle Ausdrücke des erweiterten Tupelkalküls nach SQL übersetzt. Nach Scharnofske u. a. kann die Abbildung der SQBE-Anfragen in SQL relativ einfach mittels Unteranfragen oder durch Sichten beziehungsweise temporäre Tabellen erfolgen.

6.2.3 Zusammenfassung

Ein wesentliches Merkmal der Sprache QBE ist, dass Anfragen durch Vorgabe von Beispielelementen in Tabellengerüsten formuliert werden. QBE selbst ist nur für Anfragen auf traditionellen relationalen Datenbanken geeignet. Es erlaubt somit nur numerische und alphanumerische Werte als Beispielelemente. Dieses Prinzip bietet jedoch eine gute Grundlage für die Entwicklung einer Multimedia-Anfragesprache. Insbesondere bei der Formulierung von inhaltsbasierten Anfragen ist der Einsatz von Beispielelementen hilfreich.

Die Vorteile von QBE liegen in der Einfachheit und schnellen Erlernbarkeit der Sprache. Dies ergibt sich aus der kleinen Menge an bereitstehenden Sprachmitteln. Verschiedene Studien haben die Vor- und Nachteile von QBE untersucht, indem die Unterschiede zwischen QBE und anderen Anfragesprachen analysiert wurden (Thomas und Gould, 1975; Greenblatt und Waxman, 1978; Reisner, 1988; Yen und Scamell, 1993). Als Vergleichssprachen wurden unter anderem die Structured English Query Language (SEQUEL), SQL sowie eine algebrabasierte Sprache verwendet. Das Fazit dieser Untersuchungen war, dass die Nutzer bei QBE-ähnlichen Anfragen weniger Fehler machen und die Anfragen in QBE einfacher und schneller formulieren können als in einer textbasierten Sprache. Ein wichtiger Vorteil von QBE besteht ferner darin, dass die Zufriedenheit der Nutzer bei QBE im Vergleich etwa zu SQL erheblich größer ist (Yen und Scamell, 1993).

Nachteilig ist jedoch, dass der Nutzer über ein gewisses Maß an Abstraktionsvermögen verfügen muss, um eine natürlichsprachliche Anfrage in QBE zu formulieren. Des Weiteren ist es für das Verständnis von QBE-Anfragen erforderlich, die Semantik von QBE zu kennen (Thomas und Gould, 1975). Ferner ist QBE eingeschränkt ausdrucksfähig. So ist der QBE-Kern nicht streng relational vollständig, sondern nur relational vollständig (Heuer und Saake, 2000, S. 398).

Eine Schachtelung und Verarbeitung von (Multi-)Mengen, so dass eine strenge relationale Vollständigkeit erreicht wird, bietet SQBE. Diese Anfragesprache ist eine orthogonale Erweiterung von QBE. Jedes Fenster in SQBE stellt wieder eine eigenständige QBE-Anfrage dar.

Aus diesem Grund ist es zunächst wichtig, QBE um Konzepte zur Handhabung von Präferenzen und um Sprachkonstrukte zur Formulierung von Multimedia-Anfragen anzureichern. Daher wird die Multimedia-Anfragesprache WS-QBE entwickelt. Sie wird in den folgenden Abschnitten vorgestellt. Darauf aufbauend können die Konzepte von SQBE in die Sprache WS-QBE einfließen, um so eine Schachtelung von Anfragen zu unterstützen. Die Kombination von WS-QBE und SQBE ist jedoch nicht Gegenstand dieser Arbeit.

6.3 Grundlegende Prinzipien

Die Anfragesprache WS-QBE erlaubt mit ihren Sprachmitteln eine Umsetzung des Multi-Level-Gewichtungsmodells. Es wird die Idee von QBE aufgegriffen. Somit werden auch in WS-QBE Anfragen durch das Ausfüllen von Tabellenskeletten formuliert. Abbildung 6.2 gibt einen Überblick über die wesentlichen Sprachkonstrukte, die in WS-QBE für die Anfrageformulierung zur Verfügung stehen. Dazu zählen die bekannten Tabellengerüste für Relationen. Durch die *Weighting*-Tabelle wird eine Relevanzgewichtung unterstützt. Die Formulierung komplexer Anfragen erfolgt in dem *Condition-Tree*. Ausgabebedingungen können in der *Temporal-* und *Spatial-Condition-Box* beziehungsweise im *Spatial-Condition-Frame* spezifiziert werden. Anzumerken ist, dass bei Anfragen, die Ausgaben aus mehreren Tabellen umfassen, der Einsatz einer temporären Ausgabetabelle notwendig ist. Eine solche Tabelle entspricht einer Relation, wobei der Relationenname sowie die Spaltennamen beliebig benannt werden können. Insgesamt erweitert WS-QBE den QBE-Kern und zeichnet sich durch die Möglichkeiten

1. zur Formulierung von inhaltsbasierten, unscharfen, zeitlichen und räumlichen Anfragen,
2. zur Gewichtung von Anfragetermen und
3. zur Parametrisierung der Ausgabe

aus.

In den folgenden Abschnitten werden die grundlegenden Prinzipien von WS-QBE dargelegt. Es wird dabei näher auf die bereitgestellten Sprachkonstrukte, Operationen und Junktoren eingegangen. Darüber hinaus wird die Formulierung von Anfragen skizziert. Die Formulierung von Nutzerpräferenzen bildet dabei den Schwerpunkt.

Es ist anzumerken, dass die Anfragesprache WS-QBE weniger mächtig ist wie das in Kapitel 5 beschriebene Ähnlichkeitskalkül. Das primäre Ziel bei der Entwicklung von WS-QBE ist vielmehr die Bereitstellung einer Kernfunktionalität, um den Nutzer nicht mit zu vielen

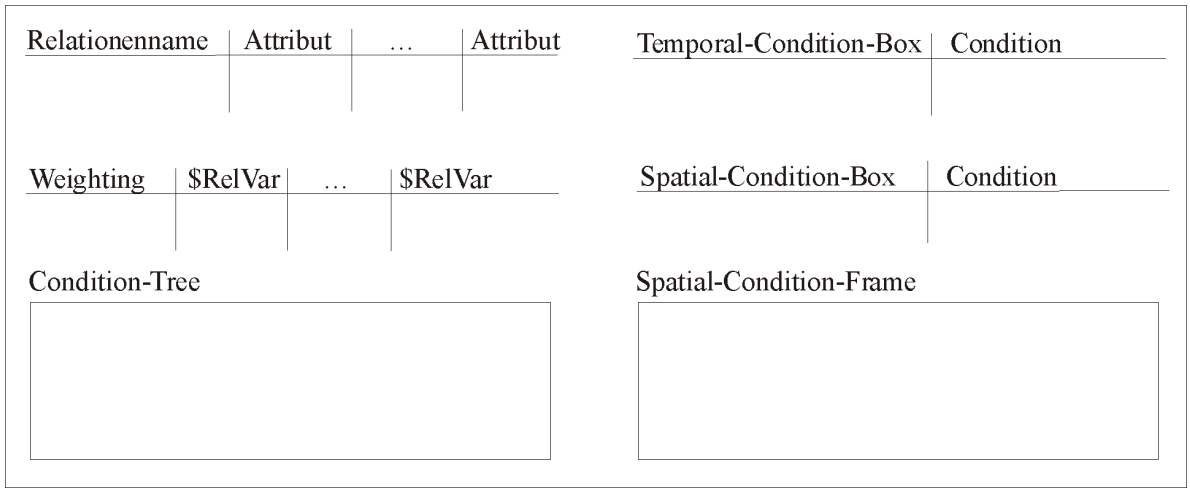


Abbildung 6.2: Wesentliche Sprachkonstrukte in WS-QBE

Sprachkonstrukten zu überfordern. So wird in WS-QBE auf die Quantoren $\exists_k$ und $\forall_k$ sowie auf die Schwellwertoperatoren $\succ\!\!\!\sim_\theta$ und $\succ\!\!\!\sim_{\theta,f}$ verzichtet. Inbesondere die parametisierbaren Quantoren sind für den unversierten Nutzer zu komplex. Für die Anfragesprache WS-QBE steht damit noch genügend Potenial für eine Erweiterung bereit. Die Erweiterung sind einfach möglich.

Anhand eines einfachen Stoffhersteller-Szenarios wird der Einsatz der vorgestellten Sprachmittel durch Beispiele unterlegt[1]. In der Relation *Stoff(SId, Name, Qualitaet, Video, Werbefoto, Audio)* werden die Informationen, wie Name und Qualität, zu den einzelnen Stoffen des Herstellers gespeichert. Zusätzlich dazu wird ein Verkaufsvideo, ein offizielles Foto für die Werbung sowie ein beschreibendes Audio-File in dieser Relation verwaltet. Da es zu jedem Stoff verschiedene Aufnahmen von Stoffproben gibt, werden diese Bilder gesondert in der Relation *Muster(BId, SId, Bild)* gespeichert.

6.3.1 Operationen

Für die Unterstützung von Multimedia-Anfragen wird die Menge der gültigen Operatoren erweitert. Neben den klassischen Vergleichsoperatoren ($<, >, =, \neq, \geq, \leq$) auf numerischen und alphanumerischen Daten stehen für

- ähnlichkeitsbasierte Anfragen die Operatoren $\sim$, $\sim_{Farbe}$, $\sim_{Form}$, $\sim_{Textur}$,
- für zeitliche Anfragen die Operatoren nach Allen (1983) $<, :, |, \%, [,], /$
- und für räumliche Anfragen die Operatoren *in*, *überlappt*, *überdeckt*, *gleich*, *sehr_nah*, *nah*, *fern*, *sehr_fern*, *links_von*, *rechts_von*, *oben_von* sowie *unten_von*

[1] Die in den Beispielen verwendeten Stoffmuster wurden der Webseite: http://www.oldhickoryfurniture.com/ entnommen.

zur Verfügung. Diese arbeiten auf Multimedia-Daten. Darüber hinaus wird der Ähnlichkeitsoperator $\sim$ auch für numerische und alphanumerische Daten angeboten, da auch beispielsweise Vergleiche bezüglich der Ähnlichkeit zweier Zeichenketten unterstützt werden sollen. Die Operatoren für ein ähnlichkeitsbasiertes Retrieval können je nach Anwendungsgebiet und zur Verfügung stehender Features beliebig erweitert werden.

Für die Formulierung zeitlicher Anfragen stehen die bekannten temporalen Relationen zur Verfügung. Darüber hinaus erlauben die räumlichen Prädikate eine Beschreibung von Distanz- und Richtungsbeziehungen sowie von topologischen Beziehungen zwischen den einzelnen Medien-Objekten. Zeitliche und räumliche Ausgabebedingungen können unter Verwendung der folgenden Operatoren spezifiziert werden: $<, :, |, \%, [,]$ sowie $/$.

6.3.2 Junktoren

Neben den klassischen Junktoren $\wedge$, $\vee$ kann bei der Formulierung komplexer Anfragen der Universaljunktor $\oplus^z$ eingesetzt werden. Durch eine Parametrisierung hat der Nutzer die Freiheit, das Verhalten des Operators zwischen *UND* und *ODER* zu verschieben. Soll sich der Universaljunktor wie ein *UND* verhalten, dann wird der Parameter z auf *eins* gesetzt. Soll er sich jedoch wie ein *ODER* verhalten, dann wird der Operator mit *null* parametrisiert. Ein z-Wert zwischen *null* und *eins* erlaubt hingegen eine unscharfe Verknüpfung.

6.3.3 Formulierung von Anfragen

Die Anfragen in WS-QBE werden, ähnlich wie in QBE, durch Ausfüllen von Tabellengerüsten formuliert. Es werden *Beispielelemente*, die als Platzhalter für Datenbankwerte dienen, in die Skelette eingetragen. Beispielelemente entsprechen Variablen. Analog zu QBE beginnen Variablen mit dem Zeichen '_'. Daneben gibt es *Konstanten* und *Vergleichswerte*. Im Unterschied zu QBE sind nicht nur numerische und alphanumerische Daten, sondern auch Multimedia-Daten als Konstanten zulässig.

Auch die Vergleichswerte beziehen sich nicht mehr nur auf Vergleiche zwischen numerischen und alphanumerischen Daten, sondern es werden zusätzlich dazu inhaltsbasierte, zeitliche und räumliche Vergleiche ermöglicht. Dies erfolgt unter Verwendung der zuvor vorgestellten Operatoren, die in WS-QBE zulässig sind.

Des Weiteren ist der *Kontrolleintrag* **P.** mit den entsprechenden Optionen, auf die im folgenden Abschnitt noch eingegangen wird, bei der Anfrageformulierung zulässig. Wie auch in QBE gibt dieser Kontrolleintrag an, welche Variablen zu den Ergebnisvariablen zählen.

Einfache Anfragen Für einfache Anfragen wird eine Relevanzgewichtung der einzelnen Selektionsbedingungen mittels *Relevanzvariablen* angeboten. Eine Relevanzvariable beginnt mit dem Zeichen '$'. In der Gewichtungstabelle *Weighting* wird den einzelnen Relevanzvariablen ein entsprechender Wert zugeordnet. Dies kann zum einen direkt durch Angabe eines numerischen Wertes aus dem Intervall [0,1] oder visuell erfolgen. Nach Fagin (1998) können für diesen Zweck Schieberegler gut verwendet werden. Daneben ist der Einsatz von linguistischen Variablen ebenso denkbar. Die Gewichtung einer einfachen Anfrage ist in den folgenden beiden Beispielen dargestellt.

Beispiel 6.2 *Welches Muster gehört zu dem Bild mit der Nummer I002 und ist dem vorgegebenen Bild ähnlich. Dabei ist die Ähnlichkeit bezüglich des vorgegebenen Bildes wichtiger als die Übereinstimmung mit der Nummer I002.*

Muster	BId	SId		Bild	
P.		I002	$S	~	$B

Weighting	$B	$S
	0.7	0.3

Beispiel 6.3 *Welcher Tartan mit hoher Qualität ist dem vorgegebenen Bild ähnlich? Dabei ist die Ähnlichkeit bezüglich des vorgegebenen Bildes am wichtigsten. Dem folgt das Kriterium bezüglich der Qualität. Die Ähnlichkeit mit dem gegebenen Namen hat die geringste Relevanz.*

Stoff	SId	Name	Qualitaet	Video	Werbefoto	Audio
P.	_I001	~ Tartan $N	hoch $Q			

Muster	BId	SId	Bild	
		_I001	~	$B

Weighting	$B	$Q	$N
	0.6	0.3	0.1

Neben der Relevanzgewichtung von Anfragetermen wird eine Schwellwertgewichtung unterstützt. Bei einfachen Anfragen können Schwellwerte direkt einzelnen Selektionsbedingungen oder einer ganzen Anfrage zugeordnet werden. Die Schwellwertgewichte können somit als Teil der Selektionsbedingung aufgefasst werden. Eine Schwellwertbedingung beginnt mit dem Zeichen '>' und wird nach einer Selektionsbedingung spezifiziert. Ist ein Schwellwertgewicht für eine Gesamtanfrage gültig, dann wird dies in der ersten Spalte der entsprechenden Ausgabetabelle vermerkt. Das folgende Beispiel verdeutlicht den Einsatz von Schwellwertgewichten.

Beispiel 6.4 *Welches Muster gehört zu dem Stoff I002 und ist dem vorgegebenen Bild ähnlich? Die Ähnlichkeit bezüglich des vorgegebenen Bildes soll mindestens 0.4 betragen. Ferner sollen nur die Informationen zu den Mustern ausgegeben werden, welche die Anfrage insgesamt gut (Mindestähnlichkeit von 0.7) erfüllen.*

Muster >0.7	BId	SId	Bild	
P.		I002	~	>0.4

Komplexe Anfragen Neben der Gewichtung und Formulierung von unscharfen, zeitlichen, räumlichen und inhaltsbasierten Anfragen besteht ein weiterer wesentlicher Unterschied zu QBE in der Formulierung komplexer Anfragen. Komplexe Bedingungen werden in WS-QBE anstelle in einer Condition-Box durch einen *Condition-Tree* realisiert. Soffer u. a.

(1998) haben bereits für Bildanfragen gezeigt, dass Suchkriterien gut in Form einer Baumstruktur spezifiziert werden können. In dem Condition-Tree können Bedingungen bezüglich der Variablen und Konstanten beliebig hierarchisch kombiniert werden. Zur Formulierung dieser stehen sämtliche Operationen für zeitliche, räumliche, scharfe und unscharfe Anfragen bereit.

Komplexe Bedingungen werden durch die Verknüpfung der einzelnen Prädikate mittels der Junktoren erstellt. Für die Formulierung komplexer Anfragen können nicht nur die bekannten klassischen Junktoren $\wedge$ und $\vee$ verwendet werden, sondern auch der Universaljunktor. Dieser wird zunächst als *ODER*-Operator initialisiert. Das Verhalten dieses Junktors kann mittels eines Schiebereglers auf der Bedienoberfläche durch den Nutzer parametrisiert werden. Für das Gewicht wird ein numerischer Wert abgeleitet und an den Operator geschrieben.

Die einzelnen Operanden, also die atomaren Selektionsbedingungen beziehungsweise die Teilanfragen, werden mittels gewichtbarer Verbinder verknüpft. Die Stärke der Verbinder visualisiert dabei die Relevanz des Operanden. Je dicker ein Verbinder dargestellt ist, desto wichtiger ist der entsprechende Operand. Die Relevanzgewichtung der Operanden ist optional. Initial sind aus diesem Grund alle Operanden gleich stark gewichtet. Dieses entspricht einer ungewichteten Anfrage. Abbildung 6.3 zeigt einen solchen Bedingungsbaum mit unterschiedlich stark gewichteten Operanden für die Anfrage aus dem folgenden Beispiel.

Beispiel 6.5 *Welches Muster zu einem Stoff, dessen Name zu 'Glacier' ähnlich ist, weist die vorgegebenen Farb-, Textur- und Formmerkmale auf? Dabei ist die Ähnlichkeit bezüglich der Farbe am wichtigsten. Dem folgt die Ähnlichkeit bezüglich der Textur, gefolgt von der Ähnlichkeit bezüglich der Form. Darüber hinaus ist das Kriterium bezüglich des Namens wichtiger als die Ähnlichkeitsbedingungen.*

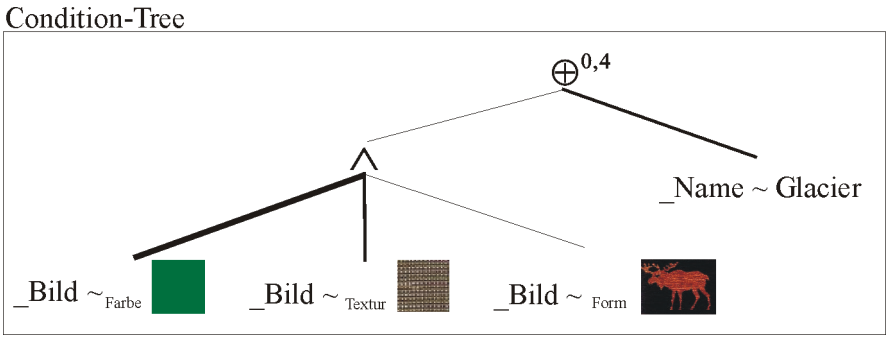
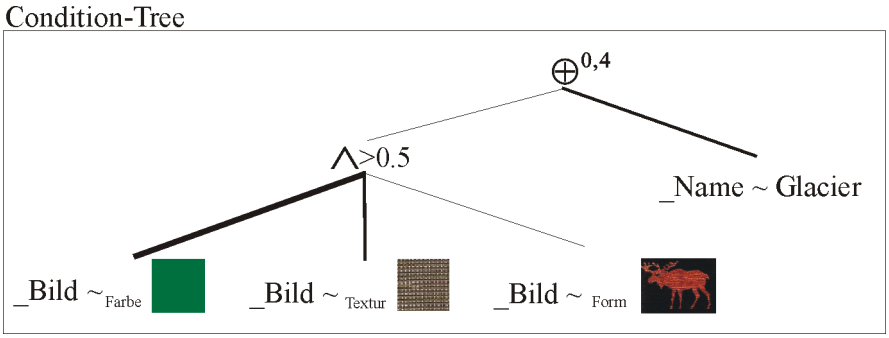


Abbildung 6.3: Darstellung eines Condition-Tree

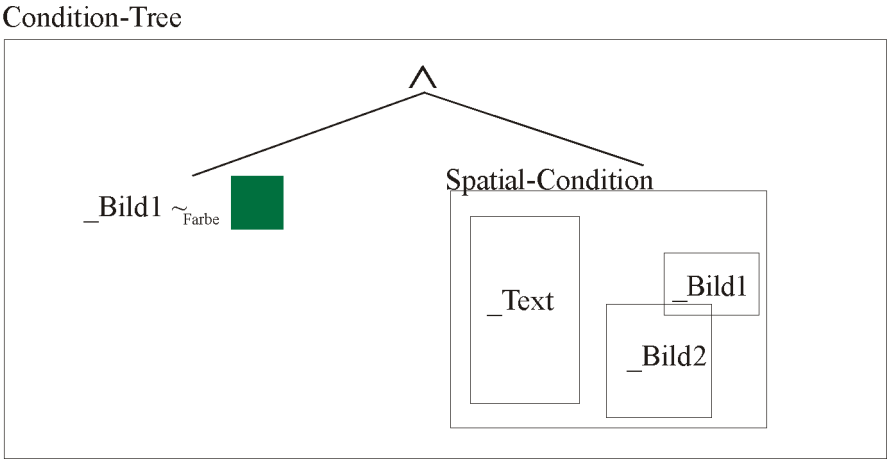
Neben der Relevanzgewichtung der Anfrageterme ist auch hier eine Schwellwertgewichtung zulässig. Analog zur Spezifizierung von Schwellwertgewichten bei einfachen Anfragen, werden sie auch bei komplexen Anfragen als zusätzliche Selektionsbedingung aufgefasst und können somit direkt angegeben werden. Dies ist im folgenden Beispiel dargestellt. Hierbei wird ein Schwellwertgewicht für eine Teilanfrage spezifiziert.

Beispiel 6.6 *Bezugnehmend auf die Anfrage aus dem vorhergehenden Beispiel 6.5 sollen nun die Muster ermittelt werden, die bezüglich der Bildähnlichkeit mindestens einen Relevanzwert von 0.5 aufweisen.*



Insbesondere bei räumlichen Anfragen sind oftmals nicht nur die Beziehungen zwischen zwei Objekten, sondern meist zwischen mehreren Objekten von Bedeutung. Daher entstehen häufig komplexe räumliche Bedingungen. Um die Spezifikation dieser adäquat unterstützen zu können, bietet der Condition-Tree die Möglichkeit, Beispielelemente räumlich anzuordnen. Wesentlich dabei ist, dass nicht nur zwei Elemente in Beziehung gesetzt werden können, sondern beliebig viele. Um die verschiedenen räumlichen Beziehungen ausdrücken zu können, werden den Beispielelementen Geometrieobjekte zugewiesen. Damit entsteht eine Spatial-Query-By-Sketch (Egenhofer, 1997). Aus dieser Skizze werden Richtungsbeziehungen und topologische Beziehungen extrahiert. Das folgende Beispiel zeigt die Formulierung einer räumlichen Bedingung im Condition-Tree.

Beispiel 6.7 *Es sollen die Objekte ermittelt werden, deren Text sich links von zwei Bildern befindet. Die Bilder sollen sich wiederum überlappen. Ferner soll Bild1 zu einer vorgegebenen Farbe ähnlich sein.*



6.3.4 Kontrolleinträge

Da WS-QBE zunächst nur als eine reine Anfragesprache entwickelt wird, ist nur der Kontrolleintrag **P.** für die Markierung einer auszugebenden Zeile oder Spalte zulässig. Um den

Anforderungen einer Multimedia-Anfragesprache gerecht zu werden, muß die Ausgabe der Medien-Objekte parametrisiert werden können. Für diesen Zweck wird der Kontrolleintrag mit weiteren Optionen ausgestattet:

- *Zeitdauer*; gibt die Dauer der Präsentation eines Medienobjektes in Sekunden an. Wird keine Zeitdauer angegeben, so wird die ursprüngliche Dauer eines kontinuierlichen Mediums genutzt. Die Präsentation diskreter Medien-Objekte erfolgt in diesem Fall für die Dauer eines spezifizierten Default-Zeitraumes.
- *Geschwindigkeit*; gibt bei kontinuierlichen Medien an, wie schnell (normal, schnell, langsam) das Medien-Objekt abgespielt werden soll. Bei zeitunabhängigen Medien gilt diese Option nicht.
- *Größe*; legt die Höhe und Breite (Höhe x Breite) der Darstellung eines Medien-Objektes in Pixeln fest. Wird diese Option nicht genutzt, so werden die Medien-Objekte nicht skaliert und in Originalgröße angezeigt.
- *Thumbnail*; gibt an (true/false), ob ein Medien-Objekt als Thumbnail präsentiert werden soll oder nicht.

Der Kontrolleintrag **P.** kann wie folgt mit den Optionen angegeben werden: **P.**[Zeitdauer, Geschwindigkeit, Größe, Thumbnail].

Um eine zeitliche und räumliche Formulierung der Ausgabebedingungen zu ermöglichen, kann für jede auszugebende Variable eine *Print-Variable* entsprechend der Notation ***P.****VariablenName* angegeben werden. Wird keine Print-Variable angegeben, so werden die entsprechenden Ergebnisobjekte standardmäßig in Tabellenform präsentiert.

Für die zeitliche Anordnung der auszugebenden Medien-Objekte wird auf die von Allen (1983) entwickelten temporalen Beziehungen zwischen Objekten zurückgegriffen. Die Angabe bezüglich einer konkreten Zeitdauer wird der '$<$'-Relation sowie der '$/$'-Relation hinzugefügt. Die Print-Variablen werden dann in einer *Temporal-Condition-Box* zueinander in Beziehung gesetzt. Hierfür wird das Prinzip der hierarchischen Komposition (siehe Abschnitt 3.1.3) verwendet. Die zeitlichen Beziehungen werden nicht in Form einer Baumstruktur angegeben, sondern als komplexe Bedingung in Form eines *Zeit-Strings*. Ein Zeit-String ist wie folgt definiert:

Definition 6.3.1. In einem *Zeit-String* werden unter Verwendung der Vergleichsoperatoren aus der Menge $R = \{<, :, |, \%, [,], /\}$ die zeitlichen Beziehungen zwischen Objekten und komplexen Objekten beschrieben. Bei Einsatz der Relationen $<$ und $/$ ist die Angabe einer Zeitdauer erforderlich. Dies erfolgt für zwei Objekte o_1 und o_2 in der Form $(o_1 < o_2, Zeitdauer)$ beziehungsweise $(o_1 / o_2, Zeitdauer)$. Die Zeitdauer wird in Sekunden angegeben. □

Zur Erstellung eines entsprechenden Zeitplans für die Präsentation der Objekte kann nach Subrahmanian (1998) auf den Bellman-Ford Algorithmus zurückgegriffen werden. Die Verwendung des Zeit-Strings und der Print-Option wird im folgenden Beispiel anhand einer Anfrage verdeutlicht.

Beispiel 6.8 *Gib alle Informationen zu dem Stoff mit der Nummer I002 aus. Nachdem die Videoaufnahme abgespielt wurde, zeige das Foto an und spiele gleichzeitig das Audio-File ab. Beides soll*

jeweils 20 Sekunden präsentiert werden. Das Werbefoto soll in einer Größe von 300x400 Pixeln angezeigt werden. Der Name, die Qualität und die SId sollen ständig angezeigt werden.

Stoff	SId	Name	Qualitaet	Video	Werbefoto	Audio
P.	I002			**P.**Video	**P.**Foto[20,,300x400,false]	**P.**Audio[20,,,]

Temporal-Condition-Box	Condition
	P.Video ∣ (**P.**Foto ⌈ **P.**Audio)

Analog zur Spezifikation der zeitlichen Darstellung der Ergebnisobjekte kann eine besondere räumliche Präsentation der Ergebnisobjekte vom Nutzer gewünscht werden. Papadias und Sellis (1995) verwenden für die Formulierung in einer QBE-ähnlichen Anfragesprache ein Bild-Skelett. So liefert die Anfrage:

Bilder

	P. _X
Bild	

all die Bild-Objekte *X* aus der Relation *Bilder*, die sich nordöstlich des Referenzbildes *Bild* befinden. Diese Art der Formulierung von räumlichen Beziehungen kann prinzipiell auch für die Spezifikation der Ausgabebedingungen gewählt werden. In dieser Arbeit wird jedoch ein grafisches Layout favorisiert.

In WS-QBE wird die Formulierung der räumlichen Präsentationsmerkmale durch eine räumliche Anordnung der Print-Variablen in einem *Spatial-Condition-Frame* unterstützt. Dafür werden den Print-Variablen geometrische Objekte zugewiesen. Alternativ können die räumlichen Beziehungen auf Basis der Print-Variablen durch Angabe eines *2D-C-Strings* (siehe Abschnitt 3.1.2) erfolgen. Dieser wird in einer *Spatial-Condition-Box* formuliert. Die Abbildung 6.4 zeigt für die Print-Variablen **P.**Video und **P.**Foto aus Beispiel 6.8 eine räumliche Anordnung im Spatial-Condition-Frame.

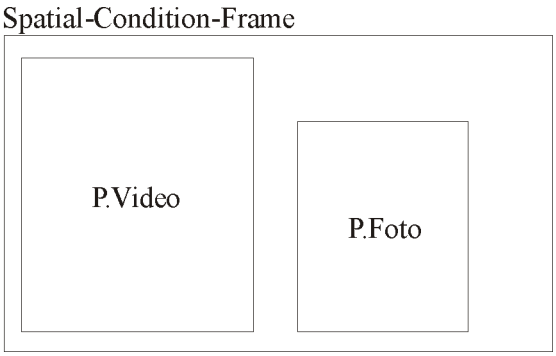


Abbildung 6.4: Räumliches Layout im Spatial-Condition-Frame

Dies kann auch wie folgt in der Spatial-Condition-Box formuliert werden:

Spatial-Condition-Box	Condition
	(**P.**Video $<$ **P.**Foto); (**P.**Video $[$ **P.**Foto)

Bei Anfragen, die Ausgaben aus mehreren Tabellen umfassen, ist in WS-QBE der Einsatz einer temporären Ausgabetabelle notwendig. In dieser werden die auszugebenden Variablen sowie die Print-Variablen, für die keine räumlichen Ausgabebedingungen spezifiziert wurden, geschrieben. Die Struktur der Tabelle kann vom Nutzer festgelegt werden. Beispielsweise können die Spaltennamen beliebig benannt werden.

6.4 Syntax

Die Angabe einer formalen Syntax für visuelle Anfragesprachen erfolgt nur selten, da die Ausdrucksfähigkeit der visuellen Anfragesprache weit über die Mächtigkeit einer String-basierten Grammatik hinausgeht (Catarci u. a., 1997). Daher findet im Allgemeinen ein Mapping der visuellen Anfragesprache auf eine textuelle Anfragesprache statt. Für die kann wiederum eine formale Syntax definiert werden. Bei QBE-ähnlichen Anfragen wird die Syntax der visuellen Operationen meist durch äquivalente Textkonstrukte beschrieben (ebd.). Aus diesem Grund werden auch WS-QBE-Anfragen auf entsprechende textuelle Sprachkonstrukte überführt, deren entsprechende formale Syntax in diesem Abschnitt beschrieben wird.

6.4.1 Prinzipien der Abbildung von WS-QBE auf Textkonstrukte

Eine WS-QBE-Anfrage besteht aus verschiedenen Sprachelementen. Dazu zählen die Anfragetabellen, die Ausgabetabelle, die Weighting-Tabelle, der Condition-Tree, die Spatial-Condition-Box und die Temporal-Condition-Box sowie der Spatial-Condition-Frame. Die Anfragetabellen und die Ausgabetabelle werden in der textuellen Notation durch die Sprachkonstrukte *Tabelle* und *Ausgabetabelle* umgesetzt.

In WS-QBE wird eine Relevanzgewichtung sowie eine Schwellwert- und Operatorgewichtung unterstützt. Dabei werden die Schwellwertgewichte direkt den betreffenden Selektionsbedingungen zugeordnet. Im Condition-Tree werden die Operatorgewichtung und die Relevanzgewichtung visuell unterstützt. Anders als bei einfachen Anfragen werden hier keine Relevanzvariablen angelegt. Bei der Überführung einer WS-QBE-Anfrage in die textuelle Notation wird für jede Art von Gewicht entweder eine *Relevanz-*, *Schwell-* oder *Operatorvariable* erzeugt. Diesen Variablen wird in der *Gewichtung* ein entsprechendes numerisches Gewicht zugeteilt.

Die Anfragebedingungen, die in WS-QBE im Condition-Tree erstellt werden, können durch Traversieren des Baums in Inorder-Reihenfolge als *Bedingung* in einer textuellen Sprache abgebildet werden. Sind räumliche Bedingungen im Condition-Tree angegeben, so werden zuvor sowohl Richtungsbeziehungen als auch topologische Beziehungen zwischen den Objekten aus der Skizze extrahiert. Die verschiedenen räumlichen Beziehungen zwischen jeweils zwei Objekten werden konjunktiv miteinander verknüpft.

In WS-QBE ist es ferner möglich, als Selektionsbedingung nur eine Konstante *Const* anzugeben. Dies impliziert einen Test auf Gleichheit zwischen den Werten des entsprechenden

Attributes und der Konstanten. In der textuellen Notation wird diese vereinfachte Schreibweise aufgelöst, so dass '= *Const*' geschrieben wird.

Des Weiteren besteht in WS-QBE die Möglichkeit, räumliche und zeitliche Ausgabebedingungen zu spezifizieren. Die zeitlichen Bedingungen aus der Temporal-Condition-Box werden unter Verwendung der *Print-Variablen* textuell in Form einer *zeitl. Bedingung* definiert. Analog dazu werden die räumlichen Ausgabebedingungen als *räuml. Bedingung* spezifiziert. Räumliche Bedingungen können in WS-QBE visuell in einem Spatial-Condition-Frame oder textbasiert in der Spatial-Condition-Box erfolgen. Für die Abbildung des visuellen Layouts auf eine textuelle räumliche Bedingung wird aus dem Layout ein 2D-C-String generiert.

6.4.2 Syntaxdefinition

Mit diesen skizzierten Sprachmitteln ist es möglich, die visuelle Multimedia-Anfragesprache WS-QBE auf eine textuelle Sprache abzubilden. Für die Beschreibung der Syntax wird auf einer Erweiterung der Backus-Naur-Form (Backus u. a., 1960, 1963), einer kontextfreien Grammatik, aufgesetzt. Die verwendete Notation ist in Tabelle 6.1 erklärt.

Symbol	Bedeutung
[ ]	optional
{ }	beliebige Wiederholungen
$\{\}^{+}$	beliebige Wiederholungen aber mindestens einmal
$\{\}^{2+}$	beliebige Wiederholungen aber mindestens zweimal
|	oder

Tabelle 6.1: Überblick über die verwendete Notation

Damit kann die Syntax wie folgt angegeben werden:

Anfrage: Eine Anfrage kann beliebig viele Tabellen und optional eine Bedingung, eine räumliche und eine zeitliche Ausgabebedingung, eine Gewichtung sowie eine Ausgabetabelle umfassen.

<Anfrage> ::= {<Tabelle>}$^{+}$
[<Bedingung>]
[<räuml. Bedingung>]
[<zeitl. Bedingung>]
[<Gewichtung>]
[<Ausgabetabelle>]

Tabelle: Eine Tabelle verfügt über einen Relationennamen. Optional kann eine Schwellvariable angegeben werden, die für die gesamte Anfrage gültig ist. Dem folgt eine Anzahl von Attributnamen. In einer Anfragezeile wird jedem Attributnamen eine Selektionsbedingung oder eine Variable zugeordnet. Zu Beginn einer Anfragezeile darf ein Tupelbefehl stehen.

<Tabelle> ::= <Relationenname> [<Schwellvariable>] {<Attributname>}$^{+}$
{[<Tupelbefehl>] <Anfragezeile>}$^{+}$

Tupelbefehl: Dieser Befehl gilt für ein ganzes Tupel. So darf hier das Negationssymbol ¬ oder der Kontrolleintrag **P.**, was die Ausgabe aller Attribute zur Folge hat, stehen.

<Tupelbefehl> ::= ¬| **P.**

Anfragezeile: Für jeden Attributnamen wird entweder eine Variable oder eine Selektionsbedingung angegeben. Dem kann der Kontrolleintrag **P.** oder eine Print-Variable vorangestellt werden. Weiterhin können bei einer Selektionsbedingung Variablen für ein Schwellwert- und Relevanzgewicht angegeben werden.

<Anfragezeile> ::= {[**P.** | <Print-Variable>]
<Variable> |
<Selektionsbedingung> [<Schwellvariable>]
[<Relevanzvariable>] }$^{+}$

Selektionsbedingung: Eine Selektionsbedingung stellt einen Vergleich auf Zahlen, Zeichenketten und Multimedia-Objekten dar. Selektionsbedingungen werden für den Vergleich mit den Komponenten eines Tupels benötigt.

<Selektionsbedingung> ::= <Vergleich> <num. Term> |
<Vergleich> <alpha. Term> |
<MMVergleich><sim. Term> |
<Zeitlicher Vergleich> <sim. Term> |
<Räumlicher Vergleich><sim. Term> |
<leer>

Bedingung: Eine Bedingung entspricht einer Formel. Dabei ist jedes Atom eine Formel. Darüber hinaus kann eine Formel unter Anwendung der klassischen Junktoren ($\wedge$, $\vee$) und des Universaljunktors ($\oplus$) aus jeweils mindestens zwei Formeln gebildet werden. Ferner ist die Negation einer Formel gestattet. Optional kann eine Gewichtung erfolgen, indem einer Formel eine Schwellwert- beziehungsweise Relevanzvariable zugeordnet wird. Die Parametrisierung des Universaljunktors erfolgt über eine Operatorvariable.

<Bedingung> ::= <Formel>

<Formel> ::= (<Atom>) [<Schwellvariable>] [<Relevanzvariable>] |
($\wedge$ ({<Formel>}$^{2+}$)) [<Schwellvariable>] [<Relevanzvariable>] |
($\vee$ ({<Formel>}$^{2+}$)) [<Schwellvariable>] [<Relevanzvariable>] |
(<$\oplus$> ({<Formel>}$^{2+}$)) [<Schwellvariable>] [<Relevanzvariable>] |
(¬ <Formel>) [<Schwellvariable>] [<Relevanzvariable>]

<Atom> ::= <num. Term> <Vergleich> <num. Term> |
<alpha. Term> <Vergleich> <alpha. Term> |
<sim. Term> <MMVergleich> <sim. Term> |
<sim. Term> <Zeitlicher Vergleich> <sim. Term> |

<sim. Term> <Räumlicher Vergleich> <sim. Term>

<⊕> ::= ⊕<Operatorvariable>

räuml. Bedingung: Eine räumliche Ausgabebedingung wird durch den 2D-C-String (siehe Abschnitt 3.1.2) realisiert. Dabei wird sowohl für die x als auch für die y-Dimension paarweise ein Richtungsvergleich zwischen den einzelnen Print-Variablen spezifiziert.

<räuml. Bedingung> ::= <2D-C-String>

<2D-C-String> ::= (<X>; <Y>)
<X> ::= {(<Print-Variable> <2D-Vergleich> <Print-Variable>)}$^+$
<Y> ::= {(<Print-Variable> <2D-Vergleich> <Print-Variable>)}$^+$

zeitl. Bedingung: Eine zeitliche Ausgabebedingung wird durch einen Zeit-String (siehe Abschnitt 6.3.4) spezifiziert.

<zeitl. Bedingung> ::= <Zeit-String>

<Zeit-String> ::= {(<Print-Variable> <Zeitlicher Vergleich>
<Print-Variable> [, <Dauer>])}$^+$

Gewichtung: In der Gewichtung wird jeder Relevanz-, Schwell- oder Operatorvariablen ein Gewicht in Form eines numerischen Wertes zugeordnet.

<Gewichtung> ::= {(<Relevanzvariable> | <Operatorvariable> | <Schwellvariable>
= <Gewicht>)}$^+$

Ausgabetabelle: In einer Ausgabetabelle wird der Name der Relation sowie eine beliebige Anzahl von Attributnamen angegeben. Jedem Attributnamen ist in der Anfragezeile eine Variable zugeordnet. Zusätzlich kann in der Ausgabetabelle eine Schwellvariable angegeben werden. Das dafür spezifizierte Schwellwertgewicht ist dann für die ganze Anfrage gültig.

<Ausgabetabelle> ::= <Relationenname> [<Schwellvariable>] {<Attributname>}$^+$
<Ausgabezeile>

<Ausgabezeile> ::= {**P.** <Variable> | <Print-Variable>}$^+$

Vergleiche: Je nach Medientyp stehen verschiedene Vergleiche zur Verfügung.

<Vergleich> ::= $< \mid > \mid = \mid \neq \mid \geq \mid \leq \mid \sim$
<MMVergleich> ::= $= \mid \neq \mid \sim \mid \sim_{Farbe} \mid \sim_{Form} \mid \sim_{Textur}$
<Zeitlicher Vergleich> ::= $< \mid : \mid -^2 \mid \% \mid [\mid] \mid /$

[2] Das Symbol $-$ entspricht dem Symbol |, welches in Abschnitt 3.1.2 eingeführt wurde.

<Räumlicher Vergleich>	::=	<Richtungsvergleich> | <Topolog. Vergleich> | <Distanzvergleich>
<Richtungsvergleich>	::=	links_von | rechts_von | oben_von | unten_von
<Topolog. Vergleich>	::=	disjunkt | berührt | in | überlappt | überdeckt | gleich
<Distanzvergleich>	::=	sehr_nah | nah | fern | sehr_fern
<2D-Vergleich>	::=	< | : | $-$[3] | % | [|] | /

Terme:

<num. Term>	::=	<Zahlkonstante> | <Variable>
<alpha. Term>	::=	<Textkonstante> | <Variable>
<sim. Term>	::=	<Multimediakonstante> | <Variable>
<Variable>	::=	_{<Zeichen ohne Sonderzeichen>}$^+$
<Operatorvariable>	::=	% {<Zeichen ohne Sonderzeichen>}$^+$
<Relevanzvariable>	::=	$ {<Zeichen ohne Sonderzeichen>}$^+$
<Schwellvariable>	::=	& {<Zeichen ohne Sonderzeichen>}$^+$
<Print-Variable>	::=	**P.**{<Zeichen ohne Sonderzeichen>}$^+$

Konstanten und Bezeichner:

<Relationenname>	::=	{<Zeichen ohne Sonderzeichen>}$^+$
<Attributname>	::=	{<Zeichen ohne Sonderzeichen >}$^+$
<Textkonstante>	::=	{<Zeichen>}$^+$
<Zahlkonstante>	::=	<übliche Zahlen>
<Multimediakonstante>	::=	<übliches Multimedia-Objekt>
<Gewicht>	::=	<übliche Zahlen $\in$ [0,1] >
<Dauer>	::=	<übliche positive Zahlen >

Unter Verwendung dieser Syntax kann für jede visuelle Anfrage eine entsprechende textuelle Notation angegeben werden. Auf die Besonderheiten, die bei der Abbildung der visuellen Anfragen auf einen textuellen Ausdruck zu berücksichtigen sind, wurde bereits im vorigen Abschnitt eingegangen. Im folgenden Abschnitt wird die Semantik für einen textuellen Ausdruck, der eine visuelle WS-QBE-Anfrage umsetzt, beschrieben.

6.5 Semantik

In diesem Abschnitt wird die formale Semantik für einen WS-QBE-Anfrageausdruck durch Transformation in einen $\mathcal{SDC}$-Ausdruck definiert. Zunächst wird jedoch die Semantik der Operatoren und Junktoren festgelegt, da diese im Ähnlichkeitskalkül nicht vordefiniert sind.

[3] Das Symbol $-$ entspricht dem Symbol |, welches in Abschnitt 3.1.2 eingeführt wurde.

6.5.1 Operationen

Die klassischen Vergleichsoperatoren $<, >, =, \neq, \geq, \leq$ weisen die allgemein bekannte Semantik auf.

Für Ähnlichkeitsvergleiche auf Multimedia-Daten wird in WS-QBE die Euklidische Distanz auf den entsprechenden Feature-Werten der Objekte eingesetzt, während die Ähnlichkeit zwischen Zeichenketten mittels der Editierdistanz und zwischen Zahlen als Betrag der Differenz bestimmt wird. Die ermittelten Distanzwerte müssen dann in einen Ähnlichkeitswert überführt werden. Eine Anpassung der Ähnlichkeits- beziehungsweise der Distanzmaße je nach Anwendungsgebiet ist möglich. So kann für den Operator $\sim_{Farbe}$ anstelle der Euklidischen Distanz auch der Histogrammschnitt verwendet werden.

Die Semantik für die zeitlichen und räumlichen Prädikate wurde bereits in Abschnitt 3.1.2 dargelegt. Aus diesem Grund wird an dieser Stelle auf diesen Abschnitt verwiesen. Für die qualitativen Distanzsymbole ist die Semantik kontextabhängig. Aus diesem Grund können im Rahmen dieser Arbeit keine konkreten, allgemein gültigen quantitativen Intervalle festgelegt werden.

6.5.2 Junktoren

Die Anfragesprache WS-QBE basiert auf dem klassischen Fuzzy-Modell. Daher wird die Semantik für den Junktor $\wedge$ durch die Funktion *min* und für $\vee$ durch die Funktion *max* definiert. Für die Negation $\neg$ wird das Einerkomplement verwendet. Damit kann die Interpretation I für diese Operatoren wie folgt angegeben werden:

$$\begin{aligned} I(\wedge) &= min(\mu_1, \ldots, \mu_m) \\ I(\vee) &= max(\mu_1, \ldots, \mu_m) \\ I(\neg) &= 1 - \mu_1, \end{aligned}$$

wobei $\mu_1, \ldots \mu_m$ Relevanzwerte im Intervall $[0, 1]$ sind. Für die gewichteten Alternativen dieser beiden Junktoren wird eine gewichtete T-Norm beziehungsweise T-Conorm unter Verwendung der Gewichtungsformel (vgl. Abschnitt 4.2.1) erzeugt. Damit wird die Semantik für die gewichteten n-ären Junktoren $\wedge_{(\theta_1,\theta_2,\ldots,\theta_m)}$ und $\vee_{(\theta_1,\theta_2,\ldots,\theta_m)}$ folgendermaßen festgelegt:

$$\begin{aligned} I(\wedge_{(\theta_1,\theta_2,\ldots,\theta_m)}) = \; & (\theta_1 - \theta_2) * min(\mu_1) + \\ & 2 * (\theta_2 - \theta_3) * min(\mu_1, \mu_2) + \\ & \ldots + \ldots \\ & m * \theta_m * min(\mu_1, \ldots, \mu_m) \end{aligned}$$

$$\begin{aligned} I(\vee_{(\theta_1,\theta_2,\ldots,\theta_m)}) = \; & (\theta_1 - \theta_2) * max(\mu_1) + \\ & 2 * (\theta_2 - \theta_3) * max(\mu_1, \mu_2) + \\ & \ldots + \ldots \\ & m * \theta_m * max(\mu_1, \ldots, \mu_m), \end{aligned}$$

wobei $\mu_1, \ldots \mu_m$ Relevanzwerte im Intervall $[0, 1]$ sind. Neben diese Junktoren besteht in WS-QBE die Möglichkeit, den Universaljunktor $\oplus^z$ und den gewichteten Universaljunktor

$\oplus^z_{(\theta_1,\theta_2,\ldots,\theta_m)}$ zu verwenden. Deren Semantik ist unter Beachtung der Relevanzwerte und der Interpretation des Operatorgewichts z wie folgt definiert:

$$I(\oplus^z) = I(z) * I(\wedge)(\mu_1,\ldots,\mu_m) + (1 - I(z)) * I(\vee)(\mu_1,\ldots,\mu_m)$$

$$\begin{aligned} I(\oplus^z_{(\theta_1,\theta_2,\ldots,\theta_m)}) = {} & I(z) * I(\wedge_{(\theta_1,\theta_2,\ldots,\theta_m)})(\theta_1,\ldots,\theta_m,\mu_1,\ldots,\mu_m) + \\ & (1 - I(z)) * I(\vee_{(\theta_1,\theta_2,\ldots,\theta_m)})(\theta_1,\ldots,\theta_m,\mu_1,\ldots,\mu_m). \end{aligned}$$

6.5.3 Transformation einer WS-QBE-Anfrage in das $\mathcal{SDC}$

Die Semantik einer WS-QBE-Anfrage wird durch die Überführung auf einen Ausdruck des Ähnlichkeitskalküls definiert. Die Überführung basiert auf der in Abschnitt 6.2.1 beschriebenen Überführung von QBE in einen Ausdruck des Bereichskalküls. Auf Grund der erweiterten Funktionalität von WS-QBE sind jedoch Modifizierungen notwendig.

Die Ausgabebedingungen werden nicht in das $\mathcal{SDC}$ überführt, da diese keine Kriterien für die Bestimmung der relevanten Ergebnisobjekte sind. Sie sind lediglich für die Präsentation der Ergebnisobjekte von Bedeutung. Demnach werden sie erst nach der Ermittlung der Ergebnisobjekte benötigt. Die Ausgabebedingungen fließen somit als Parametrisierung in die Präsentationskomponente des Anfragesystems ein.

Eine Besonderheit tritt bei der Überführung des Universaljunktors auf: Der Universaloperator $\oplus^z$ wird bei einem Operatorgewicht mit dem Wert 1 auf den Junktor $\wedge$ beziehungsweise bei Verwendung von Relevanzgewichten auf den gewichteten *UND*-Operator $\wedge_{(\theta_1,\theta_2,\ldots,\theta_m)}$ abgebildet. Dieses wird notwendig, da im $\mathcal{SDC}$ der Universaljunktor als Verallgemeinerung der Disjunktion definiert ist und daher nur Gewichte aus dem Interval $[0,1)$ unterstützt.

Unter Berücksichtigung dieser Besonderheit kann die Umsetzung einer WS-QBE Anfrage in einen Ausdruck des Ähnlichkeitskalküls $E = \{x_1,\ldots,x_n | F(x_1,\ldots,x_n)\}$ wie folgt vorgenommen werden:

1. Für alle Attribute A_i aller vorkommenden Anfragezeilen der Anfrage wird eine Variable X_i erzeugt.
2. Für jede Relevanzvariable $\$RelVar_j$ wird eine Gewichtsvariable θ_j erzeugt.
3. Für jede Schwellvariable $\&SchVar_k$ wird eine Gewichtsvariable θ_k erzeugt.
4. Für jede Operatorvariable $\%OpeVar_l$ wird eine Operatorgewichtsvariable z_l erzeugt.
5. Die Variablen $_Var$ der Ausgabetabelle werden zu den freien Variablen geschrieben: $(\{\ldots,_Var,\ldots \mid \ldots\})$.
6. Für alle positiven Anfragezeilen gilt:
   a) Steht in der Anfragezeile mit $A_l,\ldots,A_m$ der Tupelbefehl **P.**, dann wird jede Variable X_j mit $j = l,\ldots,m$ zu den freien Variablen geschrieben: $(\{\ldots,X_l,\ldots,X_m\ldots \mid \ldots\})$.
   b) Steht bei einem Attribut A_i das Kommando **P.** oder eine Print-Variable **P.***Var*, so wird die entsprechende Variable X_i zu den freien Variablen geschrieben: $(\{\ldots,X_i, \ldots \mid \ldots\})$.

c) Für jede Relation R wird unter Verwendung der Bereichsvariablen der jeweiligen Anfragezeile ein Atom $R(X_i, X_{i+1}, \ldots)$ erzeugt. Diese Atome bilden konjunktiv miteinander verknüpft die Formel F.

d) Für jedes A_i mit einer Selektionsbedingung $\delta Const$ beziehungsweise δ_Var mit $\delta \in \{\text{Vergleich} \cup \text{MMVergleich} \cup \text{Zeitlicher Vergleich} \cup \text{Räumlicher Vergleich}\}$ gilt:

i) Es wird eine Teilformel $F_i = (X_i \delta Const)$ beziehungsweise $F_i = (X_i \delta_Var)$ erzeugt.

ii) Ist zusätzlich eine Schwellvariable $\&SchVar_k$ angegeben, dann werden die Teilformeln für ein A_i unter Verwendung des softcut-Operators erzeugt: $F_i = (\succsim_{\theta_k}(X_i \delta Const))$ beziehungsweise $F_i = (\succsim_{\theta_k}(X_i \delta_Var))$.

e) Steht bei A_i eine Variable $_Var$, dann wird eine Teilformel $F_i = (X_i = _Var)$ erzeugt.

f) Alle Teilformeln F_i der Attribute A_i, bei denen eine Relevanzvariable $\$RelVar_j$ auftritt, werden unter Beachtung der Relevanzvariablen mit $\wedge_{(\theta_1,\theta_2,\ldots,\theta_m)}$ zu F_a verknüpft. Die übrigen Teilformeln F_i werden mit $\wedge$ zu F_b verknüpft. F_a und F_b werden dann konjunktiv mit F verknüpft.

7. Für alle negierten Anfragezeilen gilt:

a) Für jede Relation R wird unter Verwendung der Bereichsvariablen der jeweiligen Anfragezeile ein Atom $R(X_i, X_{i+1}, \ldots)$ erzeugt. Diese Atome bilden konjunktiv miteinander verknüpft die Teilformel F_{neg}.

b) Für jedes A_i mit einer Selektionsbedingung $\delta Const$ beziehungsweise δ_Var mit $\delta \in \{\text{Vergleich} \cup \text{MMVergleich} \cup \text{Zeitlicher Vergleich} \cup \text{Räumlicher Vergleich}\}$ gilt:

i) Es wird eine Teilformel $F_i = (X_i \delta Const)$ beziehungsweise $F_i = (X_i \delta_Var)$ erzeugt.

ii) Ist zusätzlich eine Schwellvariable $\&SchVar_k$ angegeben, dann werden die Teilformeln für A_i unter Verwendung des softcut-Operators erzeugt: $F_i = (\succsim_{\theta_k}(X_i \delta Const))$ beziehungsweise $F_i = (\succsim_{\theta_k}(X_i \delta_Var))$.

c) Steht bei A_i eine Variable $_Var$, dann wird eine Teilformel $F_i = (X_i = _Var)$ erzeugt.

d) Alle Teilformeln F_i der Attribute A_i, bei denen eine Relevanzvariable $\$RelVar_j$ auftritt, werden unter Beachtung der Relevanzvariablen mit $\wedge_{(\theta_1,\theta_2,\ldots,\theta_m)}$ zu F_a verknüpft. Die anderen Teilformeln F_i werden mit $\wedge$ zu F_b verknüpft. F_a und F_b werden konjunktiv mit F_{neg} verknüpft.

e) Alle Variablen X_i, die in einer negierten Anfragezeile auftreten, werden in F_{neg} gebunden: $\{\ldots|(\ldots(\exists X_i(F_{neg})))\}$.

f) $\neg F_{neg}$ wird konjunktiv mit F verknüpft.

8. Die in der Bedingung angegeben Formel wird konjunktiv mit F verknüpft. Dabei werden die Junktoren in WS-QBE wie folgt in das $\mathcal{SDC}$ umgesetzt[4]:

[4] Bei zwei Operanden wird die Infixnotation verwendet.

- Das Symbol '$\wedge$' wird auf den Operator $\wedge$ beziehungsweise bei gewichteten Teilformeln auf $\wedge_{(\theta_1,\theta_2,\ldots,\theta_m)}$ mit $\theta_j = \$RelVar_j$ und $j = 1,\ldots,m$ abgebildet,
- Das Symbol '$\vee$' wird auf den Operator $\vee$ beziehungsweise bei gewichteten Teilformeln auf $\vee_{(\theta_1,\theta_2,\ldots,\theta_m)}$ mit $\theta_j = \$RelVar_j$ und $j = 1,\ldots,m$ abgebildet,
- Das Symbol '$\neg$' wird auf den Operator $\neg$ abgebildet,
- Das Symbol '$\oplus$' mit der Operatorvariablen $\%OpeVar_l$ und einem Operatorgewicht kleiner *eins* wird auf den Operator $\oplus^z$ mit $z = \%OpeVar_l$ beziehungsweise bei gewichteten Teilformeln auf $\oplus^z_{(\theta_1,\theta_2,\ldots,\theta_m)}$ mit $\theta_j = \$RelVar_j$ und $z = \%OpeVar_l$, abgebildet. Es gilt $j = 1,\ldots,m$.
  Beträgt das Operatorgewicht *eins*, dann wird '$\oplus$' mit $\%OpeVar_l$ auf den Operator $\wedge$ beziehungsweise bei gewichteten Teilformeln auf $\wedge_{(\theta_1,\theta_2,\ldots,\theta_m)}$ und $\theta_j = \$RelVar_j$ mit $j = 1,\ldots,m$ abgebildet.

9. Alle in einer positiven Anfragezeile auftretenden Variablen X_i sowie $_Var$, die nicht zu den freien Variablen gehören, werden mit $\exists$ in F gebunden: $\{\ldots|\ldots(\exists X_i(F))\}$ beziehungsweise $\{\ldots|\ldots(\exists_Var(F))\}$.

10. Steht hinter dem Relationenname einer Ausgabetabelle oder einer Tabelle, in der ein **P.** oder eine Print-Variable auftritt, eine Schwellvariable $\&SchVar_k$, dann wird auf F der softcut-Operator angewendet: $(\{\ldots|(\rightsquigarrow_{\theta_k}(F))\})$.

11. Vereinfachung der Formel.

Durch eine Interpretationsfunktion I werden für die Gewichtsvariablen θ und die Operatorgewichtsvariablen z die konkreten numerischen Werte bestimmt. Es gilt $I(\theta) \in [0,1]$ und $I(z) \in [0,1)$.

Zusammenfassend stellt Tabelle 6.4 die wichtigsten Sprachmittel von WS-QBE und des Ähnlichkeitskalküls gegenüber.

WS-QBE	Ähnlichkeitskalkül
Konstanten	Konstanten
Variablen	Variablen
Relevanzvariablen	Gewichtsvariablen, gewichtete Junktoren
Schwellvariablen	Gewichtsvariablen, $softcut$-Operator
Operatorvariablen	Operatorgewichtsvariablen, Universaljunktor
leere Spalten	paarweise verschiedene mit $\exists$-quantifizierte Variablen
Spalten mit **P.**/Print-Variable	freie Variablen
Spalten ohne **P.**/Print-Variable	$\exists$-quantifizierte Variablen

Tabelle 6.4: Gegenüberstellung der Sprachmittel von WS-QBE und dem Ähnlichkeitskalkül

6.6 Beispiel

In diesem Abschnitt wird an einem Beispiel die Anfrageformulierung in WS-QBE sowie die Überführung der Anfrage in einen entsprechenden Ausdruck des Ähnlichkeitskalküls de-

monstriert. Es dienen die folgenden, bekannten Relationen des Stoffherstellers als Ausgangsbasis:

Stoff

SId	Name	Qualitaet
F001	tartan_0815	hoch
F002	tartan_0816	mittel
F003	blue_stripes	hoch
...	...	...

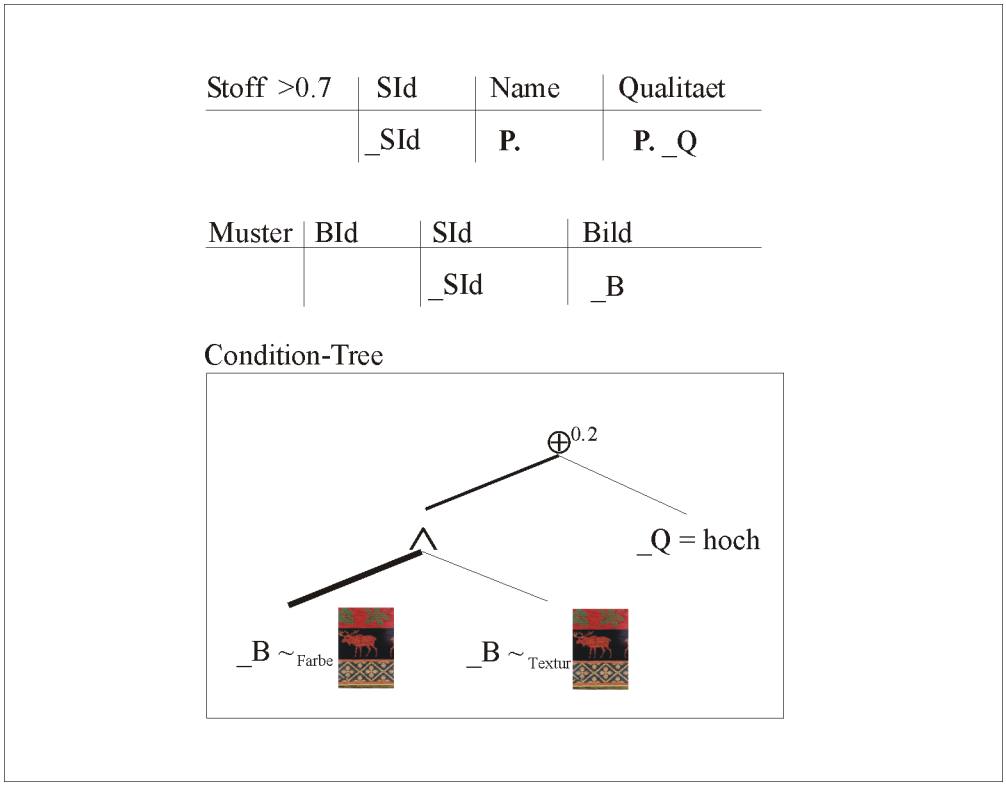
Muster

BId	SId	Bild
I001	F001	tartan_0815_1.jpg
I002	F001	tartan_0815_2.jpg
I003	F002	tartan_0816_1.jpg
I004	F003	blue_stripes_1.jpg
...	...	...

Als Beispielanfrage wird auf die Anfrage aus Abschnitt 5.5.2 zurückgegriffen:

> *Ermittle den Namen und die Qualität der Stoffe, die eine hohe Qualität haben oder dem gegebenen Bild (C_{Bild}) in Farbe und Textur ähnlich sind. Die Verknüpfung dieser Bedingungen wird mit z=0.2 reglementiert. Die Farbähnlichkeit ist wichtiger als die Ähnlichkeit bezüglich der Textur (Gewicht von 0.9 versus 0.1). Ferner ist die Qualität eines Stoffes weniger wichtig als die Ähnlichkeit eines Stoffes zum vorgegebenen Stoffmuster (Gewicht von 0.3 versus 0.7). Letztlich sollen nur die Stoffe ermittelt werden, die einen Relevanzwert größer 0.7 aufweisen.*

Diese Anfrage kann wie folgt in WS-QBE formuliert werden:



Die WS-QBE-Anfrage, bestehend aus den zwei Anfragetabellen *Stoff* und *Muster* und einem Condition-Tree, wird als erstes in einen entsprechenden textuellen Anfrageausdruck

überführt. Die Abbildung der Tabellen kann direkt erfolgen: Für die WS-QBE-Tabellen werden die Tabellen *Stoff* und *Muster* mit den entsprechenden Attributnamen angelegt. Die Anfragezeilen können übernommen werden. Für das Schwellwertgewicht in der Relation *Stoff* wird eine Schwellvariable &S erzeugt.

Zur Überführung des Condition-Trees werden zunächst für die einzelnen Relevanzgewichte sowie für das Operatorgewicht entsprechende Operator- beziehungsweise Relevanzvariablen erzeugt. In der Gewichtung werden diesen Variablen die dazugehörigen numerischen Werte zugeordnet. Zusammen mit der Schwellvariablen ergibt sich ((&S=0.7)($F = 0.9)($T = 0.1)($sim = 0.7)($Q = 0.3)(%z=0.2)). Der Condition-Tree mit den drei Selektionsbedingungen wird sodann durch Traversieren des Baumes in Inorder-Reihenfolge und unter Berücksichtigung der Operator- und Relevanzvariablen in eine Bedingung, also eine Formel, überführt. Weitere Besonderheiten, sind bei der Abbildung auf textuelle Sprachkonstrukte nicht zu berücksichtigen.

Die Abbildung auf das Ähnlichkeitskalkül erfolgt entsprechend der im vorhergehenden Abschnitt geschilderten Schritte. Zum bessern Verständnis wird im Folgenden jeder Schritt noch einmal einzeln aufgeführt:

1. Variablen erzeugen: $X_1, X_2, X_3, X_4, X_5, X_6$
2. Gewichtsvariablen für Relevanzvariablen erzeugen: $\theta_F, \theta_T, \theta_{sim}, \theta_Q$
3. Gewichtsvariablen für Schwellvariable erzeugen: θ_S
4. Operatorgewichtsvariablen erzeugen: z
5. Es existiert keine Ausgabetabelle.
6. Für positive Anfragezeilen:
   a) Es gibt kein Tupelbefehl.
   b) Attribute, die ein **P.** enthalten : $\{X_2, X_3 | \ldots\}$
   c) Relationen erzeugen und zu F verknüpfen:
   $$\{\ldots | (Stoff(X_1, X_2, X_3) \wedge Muster(X_4, X_5, X_6))\}$$
   d) bis f) Selektionsbedingungen auswerten:
   $F_b = ((((X_1 = X_{SId}) \wedge (X_3 = X_Q)) \wedge (X_5 = X_{SId})) \wedge (X_6 = X_B))$

   F_b mit F verknüpft, ergibt:
   $$\{X_2, X_3 \mid ((Stoff(X_1, X_2, X_3) \wedge Muster(X_4, X_5, X_6)) \wedge (((((X_1 = X_{SId}) \wedge (X_3 = X_Q)) \wedge (X_5 = X_{SId})) \wedge (X_6 = X_B)))\}$$
7. Es sind keine negierten Zeilen vorhanden.
8. Bedingung: $(((X_6 \sim_{Farbe} C_{Bild}) \wedge_{(\theta_F, \theta_T)} (X_6 \sim_{Textur} C_{Bild})) \oplus^z_{(\theta_\sim, \theta_Q)} (X_3 = C_{hoch}))$

mit F verknüpft, ergibt:

$$\{X_2, X_3 \mid (((Stoff(X_1, X_2, X_3) \wedge Muster(X_4, X_5, X_6)) \wedge (((((X_1 = X_{SId}) \wedge (X_3 = X_Q)) \wedge (X_5 = X_{SId})) \wedge (X_6 = X_B))) \wedge (((X_6 \sim_{Farbe} C_{Bild}) \wedge_{(\theta_F, \theta_T)} (X_6 \sim_{Textur} C_{Bild})) \oplus^z_{(\theta_\sim, \theta_Q)} (X_3 = C_{hoch})))\}$$

9. Variablen binden:

$$\{X_2, X_3 \mid (\exists X_1(\exists X_4(\exists X_5(\exists X_6(\exists X_{SId}(\exists X_Q(\exists X_B (((Stoff(X_1, X_2, X_3) \wedge Muster(X_4, X_5, X_6)) \wedge (((((X_1 = X_{SId}) \wedge (X_3 = X_Q)) \wedge (X_5 = X_{SId})) \wedge (X_6 = X_B))) \wedge (((X_6 \sim_{Farbe} C_{Bild}) \wedge_{(\theta_F, \theta_T)} (X_B \sim_{Textur} C_{Bild})) \oplus^z_{(\theta_\sim, \theta_Q)} (X_3 = C_{hoch}))))))))))\}$$

10. Schwellwertgewichtung berücksichtigen:

$$\{X_2, X_3 \mid (\tilde{\sigma}_{\theta_S}(\exists X_1(\exists X_4(\exists X_5(\exists X_6(\exists X_{SId}(\exists X_Q(\exists X_B (((Stoff(X_1, X_2, X_3) \wedge Muster(X_4, X_5, X_6)) \wedge (((((X_1 = X_{SId}) \wedge (X_3 = X_Q)) \wedge (X_5 = X_{SId})) \wedge (X_6 = X_B))) \wedge (((X_6 \sim_{Farbe} C_{Bild}) \wedge_{(\theta_F, \theta_T)} (X_6 \sim_{Textur} C_{Bild})) \oplus^z_{(\theta_\sim, \theta_Q)} (X_3 = C_{hoch}))))))))))\}$$

11. Vereinfachungen vornehmen:

$$\{X_2, X_Q \mid (\tilde{\sigma}_{\theta_S}(\exists X_{SId}(\exists X_4(\exists X_B((Stoff(X_{SId}, X_2, X_Q) \wedge Muster(X_4, X_{SId}, X_B)) \wedge (((X_B \sim_{Farbe} C_{Bild}) \wedge_{(\theta_F, \theta_T)} (X_B \sim_{Textur} C_{Bild})) \oplus^z_{(\theta_\sim, \theta_Q)} (X_Q = C_{hoch}))))))\}$$

Der entstandene $\mathcal{SDC}$-Anfrageausdruck entspricht dem gegebenen Kalkülausdruck für die gleiche natürlichsprachliche Anfrage aus Abschnitt 5.5.2. Auf dieser Basis kann nun eine Überführung dieses Kalkülausdrucks in einen Ausdruck der Ähnlichkeitsalgebra erfolgen. Nach einer möglichen Anfrageoptimierung wird der Ausdruck ausgewertet. Ein entsprechendes Beispiel für die Auswertung dieser Anfrage wurde bereits in Abschnitt 5.5.2 angegeben.

6.7 Zusammenfassung

Ziel dieses Kapitels war die Entwicklung der visuellen Multimedia-Anfragesprache WS-QBE. Die formale Semantik dieser Sprache wird durch das Ähnlichkeitskalkül definiert. Diese wurde in Kapitel 5 vorgestellt.

Zu Beginn des Kapitels wurde der Entwurf einer neuen visuellen Multimedia-Anfragesprache motiviert. Es wurde argumentiert, dass insbesondere für nicht-versierte Nutzer eine visuelle QBE-ähnliche Anfragesprache gegenüber textuellen Anfragesprachen, wie beispielsweise SQL, entscheidene Vorteile aufweist. Dies wird durch verschiedene Studien gestützt. Daher wurde die Idee entwickelt, eine auf QBE-basierende Multimedia-Anfragesprache zu entwerfen.

Daraufhin wurde die Sprache QBE vorgestellt. Die grundlegenden Prinzipien sowie die Formulierung von einfachen und komplexen Anfragen wurde anhand von Beispielen veranschaulicht. Weiterhin wurde die formale Semantik von QBE-Anfragen durch Angabe eines Transformationsalgorithmus in das Bereichskalkül beschrieben. Die Anfragesprache QBE ist zwar relational vollständig, jedoch nicht streng relational vollständig. Dies schränkt die Formulierung komplexer Anfragen, die beispielsweise Unterabfragen benötigen, ein. Daher wurde die Anfragesprache SQBE, die eine orthogonale Erweiterung zu QBE bildet, vorgestellt. Sie ist streng relational vollständig und unterstützt Unterabfragen. Dabei setzt sich eine SQBE-Anfrage aus einzelnen QBE-Anfragen zusammen. Die Zusammenfassung der Möglichkeiten und Vorteile der Formulierung von Anfragen in diesen beiden Sprachen rundete den Abschnitt ab.

Daran anschließend wurden die Grundprinzipien von WS-QBE skizziert. Die Anfrageformulierung basiert auf dem Ausfüllen von Tabellenskeletten durch Beispielvariablen und Konstanten. Da WS-QBE als Multimedia-Anfragesprache konzipiert wurde, dürfen neben Zeichenketten und Zahlen auch Medien- und Multimedia-Objekte als Konstanten verwendet werden. Darüber hinaus besteht in WS-QBE die Möglichkeit, nicht nur Anfragekriterien zu formulieren sondern auch Nutzerpräferenzen durch Einsatz von Gewichten zu spezifizieren. Bei einfachen Anfragen kann eine Relevanz- und Schwellwertgewichtung vorgenommen werden. Dafür werden den jeweiligen Selektionsbedingungen entsprechende Relevanzvariablen zugeordnet, denen in der Gewichtungstabelle entsprechende Werte zugewiesen werden. Eine Schwellwertgewichtung kann sowohl für einzelne Selektionsbedingungen als auch für die ganz Anfrage spezifiziert werden. Bei komplexen Anfragebedingungen, die im Condition-Tree formuliert werden, kann zusätzlich zu diesen beiden Möglichkeiten der Gewichtung der Universaljunktor verwendet und parametrisiert werden. Damit setzt die Anfragesprache WS-QBE das in Kapitel 4 entwickelte Multi-Level-Gewichtungsmodell um.

Ein weiteres Charakteristikum von WS-QBE wird durch die Spezifizierung von Ausgabebedingungen beschrieben. Diese können sowohl zeitlicher als auch räumlicher Natur sein. Damit finden neben den multimediaspezifischen Anforderungen auch die in Abschnitt 3.1.3 beschriebenen Anforderungen an die Ergebnispräsentation Berücksichtigung.

Ferner wurde die Syntax und die Semantik von WS-QBE festgelegt. Da es sich um eine visuelle Anfragesprache handelt, wurde zunächst die Überführung der visuellen Sprachmittel auf entsprechende textuelle Konstrukte beschrieben. Für diese Textkonstrukte wurde eine formale Syntax in einer erweiterten Backus-Naur-Form angegeben. Es wurde ferner die konkrete Semantik der verwendeten Operationen und Junktoren beschrieben. Die Semantik der Junktoren basiert auf Fuzzy-Logik. Damit können neben unscharfen Anfragen auch Boolesche Anfragen adäquat ausgewertet werden. Weiterhin wurde ein Algorithmus zur Überführung eines textuellen WS-QBE-Ausdrucks in einen entsprechenden Ausdruck des Ähnlichkeitskalküls entwickelt. Diese Abbildung wurde anhand eines Beispiels verdeutlicht.

Die Sprache WS-QBE erlaubt es, Multimedia-Anfragen auf eine deklarative Art und Weise zu formulieren. Durch Verwendung einer grafischen Oberfläche steht eine nutzerfreundliche Schnittstelle zur Spezifikation von Anfragen bereit. Die Verarbeitung dieser Anfragen ist in Abbildung 6.5 dargestellt.

WS-QBE-Anfragen werden demnach zunächst auf einen Ausdruck des Ähnlichkeitskalküls abgebildet, welcher wiederum in einen Ausdruck der Ähnlichkeitsalgebra überführt wird. Aufbauend auf diesem Algebraausdruck kann eine logische Optimierung stattfinden.

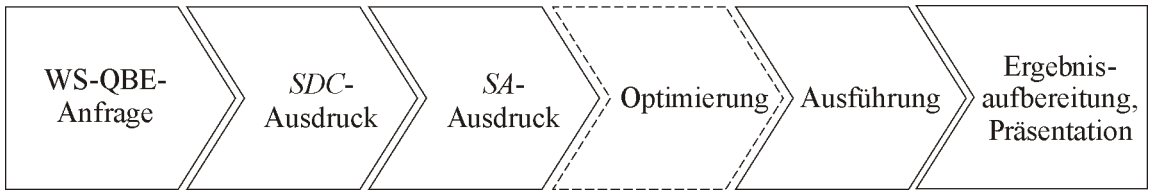


Abbildung 6.5: Verarbeitung einer WS-QBE-Anfrage

Unter Berücksichtigung einer möglichen physischen und kostenbasierten Optimierung wird der optimale Zugriffsplan ausgewählt. Dieser wird letztendlich ausgeführt. Die Optimierung ist nicht Gegenstand dieser Arbeit, jedoch wird im Ausblick dieser Arbeit auf die algebraische Optimierung noch einmal näher eingegangen (siehe Abschnitt 8.2.1). In der Präsentationskomponente des Anfragesystems werden die Ergebnisse entsprechend den in WS-QBE formulierten Ausgabebedingungen aufbereitet. Aus diesem Grund werden die zeitlichen und räumlichen Ausgabebedingungen nicht auf das Ähnlichkeitskalkül überführt. Vielmehr werden sie als Parameter für die Ergebnisaufbereitung verwendet.

7 Implementierung und Evaluierung

7.1 Implementierung

In diesem Abschnitt werden die verschiedenen Implementierungen vorgestellt, die im Rahmen dieser Arbeit realisiert wurden. Das Ziel, welches mit der prototypischen Implementierung der einzelnen Komponenten verfolgt wird, besteht in der experimentellen Validierung der verschiedenen theoretischen Modelle.

Abbildung 7.1 gibt einen Überblick über die prototypisch implementierten Komponenten des Systems. Dazu zählt die Umsetzung der in Kapitel 4 entwickelten Transformationsregeln für Anfragen mit Relevanzgewichten. Ferner existiert eine Umsetzung des in Kapitel 5 eingeführten Ähnlichkeitskalküls, der Ähnlichkeitsalgebra sowie der entsprechenden Überführung ineinander. Eine Ähnlichkeitssuche unter Verwendung von Gewichten wird in dem Anwendungsszenario LostArt realisiert. Ebenfalls im Kontext von LostArt ist die Formulierung räumlicher Anfragen möglich. Letztlich wurde die Anfragesprache WS-QBE prototypisch umgesetzt.

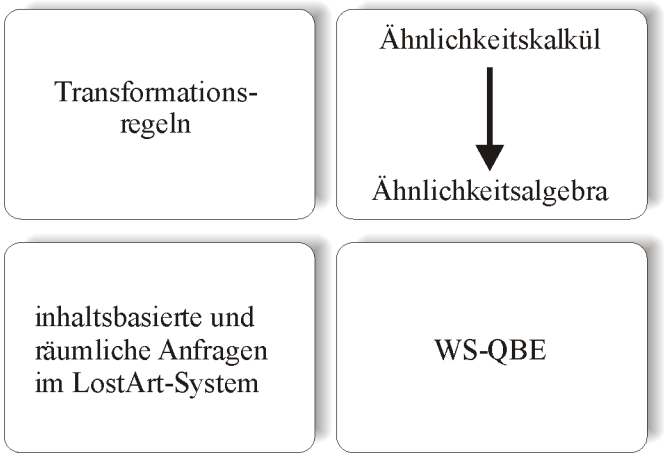


Abbildung 7.1: Überblick über prototypisch implementierte Komponenten

7.1.1 Transformationsregeln

Das Ziel der Implementierung von Transformationsregeln für gewichtete Anfragen liegt in der Validierung der Korrektheit dieser. Insbesondere ist hierbei die Überprüfung der Regeln für die Assoziativität und Distributivität von Bedeutung. Diese Aufgabe wird durch ein Java-Programm realisiert, welches anhand generierter Relevanzwerte und Gewichte aus dem Intervall $[0, 1]$ die in Abschnitt 4.3.3 entwickelten Transformationsregeln verifiziert.

Dafür wird eine erzeugte Formel in eine syntaktisch äquivalente Formel transformiert. Unter Beachtung der generierten Relevanzwerte und -gewichte werden die entsprechenden Transformationsregeln auf die Gewichtung angewendet. Anschließend wird sowohl für die Ausgangsformel als auch für die transformierte Formel der Gesamtrelevanzwert berechnet. Diese werden sodann auf Gleichheit getestet.

Die in Abschnitt 4.3.3 aufgeführten Transformationsregeln, die gegebenenfalls eine Modifizierung der Relevanzgewichte erforderlich machen, können somit auch experimentell bestätigt werden.

7.1.2 Ähnlichkeitskalkül und -algebra

Die Umsetzung des Ähnlichkeitskalküls und der Ähnlichkeitsalgebra erfolgt entsprechend der in Kapitel 5 beschriebenen Syntax und Semantik[1]. Als Implementierungsgrundlage für die Überführung eines Kalkülausdrucks in einen entsprechenden Algebraausdruck dient der in Abschnitt 5.5.1 beschriebene Reduktionalgorithmus. Die Realisierung wird zunächst nur für scharfe Relationen vorgenommen. Das Ziel liegt in der experimentellen Bestätigung der Äquivalenz von Kalkül und Algebra.

In dem Ähnlichkeitskalkül und der -algebra ist die Semantik der Junktoren und Operatoren nicht vordefiniert. Das klassische Fuzzy-Modell bildet die Grundlage für die Implementierung. Neben unscharfen und gewichteten Anfragen können damit weiterhin alle klassischen Booleschen Anfragen umgesetzt werden. Der Junktor $\wedge$ wird auf die T-Norm *min* und der Junktor $\vee$ auf die T-Conorm *max* abgebildet. Dem Universaljunktor liegt der ANDOR-Operator zu Grunde. Für die Umsetzung der Relevanzgewichtung wird die Gewichtungsformel verwendet. Der Wertebereich entspricht dem Bereich der ganzen Zahlen. Es werden verschiedene Operatoren, wie beispielsweise $=, >, <$ und $\sim$, realisiert. Der Ähnlichkeitsoperator $\sim$ bestimmt den auf $[0, 1]$ normierten Betrag der Differenz der beiden Operanden.

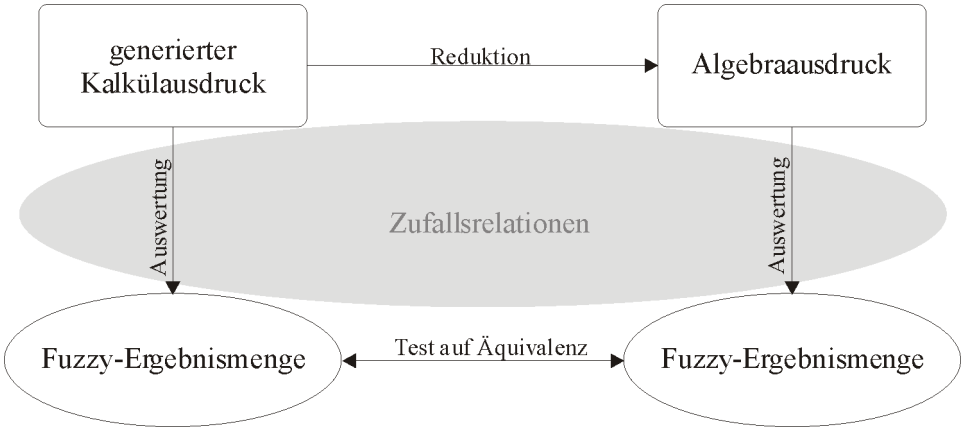


Abbildung 7.2: Test auf Äquivalenz zwischen Kalkül- und Algebraausdruck

In Abbildung 7.2 ist der Ablauf der experimentellen Untersuchungen für den Test auf

[1]Mein besonderer Dank gebührt Dr. Ingo Schmitt, welcher einen erheblichen Anteil der Implementierung der in diesem Abschnitt geschilderten Komponente trägt.

Äquivalenz zwischen $\mathcal{SDC}$ und $\mathcal{SA}$ dargestellt. Es werden zufällige, sichere $\mathcal{SDC}$-Ausdrücke generiert. Darauf aufbauend erfolgt die Reduktion der Kalkülausdrücke auf einen Ausdruck der Ähnlichkeitsalgebra. Sowohl die Kalkül- als auch die Algebraausdrücke werden ausgewertet. Dafür werden Relationen mit beliebigen Werten aus dem Bereich der ganzen Zahlen erzeugt. Die berechneten Ergebnisse werden letztlich auf Gleichheit getestet.

Ferner ist es möglich, konkrete Kalkülausdrücke vorzugeben. Diese werden dann in einen entsprechenden Algebraausdruck überführt und ausgewertet. Die Vorgehensweise wird im Folgenden an einem Beispiel gezeigt. Aus Gründen der leichteren Verständlichkeit wird dieses anhand einer sehr einfachen Booleschen Anfrage demonstriert, die den maximalen Wert für das Attribut $A1$ aus der Relation $R1(A1, A2, A3)$ ermittelt. Diese Anfrage kann wie folgt als Kalkülausdruck angegeben werden:

$$\{X1|(\exists X3(\exists X2(R1(X1, X2, X3) \wedge (\neg(\exists X7(\exists X8(\exists X9(R1(X7, X8, X9) \wedge (X7 > X1)))))))))\}.$$

Die mit Zufallszahlen erzeugte Relation $R1$, der vorgegebene $\mathcal{SDC}$-Ausdruck, der daraus generierte Algebraausdruck sowie die Ergebnisse dieser beiden Ausdrücke sind in Abbildung 7.3 so dargestellt, wie sie vom umgesetzten Programm ausgegeben werden.

```
R1=(A1,A2,A3)
Relation r1:
3 0 4
3 2 2
4 0 0
0 1 0
1 1 1

SDC-Expression:
(exists X3 (exists X2 (R1(X1,X2,X3) land (neg (exists X7 (exists X8 (exists X9 (R1(X7,X8,X9) land
(X7 > X1)))))))) [X1]

Result:
4   [1.0]

SA-Expression:
Pi_#1(Pi_#1,#3(Pi_#1,#2,#3((Pi_#1,#2,#3(R1(D1,D1,D1)) join #1=#1 (Pi_#1(R1(D1,D1,D1)) minus
Pi_#1(Pi_#1,#2(Pi_#1,#2,#3(Pi_#4,#1,#2,#3((Pi_#1,#2,#3(R1(D1,D1,D1)) join #1=#2
Sigma_{#2 > #1}((Pi_#1(R1(D1,D1,D1)) cross (Pi_#1(R1(D1,D1,D1)) union NULL))))))))))))

Result:
4   [1.0]
```

Abbildung 7.3: Reduktion anhand eines Beispiels

7.1.3 Gewichtete Ähnlichkeitssuche

Eine weitere Komponente für eine gewichtete Ähnlichkeitssuche ist in dem Anwendungsszenario LostArt implementiert. Bei LostArt handelt es sich um eine Internetdatenbank, die eine textbasierte Recherche nach kriegsbedingt verbrachten Kulturgütern unterstützt. Dazu zählen nicht nur Objekte der Malerei, sondern unter anderem auch Objekte der Schnitzkunst und Glaskunst sowie Teppiche und Waffen. Neben textuell beschreibenden Informationen werden – sofern vorhanden – Abbildungen der Kulturgüter in der Datenbank verwaltet. Insgesamt umfasst die Datenbank circa 3.500 Abbildungen.

Das wesentliche Ziel der Implementierung einer Ähnlichkeitssuche in diesem Kontext liegt in der Schaffung einer Suchoption, die die Suche nach Doubletten erleichtert und gegenüber Schreibfehlern und widersprüchlichen Beschreibungen unabhängig ist. Ferner stellt die Erprobung von Relevanz- und Schwellwertgewichten einen Schwerpunkt dar. Die Daten des LostArt-Systems werden in einer Oracle8i-Datenbank verwaltet. Daher wird für die Ähnlichkeitssuche das *Visual-Information-Retrieval-Cartridge* verwendet. Es ist der Vorgänger des *interMedia-Paketes*. Eine Ähnlichkeitsanfrage ist bezüglich der Features *lokale* und *globale Farbe*, *Form* und *Textur* möglich. In dem Prototyp werden Ähnlichkeitsanfragen auf Bildern unter Verwendung des Query-By-Example-Prinzips umgesetzt.

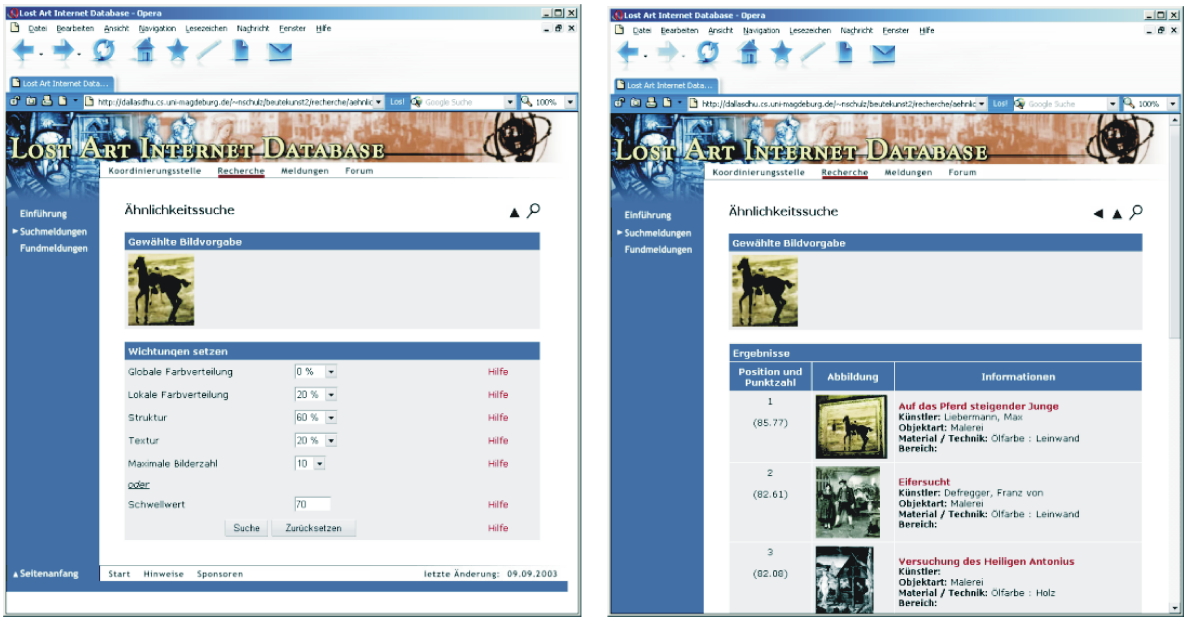


Abbildung 7.4: Screenshots des LostArt-Prototypen: Ähnlichkeitssuche anhand eines Bildausschnittes unter Berücksichtigung der spezifizierten Gewichte (links) und Darstellung der Ergebnisse mit dem gesuchten Bild an erster Stelle (rechts)

Die Ähnlichkeitssuche ist als Webschnittstelle implementiert und wurde für Test- und Vorführungszwecke in das System LostArt eingebettet. Die Realisierung der Webanwendung und die Anbindung an die Datenbank erfolgt mittels *PHP*. Abbildung 7.4 zeigt zwei Screenshots des Prototypen. Links wird dargestellt, wie eine Anfrage durch den Nutzer formuliert werden kann. Dafür gibt der Nutzer ein Bild vor und setzt die Relevanzgewichte entspre-

chend seiner Präferenzen. Die Anzahl der Ergebnisobjekte kann entweder durch Angabe eines Schwellwertes oder durch Vorgabe der Ergebnisanzahl eingeschränkt werden. Rechts ist das Ergebnis dieser Anfrage dargestellt.

Die Formulierung von Ähnlichkeitsanfragen in dem LostArt-Prototyp hat gezeigt, dass sowohl die Relevanzgewichtung als auch die Schwellwertgewichtung ein nützliches Instrumentarium für die Abbildung der Nutzerpräferenzen darstellen. In diesem einfachen Anwendungsszenario werden Schwellwertgewichte zum Eingrenzen der Ergebnismenge verwendet. Sie sind somit für die Gesamtanfrage gültig. Ferner wird deutlich, dass die Suche nach ähnlichen Objekten bei der Wahl der Features ein entsprechendes Wissen des Nutzers über die Datenbank erforderlich macht. So werden in der LostArt-Datenbank vornehmlich Schwarz-Weiss-Abbildungen verwaltet. Damit ist die Verwendung des Features Farbe für die Ähnlichkeitssuche nicht sonderlich hilfreich. Im Allgemeinen sollte jedoch nicht davon ausgegangen werden, dass der Nutzer über Hintergrundwissen über die Datenbankobjekte verfügt. Um ihn dennoch bei der Formulierung von Anfragen zu unterstützen, können beispielsweise Nutzerprofile verwendet werden. Ferner wird deutlich, dass der Einsatz von Relevanz-Feedback-Mechanismen weiterhin notwendig ist.

7.1.4 Räumliche Anfrageformulierung

Eine weitere Komponente unterstützt räumliche Anfragen in dem LostArt-System (Urban, 2003). Die grobe Architektur ist in Abbildung 7.5 dargestellt.

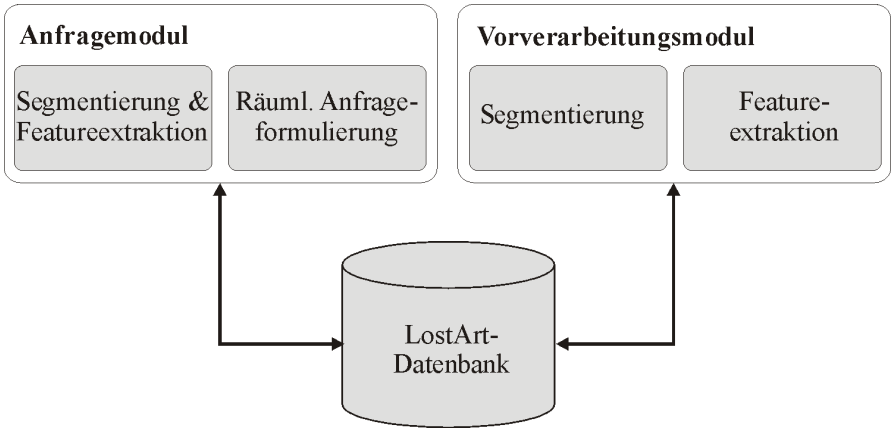
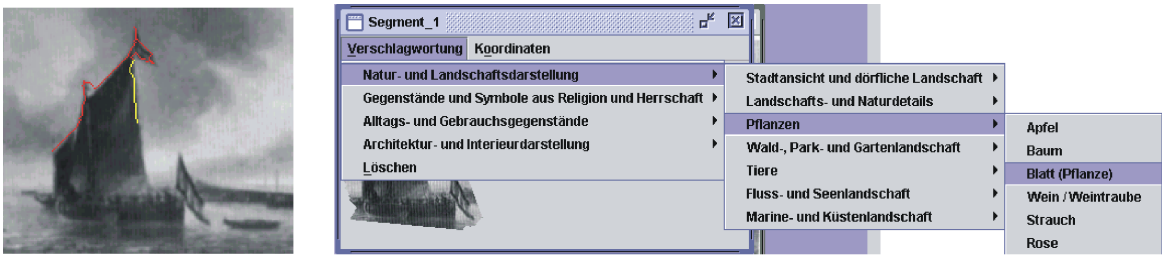


Abbildung 7.5: Grobarchitektur der Komponente für räumliche Anfragen

Für die Gewährleistung räumlicher Suchanfragen ist zunächst eine Aufbereitung der Daten notwendig. Für diesen Zweck wird ein Vorverarbeitungsmodul implementiert. Dieses untersützt die Segmentierung von Salient-Objekten innerhalb eines Bildes. Da kantenbasierte Segmentierungsverfahren bessere Ergebnisse liefern als regionen- oder punktbasierte Verfahren wird das *Life-Wire*-Verfahren (Mortensen und Barrett, 1995) eingesetzt. Dadurch kann eine semi-automatische Segmentierung realisiert werden. Dieses ist in Abbildung 7.6a dargestellt. Jedes Salient-Objekt wird dann entsprechend einer vorgegebenen Hierarchie mit einem Schlagwort versehen (siehe Abbildung 7.6b). Die für jedes Salient-Objekt extrahierten Features werden zusammen mit den Schlagworten in der Datenbank gespeichert. Letztlich

werden für alle Salient-Objekte eines Bildes die Lagebeziehungen bestimmt. Diese werden in einer modifizierten Form des 2D-C-Strings (vgl. Abschnitt 3.1.2) abgespeichert.



a) b)

Abbildung 7.6: a) Segmentierung und b) Zuordnung eines Schlagwortes zu dem Salient-Objekt

Die Formulierung der räumlichen Anfragen wird in einem Anfragemodul implementiert. Auch hier können Bilder vom Nutzer segmentiert werden, um sie so für die Anfrage vorzubereiten. Zur Formulierung der Anfrage werden Objekte auf einer Fläche angeordnet (siehe Abbildung 7.7). Diese können frei bewegt, vergrößert, verkleinert und ebenfalls verschlagwortet werden. Die Schlagworte dienen dann als Suchkriterien. Die räumlichen Lagebeziehungen werden auf Basis des vom Nutzer erstellten Layouts ermittelt. Berechnete Lagebeziehungen bilden die Basis für die Suche nach Bildern in der Datenbank, welche die gleichen Lagebeziehungen und Schlagworte aufweisen. Mit diesem Modul kann die Formulierung komplexer räumlicher Anfragen im Condition-Tree von WS-QBE umgesetzt werden.

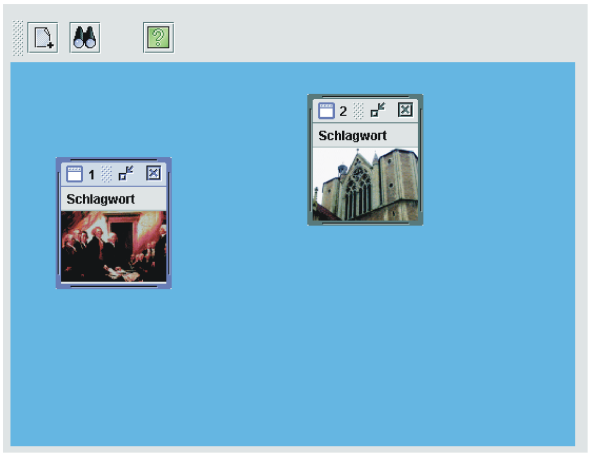


Abbildung 7.7: Screenshot des Moduls zur räumlichen Anfrageformulierung

7.1.5 Anfrageformulierung mit WS-QBE

Die Anfragesprache WS-QBE ist in einem Prototyp umgesetzt. Das Ziel dieser Implementierung liegt in der Entwicklung eines entsprechenden Nutzerinterfaces und der teilweisen

Umsetzung der in Abschnitt 6.3 beschriebenen Funktionalität von WS-QBE. Derzeitig wird durch den Prototypen die Formulierung einfacher gewichteter Anfragen unterstützt.

Der Prototyp ist in Java implementiert. Als DBMS wird Oracle9i verwendet. Für die Ähnlichkeitsberechnung wird auf das *interMedia-Paket* aufgesetzt. Damit können für den Datentyp *Bild* die Features *Farbe*, *Form* und *Textur* unterstützt werden. Eine WS-QBE-Anfrage wird in atomare Teilanfragen zerlegt. Somit wird für jede Selektionsbedingung eine Anfrage an die Datenbank geschickt. Oracle liefert bei Ähnlichlichkeitsanfragen Distanzwerte zurück. Daher erfolgt vor einer weiteren Verarbeitung ein Mapping auf Ähnlichkeitswerte. Die in einer WS-QBE-Anfrage spezifizierten Ausgabebedingungen finden bei der Ergebnisaufbereitung Berücksichtigung.

Entsprechend der logischen Verknüpfung der einzelnen Teilanfragen werden die Teilergebnislisten im *Combiner* verknüpft. Combiner gewährleisten eine effiziente Aggregation der Ähnlichkeitswerte aus den Strömen für jede Teilanfrage (Fagin, 1996; Güntzer u. a., 2000; Nepal und Ramakrishna, 1999). Für die Aggregation der Ähnlichkeitswerte werden Scoring-Funktionen (vgl. Kapitel 4) eingesetzt. Hier wird auf der Faginschen Formel aufgesetzt. Einen guten Überblick zu den verschiedenen Combinern gibt die Arbeit von Henrich und Robbert (2003).

Die ermittelten Ergebnisobjekte werden sodann unter Beachtung der Präsentationsparameter aufbereitet und dem Nutzer präsentiert. Die grobe Architektur des Prototypen ist in Abbildung 7.8 dargestellt.

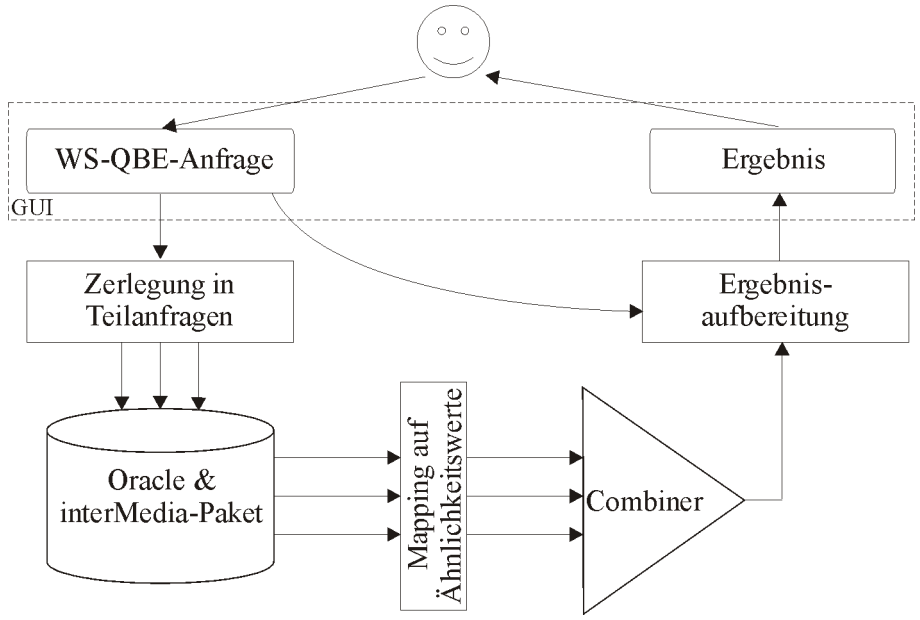


Abbildung 7.8: Grobarchitektur des Prototypen zur Anfrageformulierung in WS-QBE

Die Formulierung von Multimedia-Anfragen wird durch eine Auswahl an vordefinierten Farben, Formen und Texturen nutzerfreundlich unterstützt. Sobald mehr als zwei Gewichtsvariablen durch den Nutzer spezifiziert werden, wird eine Gewichtungstabelle *Weighting* aufgebaut. Werden weitere Gewichtsvariablen verwendet, so wird diese Gewichtungstabel-

le automatisch aktualisiert. Die initialen Gewichte sind alle gleich, so dass dieses dem ungewichteten Fall entspricht. Die konkreten Gewichte für die Selektionsbedingungen können dann vom Nutzer in der Gewichtungstabelle spezifiziert werden.

Die Abbildung 7.9 zeigt einen Screenshot vom Prototypen. Es wird eine einfache gewichtete Anfrage formuliert. Dabei soll das Bild einem vorgegeben Bild bezüglich der Farbe ähnlich sein. Das Bild wird vom Nutzer ausgewählt. Es erscheint in Form eines Thumbnails als Konstante in der Tabelle. Zusätzlich dazu wird das Suchbild dem Nutzer angezeigt. Die Anzeige kann bei Bedarf wieder geschlossen werden. Neben der Farbähnlichkeit wird vom Nutzer als zweites Suchkriterium angegeben, dass der Name des gesuchten Bildes wahrscheinlich 'Strand' ist. In der Gewichtungstabelle werden für diese beiden Suchbedingungen Relevanzgewichte spezifiziert. Es ist zu sehen, dass eine Übereinstimmung mit dem Namen weniger relevant ist als die gewünschte Bildähnlichkeit.

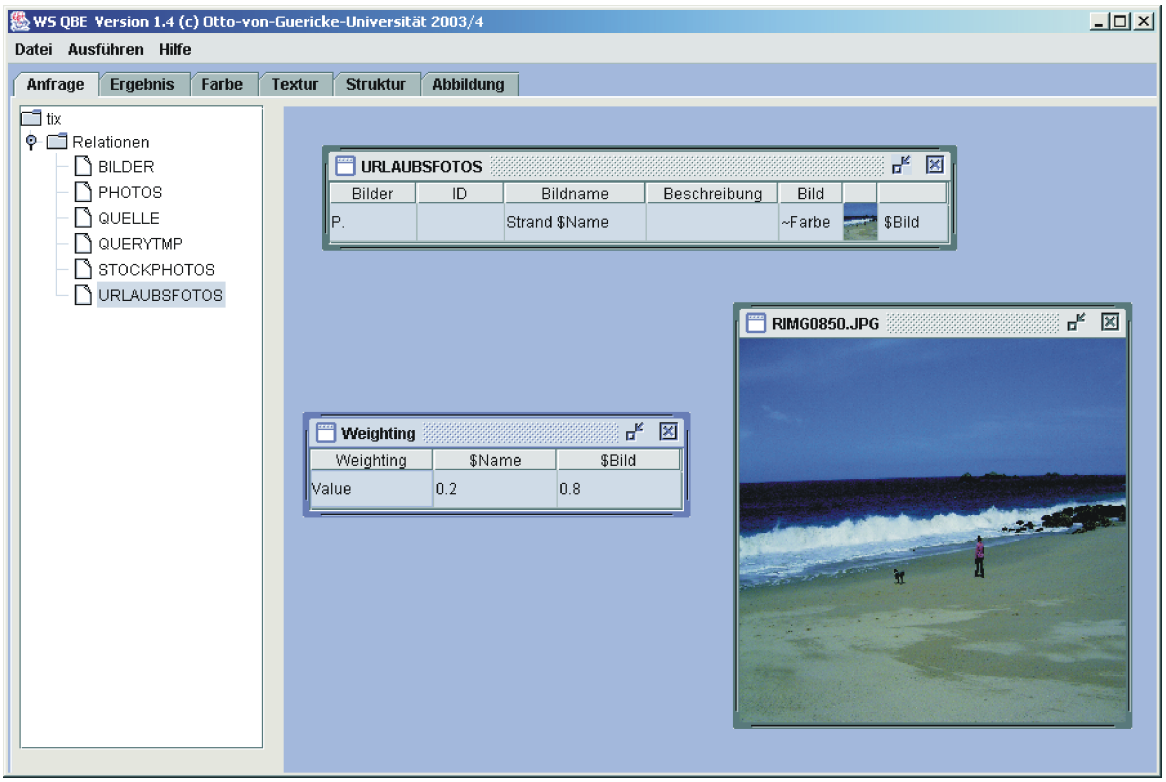


Abbildung 7.9: Screenshot der Anfrageformulierung in WS-QBE

7.2 Evaluierung der Anfragesprache

In diesem Abschnitt erfolgt eine Evaluierung der Multimedia-Anfragesprache WS-QBE. In Kapitel 3 wurden die grundlegenden Prinzipien, die bei der Entwicklung einer Multimedia-Anfragesprache zu berücksichtigen sind, vorgestellt. An dieser Stelle soll nun die Sprache WS-QBE hinsichtlich der Erfüllung dieser Anforderungen untersucht werden.

Zu Beginn wird dafür der Einsatz von WS-QBE anhand von Beispielanfragen aus dem Kontext von LostArt[2] demonstriert: Eine einfache Anfrage ist die Suche nach Informationen zu einem Objekt, zu dem lediglich ein Foto existiert. Weitere Detailinformationen sind nicht bekannt. Auf Grund des Fotos kann die Objektart als Plastik klassifiziert werden. Dadurch kann eine einfache Anfrage beispielsweise in den Suchmeldungen von LostArt folgendermaßen in WS-QBE formuliert werden:

Suchdaten	Id	Künstler	Titel	Objektart
P.	_id			Plastik

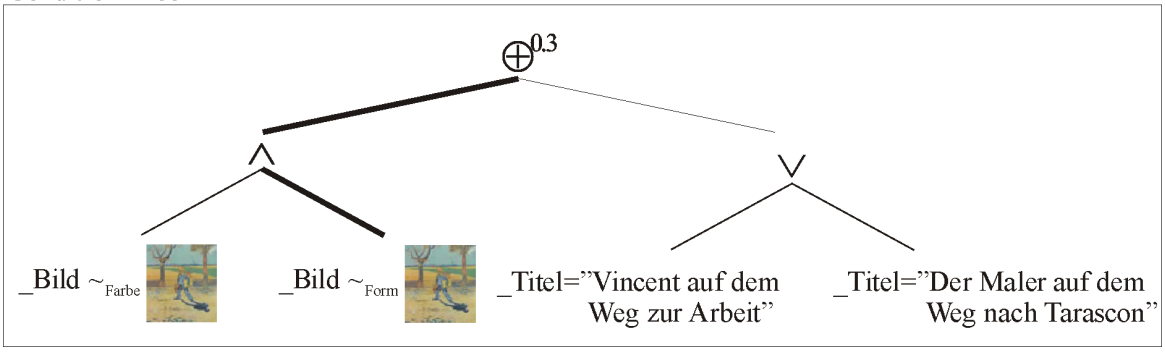
Abbildung	BId	Bild
	_id	~

Eine komplexere Anfrage im Anwendungsszenario LostArt stellt beispielsweise der Abgleich von Fund- und Suchdaten dar. Wird ein Objekt als *vermisst* gemeldet, so kann nach diesem in den Fundmeldungen recherchiert werden. Als Beispiel wäre hier die Suche nach einem Werk von Vincent van Gogh zu nennen. Es ist bekannt, dass es sich bei dem gesuchten Werk um ein Gemälde handelt. Das Gemälde kann zwei unterschiedliche Titel aufweisen. Darüber hinaus existiert eine alte Aufnahme, so dass eine inhaltsbasierte Anfrage formuliert werden kann. Eine Gewichtung drückt aus, dass zum einen die Ähnlichkeit bezüglich der Struktur des Bildes wichtiger ist als bezüglich der Farbverteilung. Ferner ist es wichtiger, dass eine Ähnlichkeit mit dem vorgegebenen Bild vorhanden ist, als dass eine Übereinstimmung mit den textuellen Suchkriterien existiert. Die Suchkriterien werden mit einem reglementierten *ODER* (z=0.3) verknüpft. In WS-QBE kann eine entsprechende Anfrage wie folgt formuliert werden:

Funddaten	Id	Künstler	Titel	Objektart
P.	_id		_Titel	Malerei

Abbildung	BId	Bild
	_id	_Bild

Condition-Tree



7.2.1 Allgemeine Anforderungen

Die allgemeinen Anforderungen, die bei der Entwicklung einer jeden Anfragesprache Berücksichtigung finden müssen, wurden bereits in Abschnitt 3.1.1 beschrieben. Nach Saake u. a. kann keine Anfragesprache alle Kriterien gleich gut erfüllen, da die spezifizierten Anforderungen zum Teil gegensätzliche Ziele verfolgen (Saake u. a., 1997, S. 236). Bei der Ent-

[2]Die verwendeten Abbildung stammen aus der über http://www.lostart.de öffentlich zugängigen LostArt-Datenbank.

wicklung von WS-QBE wurde daher auf einige Anforderungen besonders Wert gelegt, auf andere Anforderungen jedoch weniger. Im Folgenden wird für jede einzelne Anforderung untersucht, ob und inwiefern sie von der Multimedia-Anfragesprache WS-QBE erfüllt wird:

Ad-hoc-Formulierung; WS-QBE bietet eine nutzerfreundliche Ad-hoc-Formulierung von Anfragen über eine grafische Benutzerschnittstelle an. Neben der Vorgabe von textbasierten Anfragekonstrukten werden auch andere Medien unterstützt. Wird beispielsweise ein Bild als Konstante vom Nutzer gewählt, so wird dieses als Miniatur angezeigt. Ferner wird die Wahl spezieller Features, wie Farbe oder Textur, erleichtert, indem der Nutzer aus einer Palette vordefinierter Farben und Texturen wählen kann.

Generische Operatoren; Die Multimedia-Anfragesprache WS-QBE basiert auf wenigen generischen Operationen. Damit wird für semantisch gleiche Operationen auf verschiedenen Objekten die gleiche Syntax verwendet. So wird beispielsweise der Ähnlichkeitsvergleich $\sim$ für alle Medien-Typen, aber auch für numerische und alphanumerische Datentypen unterstützt.

Anwendungsunabhängigkeit; Der Entwurf der Sprache WS-QBE erfolgte unabhängig von einem bestimmten Anwendungsszenario. Bei der Entwicklung wurde darauf geachtet, dass diverse Medientypen (Video-, Audio-, Bild- und Text) unterstützt werden können. Damit kann WS-QBE universell eingesetzt werden.

Deskriptivität; WS-QBE wurde als deskriptive Anfragesprache entworfen. Die entwickelten Sprachkonstrukte sowohl für die Anfrageformulierung als auch für die Beschreibung der Ergebnispräsentation drücken lediglich die Struktur und den Inhalt aus. Es muss vom Nutzer nicht angegeben werden, wie die Anfrage auszuwerten beziehungsweise wie die Ergebnispräsentation zu erstellen ist.

Mengenorientiertheit; Die Anfragesprache WS-QBE arbeitet auf Fuzzy-Mengen. Damit können unscharfe Multimedia-Anfragen unterstützt werden. Die klassischen Mengenoperationen wurden für Fuzzy-Mengen erweitert.

Orthogonalität; Orthogonalität fordert, dass Operationen beliebig kombinierbar sind. Derzeitig weist WS-QBE diesbezüglich noch Erweiterungspotenzial auf. Um eine größtmögliche Orthogonalität bei der Formulierung von WS-QBE-Anfragen zu erreichen, sollten die Konzepte von SQBE aufgegriffen und auf WS-QBE angewendet werden.

Effizienz; Die Operationen sollen effizient realisierbar sein. Dies ist letztendlich davon abhängig, wie die Operationen konkret umgesetzt werden. In der prototypischen Implementierung werden beispielsweise effiziente Combiner für die Aggregation der Ergebnislisten verwendet. Für die Ähnlichkeitsberechnung wird jedoch auf Oracle aufgesetzt, so dass diesbezüglich keine Aussage getroffen werden kann.

Prinzipiell kann WS-QBE um benutzerdefinierte Methoden erweitert werden. So können beispielsweise weitere Ähnlichkeitsoperatoren oder aber auch Fuzzy-Prädikate integriert werden. Die Effizienz ist somit natürlich je nach Implementierung unterschiedlich. Ferner kann im Rahmen der Ergebnispräsentation in WS-QBE durch die Angabe von Print-Optionen, wie zum Beispiel der Größe des auszugebenden Medien-Objektes, die Effizienz des Systems gesteigert werden.

Erweiterbarkeit; In WS-QBE können Erweiterungen des zu Grunde liegenden Datenmodells berücksichtigt werden. Werden zum Beispiel neue Features in das Datenmodell aufgenommen, so können entsprechende Vergleichsoperatoren definiert werden.

Abgeschlossenheit; Das Ergebnis einer WS-QBE-Anfrage ist eine Relation mit einem Attribut für den Ähnlichkeitswert. Für ursprünglich scharfe Relationen wird ein künstliches Attribut für den Relevanzwert erzeugt. Damit werden sie in Fuzzy-Relationen überführt. Das Ergebnis einer WS-QBE-Anfrage ist somit im relationalen Datenbankmodell darstellbar. Eine in WS-QBE erzeugte Ergebnisrelation kann als Eingabe für nachfolgende Anfragen verwendet werden.

Adäquatheit; WS-QBE verfügt über Konstrukte zur Formulierung der zeitlichen und räumlichen Beziehungen sowie zur Formulierung unscharfer, inhaltsbasierter und gewichteter Anfragen. Dadurch, dass WS-QBE erweiterbar ist, können möglicherweise fehlende Operationen hinzugefügt werden. Somit wird Adäquatheit gewährleistet.

Vollständigkeit; WS-QBE ist relational vollständig. Die WS-QBE-Anfragen werden auf einen Ausdruck des Ähnlichkeitskalküls überführt, die dann auf eine Ähnlichkeitsalgebra abgebildet werden. Diese wiederum ist eine reine Erweiterung der Relationenalgebra (vgl. Kapitel 5). WS-QBE ist, wie auch QBE, nicht streng relational vollständig. Dies kann jedoch durch Anwendung der Konzepte von SQBE auf WS-QBE erreicht werden.

Sicherheit; Jede WS-QBE-Anfrage wird in das Ähnlichkeitskalkül überführt. Dort erfolgt die Prüfung auf Sicherheit (vgl. Abschnitt 5.3.2). Ist diese Prüfung erfolgreich, dann werden die Anfragen ausgeführt. Damit kann gewährleistet werden, dass WS-QBE-Anfragen terminieren und ein endliches Ergebnis liefern.

Optimierbarkeit; Eine WS-QBE-Anfrage wird auf einen Ausdruck der Ähnlichkeitsalgebra überführt. Hierauf können Optimierungsregeln angewendet werden. Ferner besteht die Möglichkeit, bereits den initialen WS-QBE-Ausdruck vor Überführung in einen $\mathcal{SDC}$-Ausdruck zu vereinfachen. Beispielsweise können Anfragebedingungen, denen ein Relevanzgewicht von *null* zugeteilt wird, von der Anfrage entfernt werden.

Formale Semantik; Die formale Semantik der Multimedia-Anfragesprache WS-QBE wird durch die Abbildung auf das Ähnlichkeitskalkül beschrieben.

Eingeschränktheit; WS-QBE ist nicht berechnungsvollständig.

7.2.2 Multimediaspezifische Anforderungen

Neben den allgemeinen Anforderung an Multimedia-Anfragesprachen stellen auch die in Abschnitt 3.1.2 aufgeführten multimediaspezifischen Anforderungen ein wichtiges Evaluierungskriterium dar. In diesem Sinne gilt es zu untersuchen, inwiefern WS-QBE Konzepte zur Formulierung inhaltsbasierter, unscharfer, räumlicher und zeitlicher Anfragen bereitstellt. Des Weiteren muss eine Spezifikation der Nutzerpräferenzen möglich sein.

WS-QBE stellt neben den klassischen Vergleichsoperatoren ($<, >, =, \neq, \geq, \leq$) auf numerischen und alphanumerischen Daten diverse Operationen für verschiedene Medien-Typen

bereit. Damit ist es möglich, sowohl inhaltsbasierte, unscharfe und räumliche als auch zeitliche Anfragen zu formulieren. Um eine nutzerfreundliche Formulierung zu gewährleisten, kann in WS-QBE eine räumliche Bedingung als Layout formuliert werden.

Darüber hinaus wird die Formulierung von Präferenzen unterstützt. Dies basiert auf dem Gewichtungsmodell. So können in WS-QBE Relevanzgewichte und Schwellwertgewichte für Anfragebedingungen sowie Gewichte zur Parametrisierung des Universaljunktors eingesetzt werden.

Eine Anfrageverfeinerung beziehungsweise -modifikation wird bisher nicht in WS-QBE angeboten. Das der Anfragesprache WS-QBE zu Grunde liegende Multi-Level-Gewichtungsmodell bietet dafür jedoch die notwendige theoretische Basis. Es gilt nun, entsprechende Feedback-Methoden zu entwickeln und in WS-QBE zu integrieren. Mögliche Ansätze werden im Ausblick dieser Arbeit kurz skizziert (siehe Abschnitt 8.2.2). Trotz dieses Defizits kann WS-QBE die anderen multimediaspezifischen Anforderungen, wie oben geschildert, gut erfüllt.

7.2.3 Anforderungen an die Präsentation

Schließlich gilt es, die Anforderungen bezüglich der Präsentation zu verifizieren. WS-QBE verfügt über Sprachmittel, welche die Parametrisierung der Präsentation der Ergebnisobjekte zulassen. Dies wird mittels einer Option für den Kontrolleintrag **P.** realisiert (vgl. Abschnitt 6.3.4). Damit kann der Forderung nach einer parametrisierbaren Ergebnispräsentation nachgekommen werden.

7.2.4 Fazit

Die Entwicklung der Anfragesprache WS-QBE erfolgte unter Berücksichtigung der Prinzipien für den Entwurf einer Multimedia-Anfragesprache. Aus diesem Grund werden die grundlegenden Anforderungen an eine Multimedia-Anfragesprache erfüllt. So gilt für WS-QBE:

- Es ist universell einsetzbar, da prinzipiell alle gängigen Medientypen (Audio, Video, Bild, Text) unterstützt werden können.
- Es erlaubt die Formulierung verschiedenster Multimedia-Anfragen.
- Es unterstützt die Abbildung von Nutzerpräferenzen.
- Es bietet die Möglichkeit zur Parametrisierung der Ergebnispräsentation.
- Es wird durch eine formale Semantik beschrieben.

Keine Berücksichtigung bei der Entwicklung der Sprache fanden Funktionen. Hier ist eine Erweiterung von WS-QBE wünschenswert. Zu beachten ist dabei, dass sowohl für numerische und alphanumerischen Datentypen als auch für die verschiedenen Medientypen Funktionen bereitgestellt werden. Um eine größtmögliche Orthogonalität sowie eine strenge relationale Vollständigkeit zu erreichen, kann WS-QBE um Konzepte von SQBE erweitert werden. Ferner sollten Mechanismen zur Modifikation der Anfrage in WS-QBE integriert werden.

Weitere Potenziale liegen in der Erweiterung der Sprachmittel. So ist es beispielsweise möglich, parametrisierbare Quantoren und weitere Schwellwertoperatoren in WS-QBE zu integrieren. Diese Funktionalität wird bereits durch das Ähnlichkeitskalkül bereitgestellt. Um jedoch WS-QBE zunächst einfach zu halten, wurden diesen Möglichkeiten bis jetzt nicht berücksichtigt.

7.3 Fallstudie zum Multi-Level-Gewichtungsmodell

Zur Evaluierung des Multi-Level-Gewichtungsmodells wird eine Fallstudie durchgeführt. Ziel der Studie ist die Klärung der Frage, ob die Abbildung von Nutzerpräferenzen tatsächlich einen Vorteil bringt. Die Evaluierung erfolgt exemplarisch unter Verwendung eines Bild-Retrieval-Systems. Es gilt, die folgende Hypothese zu überprüfen:

Hypothese: Die Abbildung zusätzlicher Nutzerpräferenzen durch Einsatz von Gewichten bei der Anfrageformulierung bringt im Vergleich zu ungewichteten Anfragen bessere Retrieval-Ergebnisse. Dieses wird durch bessere Precision- und Recall-Werte ausgedrückt.

Die Durchführung einer solchen Fallstudie ist nicht trivial (Smeaton und Over, 2002). Aus diesem Grund gibt es im Bereich des Multimedia-Retrievals nur sehr wenige Veröffentlichungen zum Thema. Erst seit 2001 gibt es beispielsweise die *TREC Video Retrieval Evaluation* des National Institute of Standards and Technology (NIST). Seit dem Jahr 2003 läuft die Video-Evaluierung unter dem Namen TRECVID. Sie ist seitdem unabhängig von TREC.

Die der Studie zum Multi-Level-Gewichtungsmodell zu Grunde liegende Methodik basiert auf der Herangehensweise der Evaluierung von Video-Retrieval-Systemen bei TREC, wie sie von Smeaton und Over (2002) veröffentlicht wurde. Ferner orientiert sie sich an den von Yin (1994) beschriebenen Schritten zur Durchführung von Fallstudien. Im Folgenden wird auf die Vorbereitung, die Durchführung und die Auswertung der im Rahmen dieser Arbeit durchgeführten Fallstudie zum Multi-Level-Gewichtungsmodell eingegangen (siehe Abbildung 7.10).

Abbildung 7.10: Schritte zur Durchführung einer Fallstudie

7.3.1 Vorbereitung

Die verwendete Datenkollektion, die Herleitung der natürlichsprachlichen Anfragen sowie die Vorgehensweise bei der Übersetzung dieser in entsprechende WS-QBE-Anfragen wird in diesem Abschnitt beschrieben.

Datenkollektion

Nach Smeaton und Over (2002) ist es wichtig, dass als Grundlage für die Datenkollektion öffentliches Material verwendet wird. Damit ist es möglich, dass die Fallstudie auch von Dritten nachvollzogen werden kann. Hier wird daher die *zefa29-zeitgeist*-Bildkollektion verwendet. Die darin enthaltenen 1288 Fotografien sind kategorisiert und für die Erstellung von Werbebroschüren und ähnlichen Materialien gedacht. Die Anzahl der Bilder pro Kategorie ist in der folgenden Tabelle 7.1 aufgeführt.

Kategorie	Anzahl Bilder
Zeitgeist	15
Growing up	168
Loving and Laughing	180
Living together	102
Eating and drinking	113
Getting Fit	78
Relaxing	58
Shopping	48
Working	187
Having Ideas	197

Tabelle 7.1: Kategorien der *zefa29-zeitgeist*-Bildkollektion

Hierbei ist es wichtig anzumerken, dass die Datenkollektion unabhängig vom Retrieval-System ist. Es wurden keine Features extrahiert, die speziell auf diese Kollektion zugeschnitten sind. Vielmehr wird, wie in Abschnitt 7.1.5 beschrieben, für die Umsetzung der Ähnlichkeitsberechnung auf die Funktionalität des *interMedia-Paketes* von Oracle zurückgegriffen.

Wahl der Anfragen

Natürlichsprachliche Anfragen bilden die Grundlage für die Prüfung der aufgestellten Hypothese. Dafür sollten nach Möglichkeit *typische* Multimedia-Anfragen verwendet werden. Bei der Wahl solcher Anfragen sind verschiedene Aspekte zu berücksichtigen: Zunächst stellt sich die Frage, was typische Anfragen an ein Multimedia-Retrieval-System sind. Die Beantwortung dieser Frage ist nach Smeaton und Over auf Grund fehlender Studien, Unterschiede in den Kollektionen oder auch verschiedener Eigenschaften des Suchenden derzeit völlig unklar. Eine weitere Frage ist, wie Multimedia-Prädikate am effektivsten und natürlichsten eingesetzt werden können. Auch dieses ist eine offene Fragestellung.

Smeaton und Over empfehlen, die Anfragen von echten Nutzern auf ihren Daten erstellen zu lassen. Oftmals ist dieses aber nicht möglich, da es an entsprechenden Nutzern und Systemen mangelt. Aus diesem Grund kann ein Experte, der mit der Testkollektion vertraut ist, die Anfragen formulieren.

Für die Studie zum Multi-Level-Gewichtungsmodell wird daher dieser Ansatz von TREC übernommen: Es wird rückwärts von der Testkollektion ausgehend gearbeitet. Die Anfragen werden von einem Experten formuliert, wobei ein bestimmtes Anfrageergebnis fokussiert wird. Die Vorgehensweise ist dabei wie folgt:

1. Die Kollektion betrachten.

2. Notizen über Zielobjekte machen. Dabei ist es wichtig, dass die fokussierten Beispiele sich in Farbe, Form sowie Textur ähnlich sind. Dieses abzuschätzen ist problematisch bis nahezu unmöglich, da es auch immer abhängig von den verwendeten Extraktionsmethoden ist.

3. Anfrage natürlichsprachlich formulieren.

Bei TREC wird zwischen generischen und spezifischen Anfragen, den sogenannten *Topics* unterschieden. Der folgende Auszug aus den Topics des Jahres 2002 zeigt, dass bei den spezifischen Topics (Topic 77) die Interpretation der Medien-Objekte im Vordergrund steht, während die generischen Topics (Topic 78) eher allgemeingültig sind (Voorhees und Buckland, 2002):

Topic 77: Find pictures of George Washington.

Topic 78: Find shots of people spending leisure time at the beach, for example walking, swimming, sunning, playing in the sand. Some part of the beach or buildings should be visible.

Für die hier durchgeführte Fallstudie dienen die folgenden generischen natürlichsprachlichen Anfragen als Grundlage:

A1: Finde Bilder mit einer Orange auf blauem Hintergrund.

A2: Finde Bilder mit Menschen auf einer grünen Wiese. Teile der Wiese sollen sichtbar sein.

A3: Finde Bilder mit einer Person im oder unter Wasser. Das Wasser ist kristallklar und die typischen Wasserkennzeichen, wie beispielsweise Blasen, Spritzer und Lichtreflektionen, sollen erkennbar sein. Außer der Person und dem Wasser soll möglichst nichts weiter im Bild zu sehen sein.

A4: Finde Bilder mit einer auf einer blau/weiß-karierten Tischdecke liegenden Apfelsine, Banane oder Flasche. Der jeweilige Gegenstand soll sich in der Mitte der Abbildung befinden.

Diese Anfragen beschreiben zunächst einen allgemeinen Suchwunsch. Tabelle 7.2 gibt einen Überblick über die verwendeten Anfragen. Dabei wird angegeben, wieviel Bilder in der Kollektion bezüglich jeder Anfrage relevant sind. Ferner wird dargestellt, bis zu wieviele Beispielelemente für die Anfrageformulierung verwendet werden dürfen. Als Beispielelement kann ein Bild oder ein Text verwendet werden.

Die natürlichsprachlichen Anfragen bilden die Basis für die Formulierung von entsprechenden Datenbank-Anfragen. Auf diesen Aspekt wird im Folgenden eingegangen.

Übersetzung der Anfragen

Generell stellt sich die Frage, wie eine natürlichsprachliche Anfrage in eine systemverständliche Datenbank-Anfrage umgesetzt werden kann. Nach Smeaton und Over ist eine automatische Umsetzung eines Suchwunsches in eine Datenbankanfrage sehr schwierig, so dass bei TREC die Anfragen zum Großteil manuell formuliert werden.

Nummer	Kurzform der Anfrage	Anzahl an Beispielen		Relevante Bilder
		Bild	Text	
A1	Orange	1	1	2
A2	Wiese	3		28
A3	Wasser	2		5
A4	Karodecke	2		4

Tabelle 7.2: Überblick über die natürlichsprachlichen Anfragen

In dieser Studie wird daher der allgemeine Suchwunsch von einem 'Übersetzer' interpretiert und in Form einer entsprechenden Datenbank-Anfrage *bestmöglich* formuliert. Dies ist natürlich sehr subjektiv. Der Prozess zur Erstellung der Datenbank-Anfragen ist in Abbildung 7.11 dargestellt. Bei der konkreten Formulierung der Anfragen können Nutzerpräferenzen, beispielsweise auf Grund des zur Verfügung stehenden Bildmaterials zur Formulierung der Anfragen, entstehen. So kann ein Anfragebild beispielsweise besonders gut die gewünschte Textur beschreiben, selbst wenn es für die Beschreibung der gewünschte Farbe weniger geeignet ist. In WS-QBE ist es möglich, diese Präferenzen in Form von Gewichten bei der Anfrageformulierung zu berücksichtigen. Aus diesem Grund formuliert der 'Übersetzer' in dieser Studie den Suchwunsch daher einmal mit und einmal ohne Präferenzen in Form von WS-QBE-Anfragen.

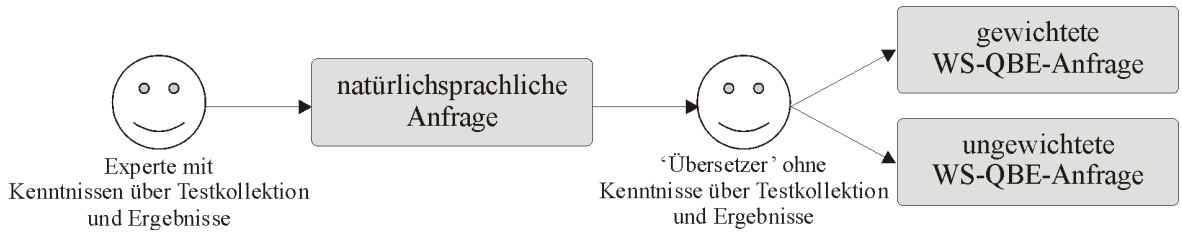


Abbildung 7.11: Erstellung und Übersetzung von Anfragen

Damit die Übersetzung der Anfragen optimal erfolgen kann, werden die folgenden Anforderungen an den 'Übersetzer' gestellt: Er muss sich mit dem Interface für die Formulierung der Anfrage auskennen, darf jedoch über kein Hintergrundwissen bezüglich der Testdatenmengen oder den Testergebnissen verfügen. Wird diese Übersetzung von einem einzigen 'Übersetzer' vorgenommen, dann ist ein Vergleich der Anfrageergebnisse in einem System gewährleistet. Es ist darauf zu achten, dass der 'Übersetzer' genau einen Versuch erhält, um die Datenbank-Anfragen zu formulieren.

7.3.2 Durchführung

Zunächst wird der 'Übersetzer' geschult. Er wird mit der Anfrageschnittstelle sowie dem Retrieval-System vertraut gemacht. Zur Übung formuliert der 'Übersetzer' einige Anfragen. Dafür wird eine Bildkollektion verwendet, die nicht der für die Studie verwendeten Testkol-

lektion entspricht. Nach dem Coaching des 'Übersetzers' wird mit der eigentlichen Studie begonnen.

Als Grundlage dient nun die in Abschnitt 7.3.1 beschriebene Testkollektion. Der 'Übersetzer' formuliert für eine natürlichsprachliche Anfrage jeweils eine gewichtete und eine ungewichtete Datenbank-Anfrage (siehe Anhang B). Das Ziel dabei soll es sein, ein bestmögliches Anfrageergebnis zu erreichen.

Nach der Ausführung der Anfragen werden die jeweils ersten 100 Ergebnisobjekte werden bewertet. Dafür werden die erzielten Ergebnisse mit den erwarteten Ergebnissen der natürlichsprachlichen Anfrage abgeglichen. Dies erfolgt nicht durch den 'Übersetzer' sondern durch den Ersteller der natürlichsprachlichen Anfragen. Auf die Auswertung der Ergebnisse wird im folgenden Abschnitt näher eingegangen.

7.3.3 Auswertung

Für die Bewertung der Ergebnisobjekte bezüglich ihrer Relevanz für die Anfrage werden die *Precision-* und *Recall*-Werte ermittelt. Die Precision **P** beschreibt die Anzahl relevanter Ergebnisobjekte im Verhältnis zur Ergebnisanzahl:

$$\mathbf{P} = \frac{|R_a|}{|A|},$$

wobei A die Ergebnismenge und R_a die Menge relevante Objekte in der Ergebnismenge ist. Die Precision dient als Maß für die Qualitätseinschätzung und kann daher auch als *Genauigkeit* bezeichnet werden. Der Recall **R** ergibt sich aus dem Verhältnis der Anzahl relevanter Ergebnisobjekte zur Anzahl aller für die Anfrage relevanter Objekte:

$$\mathbf{R} = \frac{|R_a|}{|R|},$$

wobei R die Menge relevanter Objekte bezüglich einer Anfrage darstellt. Demnach ist der Recall ein Maß für die Quantitätseinschätzung und wird auch als *Vollständigkeit* bezeichnet.

Für die Testanfragen *A1* bis *A4* von Seite 175 sind diese beiden Werte in dem Diagramm in Abbildung 7.12 abgetragen. Die Precision ist als Funktion der Recall-Werte dargestellt, wobei für gleiche Recall-Werte jeweils der größte Precision-Wert verwendet wird. Typischerweise fallen die Graphen von rechts nach links. Dies bedeutet, dass je mehr relevante Objekte ermittelt werden auch gleichzeitig umso mehr irrelevante Objekte ermittelt werden.

Aus dem Diagramm geht deutlich hervor, dass durch eine Abbildung der Nutzerpräferenzen bereits bei der Anfrageformulierung bessere Retrieval-Ergebnisse erzielt werden können. Dies drückt sich im Vergleich der gewichteten Anfragen zu den entsprechenden ungewichteten Anfragen in höheren Precision- und Recall-Werten aus.

Werden die durchschnittlichen Precision-Werte über alle relevanten Objekte sowie die maximalen Recall-Werte betrachtet, so wird diese Aussage bekräftigt. Abbildung 7.13 zeigt diese Werte in jeweils einem Diagramm. Bei den Anfragen *A1* und *A4* werden sowohl im gewichteten als auch im ungewichteten Fall alle relevanten Objekte ermittelt. Jedoch wird bei Betrachtung der durchschnittlichen Precision-Werte deutlich, dass die gewichteten Anfragen den ungewichteten Anfragen überlegen sind. Eine hohe durchschnittliche Precision drückt aus, dass relevante Objekte frühzeitig gefunden werden.

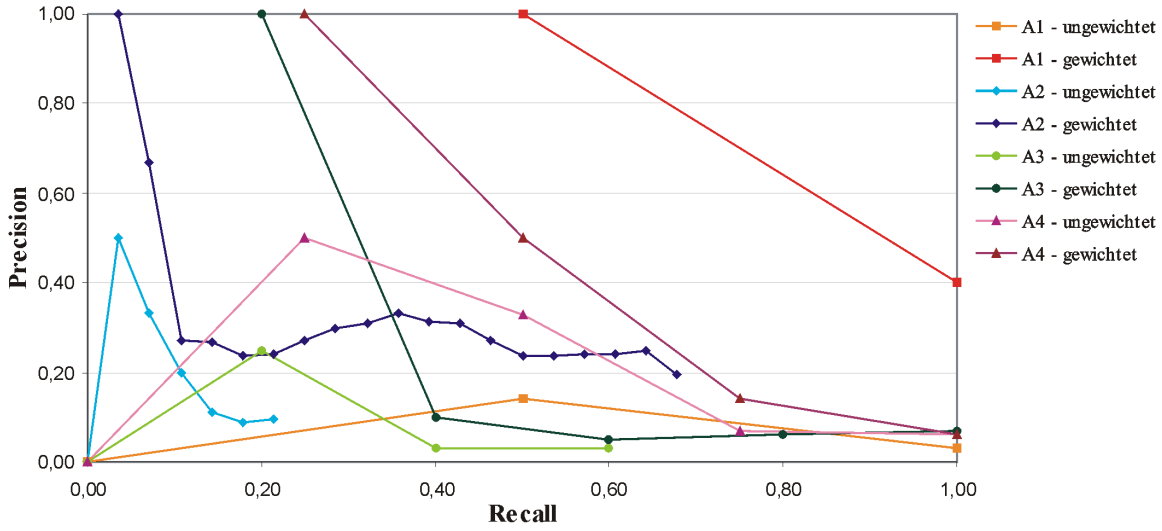


Abbildung 7.12: Precision-Recall der Anfragen A1-A4 gewichtet und ungewichtet

Durch dieses Ergebnis wird die eingangs aufgestellte Hypothese bestätigt. Natürlich sind weitere Studien wünschenswert, um eine allgemein gültige Aussage über die Vorteile des Multi-Level-Gewichtungsmodells treffen zu können. Jedoch kann hier bereits festgestellt werden, dass das Multi-Level-Gewichtungsmodell durchaus eine gute Abbildung der Nutzerpräferenzen ermöglicht. Anzumerken sei, dass die erzielten Ergebnisse natürlich von der Wahl der Gewichte und damit vom subjektiven Nutzerempfinden abhängen. Daher können gewichtete Anfragen im Vergleich zu ungewichteten Anfragen auch schlechtere Retrieval-Ergebnisse erzielen, sofern die Gewichte unglücklich gewählt werden. Eine Unterstützung des Nutzers bei der Abbildung seiner Präferenzen wäre hier wünschenswert und stellt eine Aufgabe für zukünftige Forschungstätigkeiten dar.

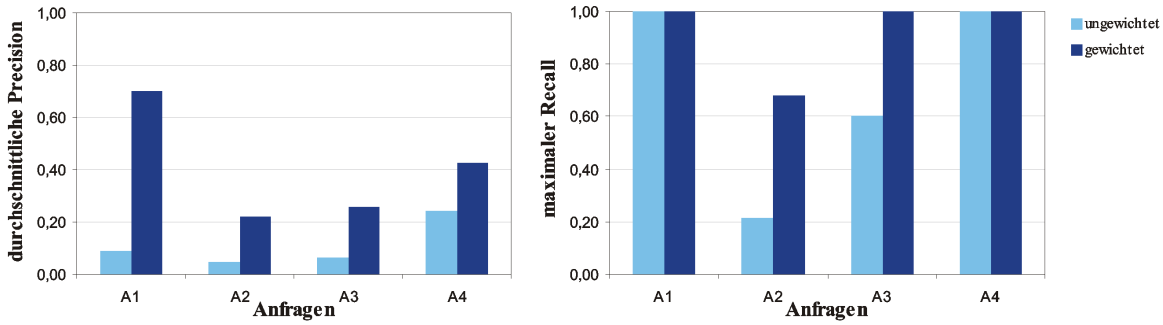


Abbildung 7.13: durchschnittliche Precision und maximaler Recall für die Anfragen A1-A4

Bei der Untersuchung der vom 'Übersetzer' formulierten gewichteten Anfragen kann festgestellt werden, dass sowohl Relevanz-, Schwellwert- und Operatorgewichte Anwendung

finden (vgl. Anhang B). Während Relevanz und Operatorgewichte in jeder Anfrage eingesetzt werden, werden die Schwellwertgewichte nur selten genutzt. Dies liegt zum Teil darin begründet, dass die Ergebnismenge nicht zu stark reglementiert werden durfte, da jeweils die ersten 100 Objekte bewertet werden sollten. Zusammenfassend zeigt dieses, dass die Verwendung verschiedener Gewichtungen nützlich ist. Insbesondere gilt das für die Operatorgewichte, wodurch die scharfe Semantik der Junktoren abgeschwächt werden kann, so dass sie der menschlichen Intuition näher kommen.

Zusammenfassend kann festgestellt werden, dass Nutzerpräferenzen, die auf verschiedenste Art und Weise entstehen, gut durch das Multi-Level-Gewichtungsmodell umgesetzt werden können. Ferner ist es möglich, durch gewichtete Anfragen im Vergleich zu herkömmlichen ungewichteten Anfragen bessere Retrieval-Ergebnisse zu erzielen.

7.4 Zusammenfassung

In diesem Kapitel wurden die verschiedenen, prototypisch umgesetzten Systemkomponenten vorgestellt. Es wurde dargelegt, dass die entwickelten Transformationsregeln für gewichtete Anfragen sowie die Korrektheit der Reduktion eines $\mathcal{SDC}$-Ausdrucks in einen Ausdruck der Ähnlichkeitsalgebra auch experimentell bestätigt werden können. Darüber hinaus wurde die Ähnlichkeitssuche in dem System LostArt vorgestellt. Es wurde gezeigt, wie in diesem System inhaltsbasierte und räumliche Anfragen formuliert werden. Diese beiden Suchoptionen erweitern die bisherigen Möglichkeiten der textbasierten Suche in LostArt. Sie unterstützen damit eine verbesserte Recherche nach Doubletten. Denn nun kann eine Suche nach ähnlichen Objekten unabhängig von möglicherweise fehlerhaften textuellen Beschreibungen erfolgen. Als letzte Komponente wurde ein Prototyp zur Formulierung von WS-QBE-Anfragen vorgestellt. Dieser setzt auf Oracle9i auf und greift für die Ähnlichkeitsberechnung auf das *interMedia-Paket* zurück.

Im zweiten Teil des Kapitels erfolgte die Evaluierung der Multimedia-Anfragesprache WS-QBE. Anhand von Beispielanfragen aus dem Kontext von LostArt wurde zunächst demonstriert, wie Anfragen in WS-QBE formuliert werden können. Dem schloss sich die Untersuchung der Sprache bezüglich der Erfüllung der Anforderungen an eine Multimedia-Anfragesprache an. Es wurde analysiert, inwiefern WS-QBE den allgemeinen Anforderungen an eine Anfragesprache gerecht wird. Hier konnte festgestellt werden, dass die geforderten Kriterien prinzipiell erfüllt werden. WS-QBE ist nicht streng relational vollständig. Dies kann jedoch durch eine orthogonale Erweiterung der Sprache im Sinne von SQBE erreicht werden. Die weiteren Untersuchungen von WS-QBE erfolgten in Hinblick auf die Erfüllung der multimediaspezifischen Anforderungen. Auch diesen wird WS-QBE gerecht. An dieser Stelle bestehen jedoch Potenziale für einen Ausbau der Anfragesprache. So kann die Sprache beispielsweise um Funktionen erweitert werden. Letztlich wurden die Möglichkeiten zur Parametrisierung der Multimedia-Anfragesprache evaluiert. Durch Einführung der Print-Option und durch die Möglichkeiten, räumliche und zeitliche Ausgabebedingungen zu spezifizieren, kann WS-QBE dieser Anforderung gerecht werden.

Ferner wurde in diesem Kapitel eine Fallstudie zum Multi-Level-Gewichtungsmodell vorgestellt. Ziel war es, zu untersuchen, ob Präferenzen durch das Gewichtungsmodell adäquat umgesetzt werden können, so dass im Vergleich zu ungewichteten Anfragen bessere Retrieval-Ergebnisse erzielt werden. Die Studie lehnte sich dabei an die Vorgehensweise bei

TREC an. Es wurde auf die Vorbereitung, die Durchführung und die Auswertung der Studie eingegangen. Die Auswertung zeigte, dass für gewichtete Anfragen bessere Precision- und Recall-Werte erzielt werden als für ungewichtete Anfragen. Damit wird die Hypothese, dass durch die Abbildung von Nutzerpräferenzen bereits bei der Anfrageformulierung besser Retrieval-Ergebnisse erzielt werden, durch das Ergebnis der Studie bestätigt. Hier sind weitere Studien wünschenswert, die es erlauben eine allgemein gültige Aussage über die Vorteile des Multi-Level-Gewichtungsmodells zu treffen.

8 Zusammenfassung und Ausblick

In diesem Kapitel wird diese Arbeit zusammengefasst. Es wird ferner ein Ausblick auf weitere Forschungsansätze gegeben.

8.1 Zusammenfassung

Multimedia-Retrieval-Systeme unterstützen neben den klassischen scharfen Prädikaten auch unscharfe Prädikate. So werden beispielsweise Ähnlichkeitsvergleiche ermöglicht. Dadurch kann bereits ein Defizit klassischer Datenbanksysteme abgebaut werden. Dennoch fehlt es bei den derzeitigen Multimedia-Retrieval-Systeme an Mechanismen bei der Anfrageformulierung, die die Spezifikation individueller Nutzerpräferenzen erlauben. Bestehende Multimedia-Anfragesprachen können die Formulierung von Präferenzen nur bedingt unterstützen. Ferner weisen sie oftmals bezüglich der universellen Einsetzbarkeit Mängel auf. Um diese Defizite der MIRS abzubauen, wurde in dieser Arbeit die Integration von Nutzerpräferenzen im Kontext des Multimedia-Retrievals untersucht. Dies mündete in einem Vorschlag für eine viuselle Multimedia-Anfragesprache.

In Abbildung 8.1 sind die aufeinander aufbauenden Ergebnisse dieser Arbeit dargestellt. Beginnend beim Multi-Level-Gewichtungsmodell, über eine entsprechende Erweiterung des Bereichskalküls und der relationalen Algebra bis hin zu der nutzerfreundlichen Multimedia-Anfragesprache WS-QBE wurde somit eine in sich schlüssige Lösung für die Formulierung von Präferenzen in Multimedia-Anfragen entwickelt.

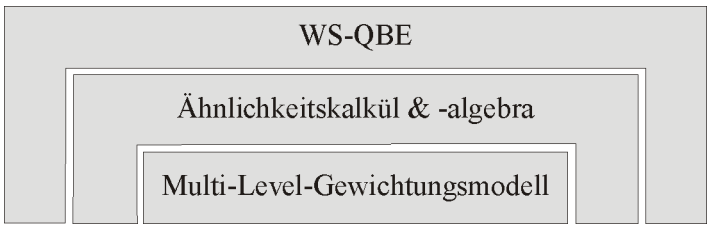


Abbildung 8.1: Zusammenfassung der Arbeit

In Kapitel 2 und 3 wurde in das Gebiet des Multimedia-Information-Retrievals eingeführt. Es wurden sowohl die grundlegenden Prinzipien sowie die Charakteristika des Multimedia-Retrievals als auch die vielfältigen Möglichkeiten zur Einbindung von Nutzerpräferenzen in den Retrieval-Prozess beschrieben. Ferner wurden Anforderungen an eine Multimedia-Anfragesprache zusammengetragen. Diesbezüglich erfolgte dann eine Untersuchung bestehender Multimedia-Anfragesprachen. Darüber hinaus wurden verschiedene, theoretisch fundierte Ansätze zur Abbildung von Nutzerpräferenzen bei der Anfrageformulierung vorgestellt. Hierbei wurden die Defizite jedoch auch die Vorteile bestehender Sprachen und

Ansätze deutlich.

Das sodann in Kapitel 4 entwickelte *Multi-Level-Gewichtungsmodell* bildet die theoretische Grundlage für die Entwicklung einer neuen Multimedia-Anfragesprache. Es integriert eine Relevanzgewichtung, eine Schwellwertgewichtung sowie eine Operatorgewichtung. Es können daher nicht nur Präferenzen bezüglich der Wichtigkeit einzelner Anfrageterme formuliert werden. Vielmehr ist es darüber hinaus möglich, sowohl die Ergebnismenge durch die Schwellwertgewichtung zu beeinflussen als auch das Verhalten der Junktoren zu modifizieren. Letzteres wird notwendig, da das Verhalten der klassischen Junktoren *UND* und *ODER* nicht der menschlichen Intuition entspricht (Zimmermann und Zysno, 1980).

Das Multi-Level-Gewichtungsmodell unterstützt damit eine detaillierte Spezifikation der Suchanforderungen eines Nutzers. Der Nutzer verfügt bereits bei der Anfrageformulierung über die Freiheit, seine Präferenzen unter Verwendung verschiedener Gewichtungsalternativen zu beschreiben. Dadurch ist es möglich, dass bereits für die Initialanfrage bessere Retrieval-Ergebnisse erzielt werden können. Anhand einer Fallstudie zum Gewichtungsmodell wurde dies exemplarisch belegt. Für diesen Zweck wurden die *Precision* und der *Recall* für verschiedene Anfragen – einmal mit und einmal ohne Nutzerpräferenzen – bestimmt.

Für die Evaluierung gewichteter Anfragen werden *gewichtete Scoring-Funktionen* eingesetzt. Dafür wurden Anforderungen hinsichtlich einer adäquaten Aggregation der Relevanzwerte unter Berücksichtigung der Gewichtung spezifiziert. Für komplexe Anfragen wurde eine Multi-Level-Scoring-Funktion sowie entsprechende Anforderungen an diese eingeführt. Konkrete gewichtete Scoring-Funktionen, die diesen Anforderungen gerecht werden, wurden ebenfalls vorgestellt. Darüber hinaus wurden *Transformationsregeln* für relevanzgewichtete Anfragen entwickelt. Auf Grund des Einflusses der Gewichte sind die klassischen Transformationsregeln für gewichtete Anfragen nicht uneingeschränkt gültig. Um neben Kommutativität, De Morgan, Idempotenz und Involution auch Assoziativität und Distributivität zu gewährleisten, war die Einführung von Regeln zur Modifikation der Gewichte notwendig. Die entwickelten Transformationsregeln können im Rahmen einer späteren Anfrageoptimierung genutzt werden.

Aufbauend auf dem Multi-Level-Gewichtungsmodell wurde in Kapitel 5 ein *Ähnlichkeitskalkül* vorgestellt. Dies basiert auf dem Vorschlag von Schmitt und Schulz (2004a,b). Es wurde derart erweitert, dass der Forderung nach Abgeschlossenheit nachgekommen wird, so dass das Ergebnis einer Anfrage eine Fuzzy-Relation ist, die wiederum als Eingabe für eine Anfrage verwendet werden kann. Ferner wurden n-äre Junktoren, ein Universaljunktor sowie ein weiterer Schwellwertoperator eingeführt. Das Ähnlichkeitskalkül erlaubt die Formulierung von Anfragen auf eine deklarative Art und Weise. Es erweitert das klassische Bereichskalkül um unscharfe Prädikate und eine Gewichtung. Die konkrete Semantik der Operationen und Junktoren wurde dabei offen gelassen. Es wurden lediglich Anforderungen angegeben, die von diesen Operatoren erfüllt werden müssen. Beispielsweise wird für die Umsetzung der Junktoren gefordert, dass dafür Scoring-Funktionen verwendet werden. Das Ähnlichkeitskalkül kann daher als eine Art Framework verstanden werden, welches für konkrete Anwendungen angepasst werden kann.

Die Ausdrücke des Ähnlichkeitskalküls können auf einen entsprechenden Ausdruck der *Ähnlichkeitsalgebra* überführt werden. Die Ähnlichkeitsalgebra ist eine Erweiterung der Relationenalgebra. Sie ist somit mächtiger als die relationale Algebra. Es können nicht nur scharfe Anfragen sondern auch unscharfe sowie gewichtete Anfragen formuliert werden. Damit wird ein Ziel der Arbeit, die Spezifikation von Nutzerpräferenzen bei der Formulierung von

Multimedia-Anfragen, erreicht.

Neben der Beschreibung der Ähnlichkeitsalgebra wurde für die Überführung eines Ähnlichkeitskalkülausdrucks in einen Algebraausdruck ein Reduktionsalgorithmus angegeben. Dieser basiert auf der Vorgehensweise bei der Transformation eines Ausdrucks des Bereichskalküls in einen Ausdruck der Relationenalgebra. Die Reduktion eines Ausdrucks des Ähnlichkeitskalküls wurde an einem Beispiel verdeutlicht.

Die Formulierung von Anfragen in Form eines Kalkülausdrucks ist jedoch für den ungeübten Nutzer viel zu schwierig und daher nicht nutzerfreundlich. Dies machte die Entwicklung einer deklarativen Multimedia-Anfragesprache, die eine gute Ad-hoc-Formulierung von Multimedia-Anfragen unterstützt, notwendig. Im Multimedia-Information-Retrieval ist die Formulierung von Query-By-Example Anfragen sehr beliebt und weit verbreitet. Darüber hinaus ist diese Art der Formulierung von Anfragen im Vergleich zu textbasierten Anfragen intuitiver. Aus diesem Grund wurde in Kapitel 6 die visuelle QBE-ähnliche *Multimedia-Anfragesprache WS-QBE* entworfen. Sie erlaubt die Umsetzung des Gewichtungsmodells durch die Bereitstellung verschiedener Sprachkonstrukte zur Beschreibung der Präferenzen. So können einfache Multimedia-Anfragen, ähnlich wie in QBE, durch Ausfüllen von Tabellengerüsten formuliert werden. Zusätzlich ist eine Gewichtung von Anfragetermen durch Vergabe von Gewichtsvariablen oder durch Angabe eines Schwellwertes möglich. Wichtig bei Multimedia-Anfragen ist die Möglichkeit der Formulierung komplexer Anfragen. Dafür wurde der Condition-Tree eingeführt. Dieser ermöglicht es, Suchbedingungen hierarchisch als Baumstruktur zu organisieren. Die Vergabe von Relevanzgewichten, Schwellwertgewichten sowie eine Gewichtung der Operatoren wird für komplexe Anfragen unterstützt. Die Anfragesprache WS-QBE bietet somit dem Nutzer eine gute Möglichkeit, seine Präferenzen bereits bei der ersten Anfrageformulierung auszudrücken.

Da WS-QBE als Multimedia-Anfragesprache entworfen wurde, musste der Aspekt der Formulierung von Ausgabebedingungen berücksichtigt werden. Hierfür wurde eine Möglichkeit geschaffen, spezielle Ausgabebedingungen optional zu spezifizieren. Auch für diesen Zweck werden Variablen, so genannte Print-Variablen, eingesetzt. Die Beschreibung der räumlichen Ausgabebedingungen wird grafisch unterstützt, indem der Nutzer unter Verwendung der Print-Variablen ein Ausgabelayout erstellt.

Die formale Semantik von WS-QBE wird durch die Überführung der Anfragen auf das Ähnlichkeitskalkül beschrieben. Hierfür wurde ein entsprechender Algorithmus angegeben. Anhand eines Beispiels wurde die Überführung einer WS-QBE in das Ähnlichkeitskalkül demonstriert.

Abschließend wurden in Kapitel 7 die verschiedenen prototypisch implementieren Systemkomponenten vorgestellt. Sowohl die Transformationsregeln für relevanzgewichtete Anfragen als auch das Ähnlichkeitskalkül, die -algebra und die entsprechende Überführung ineinander wurden in jeweils einer Komponente implementiert. Darüber hinaus wurde eine gewichtete Ähnlichkeitssuche sowie die Formulierung räumlicher Anfragen im Anwendungsszenario LostArt realisiert. Als letzte Komponente wurde die Umsetzung der Anfragesprache WS-QBE vorgestellt.

Das Ziel, welches mit der prototypischen Implementierung der einzelnen Komponenten verfolgt wurde, bestand in der experimentellen Validierung der verschiedenen theoretischen Modelle. So konnten die Transformationsregeln und die Korrektheit des Reduktionsalgorithmus bestätigt werden. Letztlich hat die Evaluierung der Anfragesprache WS-QBE gezeigt, dass die in Kapitel 3 spezifizierten Anforderungen an eine Multimedia-Anfragesprache zum

Großteil gut erfüllt werden. Es wurde ferner festgestellt, dass weitere Potenziale für einen Ausbau der Sprache bestehen. Hierauf und auf weitere sich aus dieser Arbeit ergebenden Forschungsthemen wird im folgenden Abschnitt näher eingegangen.

8.2 Ausblick

Die in dieser Arbeit entwickelten Ansätze für die Formulierung und Abbildung von Präferenzen bilden die Grundlagen für weitere Forschungsarbeiten. So basiert das Multi-Level-Gewichtungsmodell derzeit auf Präferenzen, die durch unterschiedlichste Arten von Gewichten ausgedrückt werden können. Es stellt sich daher die Frage, wie Präferenz in Form von Halbordnungen (vgl. Abschnitt 2.4.1) in das Gewichtungsmodell integriert werden können. Damit wäre es möglich, Präferenzen auf einer abstrakteren Ebene zu formulieren.

Auf Grundlage der durchgeführten Fallstudie zum Multi-Level-Gewichtungsmodell können und sollten weitere Studien durchgeführt werden. Wünschenswert ist es hierbei zu untersuchen, inwiefern die Qualität der erzielten Ergebnisse von unterschiedlichen Erfahrungen einzelner Nutzer abhängt. Sollten sich daraus unterschiedlich qualifizierte Nutzergruppen ergeben, so sind entsprechende Richtlinien abzuleiten, welche die Nutzer bei der Formulierung ihrer Präferenzen in Form von Gewichten unterstützen.

Ferner sollte die Möglichkeit untersucht werden, Nutzerprofile (vgl. Abschnitt 2.4.5) in das Gewichtungsmodell zu integrieren. Bisher wird als Initialgewichtung für die Relevanzgewichte stets eine Gleichgewichtung der Bedingungen betrachtet. Dies entspricht dem ungewichteten Fall. Auch für den Universaljunktor wurde ein beliebiger z-Wert festgelegt. Er wurde hier als *ODER*-Operator initialisiert. An Stelle dieser vordefinierten Gewichte können jedoch die Informationen aus Nutzerprofilen einfließen und einen nutzertypischen Gewichtungsvorschlag unterstützen.

Einen weiteren Forschungsschwerpunkt könnte die Erweiterung der entwickelten Multimedia-Anfragesprache WS-QBE darstellen. Die Sprache wurde zunächst bewusst einfach gehalten und nicht mit Funktionalität überladen. Daher wurden nicht alle im Ähnlichkeitskalkül bereitgestellten Funktionen in WS-QBE übernommen (vgl. Kapitel 5). Dazu zählen beispielsweise die parametrisierbaren Quantoren und weitere Schwellwertoperatoren. Diese Funktionen können bei einem Ausbau der Sprache Berücksichtigung finden.

Ferner ist die Integration von Funktionen und Gruppierungsoperationen in das Ähnlichkeitskalkül, die Ähnlichkeitsalgebra sowie in die Multimedia-Anfragesprache WS-QBE wünschenswert. Dies ist eine besonders herausfordernde Aufgabe, da dadurch die bisher angebotene Kernfunktionalität der Sprachen erweitert wird. Demnach können mehr anwendungsspezifische und komplexere Multimedia-Anfragen realisiert werden. In diesem Kontext erscheint es auch sinnvoll, die Konzepte von SQBE aufzugreifen und WS-QBE derart zu erweitern, dass eine strenge relationale Vollständigkeit erreicht wird (vgl. Abschnitt 6.2.2). Damit verbunden ist die Erreichung einer größtmöglichen Orthogonalität der Sprache WS-QBE.

Einen weiteren wichtigen Schwerpunkt im Zusammenhang mit der Anfrageauswertung bildet die algebraische Optimierung. Aufgrund der Gewichte sind die klassischen Optimierungsregeln der Relationenalgebra nicht uneingeschränkt gültig. Darüber hinaus stellt auch eine iterative Anfrageverfeinerung unter Berücksichtigung des Multi-Level-Gewichtungsmodells eine herausfordernde Aufgabe dar. In den folgenden Abschnitten wird auf diese

zwei vielversprechenden Forschungsthemen, die sich direkt aus dieser Arbeit ergeben, näher eingegangen.

8.2.1 Algebraische Optimierung

An dem Beispiel des Stoffherstellers aus Abschnitt 5.5.2 (siehe auch Abbildung 8.2) wurde bereits deutlich, dass die aus dem Kalkül generierten Ausdrücke der Ähnlichkeitsalgebra sehr komplex sind. Dies macht eine algebraische Optimierung notwendig. Für diese Aufgabe müssen spezielle Optimierungsregeln, die auf die Überführungsregeln eines Ähnlichkeitskalkülausdrucks in einen entsprechenden Algebraausdruck abgestimmt sind, entworfen werden. Neben einer Optimierung, die im Anschluss an die Überführung stattfindet, erscheint es auch sinnvoll, bereits während der Überführung entsprechende Optimierungsregeln zu integrieren. Letzteres stellt eine besonders interessante Herausforderung dar.

Im Allgemeinen werden für die algebraische Optimierung verschiedene Äquivalenzregeln der Relationenalgebra und Heuristiken ausgenutzt. Diese werden beispielsweise in (Saake und Heuer, 1999, S. 356ff) sowie in (Elmasri und Navathe, 2002, S. 648ff) näher beschrieben. Für die Ähnlichkeitsalgebra ist zunächst zu prüfen, ob diese Regeln übertragbar sind beziehungsweise wie eine Anpassung unter Berücksichtigung der Gewichte erfolgen kann. Am Beispiel des Stoffherstellers wird die Anwendung einiger aus der Relationenalgebra bekannter Optimierungsregeln beschrieben. Zusätzlich dazu werden auf das Ähnlichkeitskalkül abgestimmte Regeln skizziert. Der generierte Anfragebaum, der die Ausgangsbasis für die Optimierung bildet, ist dafür noch einmal in Abbildung 8.2 dargestellt. Die zu Grunde liegenden Relationen sind *Stoff(SId, Name, Qualitaet)* und *Muster(BId, SId, Bild)*.

Eine wesentliche Aufgabe der Optimierung ist das Erkennen von Verbunden statt Kreuzprodukten sowie das Verschieben von Selektionen und Projektionen möglichst weit in Richtung der Blätter, um die Zwischenergebnisse klein zu halten (Saake und Heuer, 1999, S. 354, 362ff). Neben diesen Heuristiken werden spezielle Transformationsregeln ausgenutzt. Im Folgenden seien r_1 und r_2 Relationen über den Schemata R_1 und R_2. Für den Beispielausdruck sind die folgenden Ersetzungsregeln von Bedeutung:

- Bei Kaskaden von π-Operationen können alle außer der letzten Projektion ignoriert werden:
$$\pi_{\#_1}(\pi_{\#_2}(r_1)) \longleftrightarrow \pi_{\#_1}(r_1).$$

- Komplexe Selektionsbedingungen können durch Mengenoperationen umgesetzt werden, wie beispielsweise:
$$\sigma_{F_1 \vee F_2}(r_1) \longleftrightarrow \sigma_{F_1}(r_1) \cup \sigma_{F_2}(r_1).$$
Diese Regel gilt auch in der Ähnlichkeitsalgebra. Aufgrund der Gewichtung ergeben sich hierbei weitere Regeln, wie etwa:
$$\begin{aligned}
\sigma_{F_1 \wedge_{(\theta_1,\theta_2)} F_2}(r_1) &\longleftrightarrow \sigma_{F_1}(r_1) \cap_{(\theta_1,\theta_2)} \sigma_{F_2}(r_1) \\
\sigma_{F_1 \vee_{(\theta_1,\theta_2)} F_2}(r_1) &\longleftrightarrow \sigma_{F_1}(r_1) \cup_{(\theta_1,\theta_2)} \sigma_{F_2}(r_1) \\
\sigma_{F_1 \oplus^z F_2}(r_1) &\longleftrightarrow \sigma_{F_1}(r_1) \uplus^z \sigma_{F_2}(r_1) \\
\sigma_{F_1 \oplus^z_{(\theta_1,\theta_2)} F_2}(r_1) &\longleftrightarrow \sigma_{F_1}(r_1) \uplus^z_{(\theta_1,\theta_2)} \sigma_{F_2}(r_1).
\end{aligned}$$

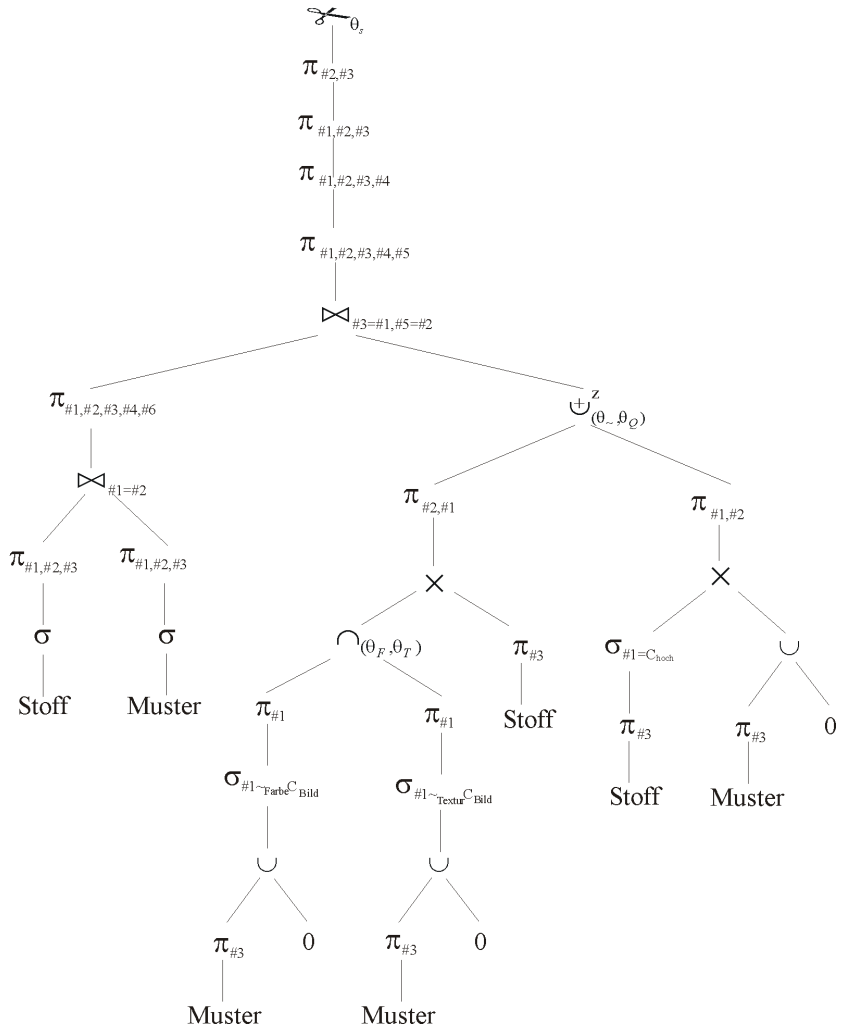


Abbildung 8.2: Anfragebaum des generierten Algebraausdrucks aus dem Beispiel des Stoffherstellers

Diese Regeln sind auf die n-ären Operationen übertragbar.

- Projektionen sind mit $\cup$ kommutativ:

$$\pi_{\#_1}(r_1 \cup r_2) \longleftrightarrow (\pi_{\#_1}(r_1)) \cup (\pi_{\#_1}(r_2)).$$

Dies gilt auch für den gewichteten Fall sowie für den Universaloperator, da dieser auch eine Disjunktion symbolisiert:

$$\begin{aligned}
\pi_{\#_1}(r_1 \cup_{(\theta_1,\theta_2)} r_2) &\longleftrightarrow (\pi_{\#_1}(r_1)) \cup_{(\theta_1,\theta_2)} (\pi_{\#_1}(r_2)) \\
\pi_{\#_1}(r_1 \uplus^z r_2) &\longleftrightarrow (\pi_{\#_1}(r_1)) \uplus^z (\pi_{\#_1}(r_2)) \\
\pi_{\#_1}(r_1 \uplus^z_{(\theta_1,\theta_2)} r_2) &\longleftrightarrow (\pi_{\#_1}(r_1)) \uplus^z_{(\theta_1,\theta_2)} (\pi_{\#_1}(r_2)).
\end{aligned}$$

Diese Regeln sind auf die *n*-ären Operationen übertragbar.

Unter Verwendung dieser Transformationsregeln sowie der allgemeinen Heuristiken für die algebraische Optimierung stellt der in Abbildung 8.3 gezeigte Anfragebaum eine optimierte Variante des ursprünglichen Algebrausdrucks (Abbildung 8.2) dar.

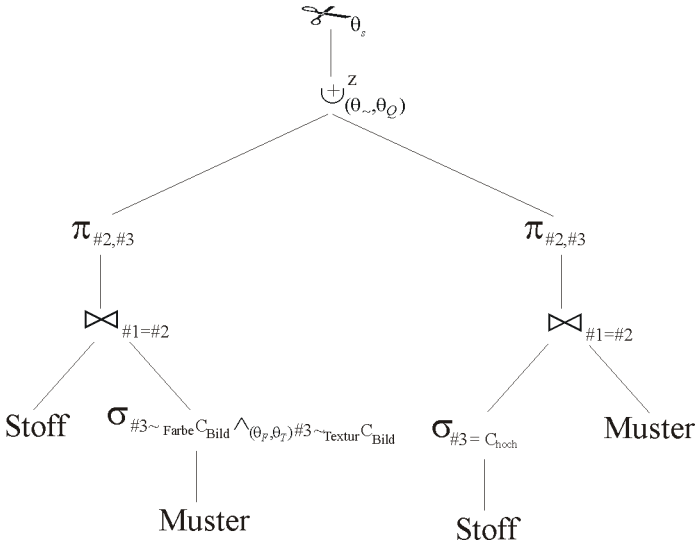


Abbildung 8.3: Optimierte Variante des Algebraausdrucks aus dem Beispiel des Stoffverkäufers

Aufbauend auf diesen kurz skizzierten Optimierungsregeln, sind weitere Regeln zu entwickeln. Ein erster Schritt ist dabei die Überprüfung der Übertragbarkeit der anderen Transformationsregeln der Relationenalgebra auf die Ähnlichkeitsalgebra. Ferner müssen unter Umständen zusätzliche Regeln abgeleitet werden. Von besonderem Interesse ist dabei die Entwicklung von Optimierungsregeln für die gewichteten Operatoren. Die Transformationsregeln aus Abschnitt 4.3.3 können bei dem Entwurf von Optimierungsregeln Berücksichtigung finden, um beispielsweise die Assoziativität von gewichteten Operatoren zu gewährleisten. Eine weitere Möglichkeit zur Optimierung eines Ausdrucks der Ähnlichkeitsalgebra kann durch einen gewichteten Join-Operator erzielt werden.

Nachfolgend kann dann für die internen ausführbaren Pläne einer Anfrage eine kostenbasierte Optimierung vorgenommen werden. Dafür ist die Erstellung entsprechender Statistiken für die Auswertung der gewichteten Operationen notwendig.

8.2.2 Relevanz-Feedback

Die Spezifizierung von zusätzlichen Nutzerpräferenzen durch eine Gewichtung bei der Formulierung von Anfragen kann nicht die semantische Lücke zwischen dem Nutzer und dem System schließen. Aus diesem Grund verlieren die bestehenden Methoden für eine iterative Anfragemodifizierung nicht an Bedeutung. Vielmehr können auf Grund der theoretischen Basis, die mit dem Multi-Level-Gewichtungsmodell gegeben ist, neue Relevanz-Feedback-Methoden entwickelt werden. Das Gewichtungsmodell legt die Arten von Gewichten fest,

die bei der Anfrageformulierung Berücksichtigung finden. Wesentlich dabei ist, dass beschrieben wird, wie die verschiedenen Gewichte in die Anfrageauswertung einfließen.

Im Unterschied zu den bestehenden Ansätzen des Relevanz-Feedbacks, wie in Abschnitt 2.4.3 beschrieben, können nun nicht mehr nur Relevanzgewichte sondern auch zusätzlich Schwellwert- und Operatorgewichte modifiziert werden. Ein sehr einfaches Verfahren für eine Anfrageverfeinerung soll kurz vorgestellt werden: Wird eine Anfrage unter Verwendung von Schwellwertgewichten zu restriktiv formuliert, so dass keine Ergebnisobjekte ermittelt werden, dann können automatisch die Schwellwertgewichte verringert werden. Damit wird die neue Anfrage weniger restriktiv, so dass eine größere Ergebnismenge ermittelt werden kann. Analog dazu können die Schwellwertgewichte nach oben gesetzt werden, falls die Anfrage zu viele Ergebnisobjekte ermittelt.

Neben diesem naiven Ansatz verlangt die Justierung der unterschiedlichen Gewichte jedoch nach weiteren adäquaten Relevanz-Feedback-Methoden. Diese müssen ein Mapping der Feedback-Informationen des Nutzers auf die verschiedenen Gewichte erlauben. Insbesondere stellen sich hier die Fragen:

- Welche Feedback-Informationen sind für die Modifizierung der Schwellwert- und Operatorgewichte von Bedeutung?
- Ist eine relevant/nicht relevant-Entscheidung ausreichend, um beispielsweise Operatorgewichte zu modifizieren?
- Sind weitere Informationen als Feedback vom Nutzer notwendig?

Allein diese Fragen machen bereits deutlich, dass die Entwicklung neuer Feedback-Methoden sehr vielschichtig ist. So muss gewährleistet werden, dass das Nutzerempfinden durch eine Modifizierung der Gewichtung adäquat abgebildet wird. Damit besteht zur Erreichung dieses Zieles ein erhebliches Forschungspotenzial.

8.2.3 Fazit

Mit den hier skizzierten Erweiterungen der Sprache und einer optimierten Anfrageevaluierung wäre es denkbar, WS-QBE als Anfrageschnittstelle für ein Multimedia-Retrieval-System anzubieten. Vorstellbar wäre der Einsatz beispielsweise für Oracle und DB2. Diese DBMS stellen bereits entsprechende Retrieval-Funktionalitäten durch das *InterMedia*-Paket beziehungsweise den *Datablades* bereit. Es fehlt hier jedoch an einer nutzerfreundlichen Schnittstelle für die Anfrageformulierung.

Eine andere Möglichkeit bietet der Ausbau der Standardsoftware MS-Access. Dieses Datenbank-Management-System ist bei einem breiten Spektrum an Nutzern mit unterschiedlichem Wissensstand und Kenntnissen über Datenbanken etabliert. Ein Großteil der Nutzer verwendet die QBE-Schnittstelle für die Anfrageformulierung in MS-Access. Wird diese Schnittstelle um die in WS-QBE beschriebenen Konzepte erweitert und werden ferner entsprechende Multimedia-Retrieval-Funktionalitäten bereitgestellt, kann ein nutzerfreundliches Multimedia-Retrieval angeboten werden. Dies kommt insbesondere vielen Laien entgegen, die auf Grund der Verbreitung digitaler Technologien, wie beispielsweise Digitalkameras und Foto-Handys, über größere Bildarchive verfügen.

Literaturverzeichnis

[ACM 2000] RIECKEN, D. (Hrsg.): *Special Issue of the Communication of the ACM on Personalization*. 2000

[Adali u. a. 1998] ADALI, S. ; BONATTI, P. ; SAPINO, M. L. ; SUBRAHMANIAN, V. S.: A Multi-Similarity Algebra. In: HAAS, L. M. (Hrsg.) ; TIWARY, A. (Hrsg.): *SIGMOD 1998, Proceedings ACM SIGMOD International Conference on Management of Data*, ACM, 1998, S. 402–413

[Allen 1983] ALLEN, J. F.: Maintaining Knowledge about Temporal Intervals. In: *Communications of the ACM (CACM)* 26 (1983), Nr. 11, S. 832–843

[Amato u. a. 1998] AMATO, G. ; MAINETTO, G. ; SAVINO, P.: An Approach to a Content-Based Retrieval of Multimedia Data. In: *Multimedia Tools and Applications* 7 (1998), Nr. 1-2, S. 9–36

[ANSI 1999] Information technology – Database languages – SQL multimedia and application packages – Part 3: Spatial / ANSI & ISO. 1999 (ISO/IEC 13249-3:2002). – Standardbeschreibung

[ANSI 2000] Information technology – Database languages – SQL multimedia and application packages – Part 2: Full-Text / ANSI & ISO. 2000 (ISO/IEC 13249-2:2002). – Standardbeschreibung

[ANSI 2001] Information technology – Database languages – SQL multimedia and application packages – Part 5: Still Image / ANSI & ISO. 2001 (ISO/IEC 13249-5:2002). – Standardbeschreibung

[ANSI 2002] Information technology – Database languages – SQL multimedia and application packages – Part 1: Framework / ANSI & ISO. 2002 (ISO/IEC 13249-1:2002). – Standardbeschreibung

[Aslandogen u. a. 1995] ASLANDOGEN, Y. A. ; THIER, C. ; YU, C. T. ; LIN, C. ; NAIR, K. R.: Design, Implementation and Evaluation of SCORE (a System for COntent based REtrival of Pictures). In: *Proceedings of the 11th International Conference on Data Engineering (ICDE)*, IEEE Computer Society, 1995, S. 280–287

[Atnafu u. a. 2001] ATNAFU, S. ; BRUNIE, L. ; KOSCH, H.: Similarity-Based Operators and Query Optimization for Multimedia Database Systems. In: *International Database Engineering and Application Symposium*, ACM, 2001, S. 346–355

[Avrahami und Kareev 1993] AVRAHAMI, J. ; KAREEV, Y.: What Do You Expect to Get When You Ask for "A Cup of Coffee and a Muffin or a Croissant"? In: *International Journal of Man-Machine Studies* 38 (1993), Nr. 3, S. 429–434

[Backus u. a. 1960] BACKUS, J. W. ; BAUER, F. L. ; GREEN, J. ; KATZ, C. ; MCCARTHY, J. ; NAUR, P. ; PERLIS, A. J. ; RUTISHAUSER, H. ; SAMUELSON, K. ; WEGSTEIN, B. Vauquois J. H. ; WIJNGAARDEN, A. van ; WOODGER, M.: Report on the Algorithmic Language Algol 60. In: *Communications of the ACM* 3 (1960), Nr. 5, S. 299–314

[Backus u. a. 1963] BACKUS, J. W. ; BAUER, F. L. ; GREEN, J. ; KATZ, C. ; MCCARTHY, J. ; NAUR, P. ; PERLIS, A. J. ; RUTISHAUSER, H. ; SAMUELSON, K. ; WEGSTEIN, B. Vauquois J. H. ; WIJNGAARDEN, A. van ; WOODGER, M.: Revised Report on the Algorithmic Language Algol 60. In: *Computer Journal* 5 (1963), Nr. 4, S. 349–367

[Baeza-Yates und Ribeiro-Neto 1999] BAEZA-YATES, R. ; RIBEIRO-NETO, B.: *Modern Information Retrieval*. Addison Wesley, 1999

[Balabanovic und Shoham 1997] BALABANOVIC, M. ; SHOHAM, Y.: Fab: Content-based, collaborative recommendation. In: *Communications of the ACM* 40 (1997), Nr. 3, S. 66–72

[Baral u. a. 1998a] BARAL, C. ; GONZALEZ, G. ; NANDIGAM, A.: SQL+D: Extended Display Capabilities for Multimedia Database Queries. In: *Proceedings ACM Multimedia 98*, ACM Press, 1998 (ACM Multimedia), S. 109–114

[Baral u. a. 1998b] BARAL, C. ; GONZALEZ, G. ; SON, T. C.: Design and Implementation of Display Specification for Multimedia Answers. In: *Proceedings of the 14th International Conference on Data Engineering*, IEEE Computer Society, 1998, S. 558–565

[Batini u. a. 1992] BATINI, C. ; CATARCI, T. ; COSTABILE, M. F. ; LEVIALDI, S.: Visual Query Systems: A Taxonomy. In: KNUTH, E. (Hrsg.) ; WEGNER, L.M. (Hrsg.): *Visual Database Systems (VDB), II. Proceedings of the IFIP TC2/WG 2.6 Second Working Conference on Visual Database Systems* Bd. A-7, North-Holland, 1992, S. 153–168

[Beckmann u. a. 1990] BECKMANN, N. ; KRIEGEL, H.-P. ; SCHNEIDER, R. ; SEEGER, B.: The R*-Tree: An Efficient and Robust Access Method for Points and Rectangles. In: GARCIA-MOLINA, H. (Hrsg.) ; JAGADISH, H.V. (Hrsg.): *Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data*, ACM Press, 1990, S. 322–331

[Bellman und Giertz 1973] BELLMAN, R. ; GIERTZ, M.: On the Analytic Formalism of the Theory of Fuzzy Sets. In: *Information Sciences* 5 (1973), S. 149–156

[Bertino u. a. 1999] BERTINO, E. ; CATANIA, B. ; FERRARI, E.: *Multimedia IR: Models and Languages*. S. 325–344. In: BAEZA-YATES, R. (Hrsg.) ; RIBEIRO-NETO, B. (Hrsg.): *Modern Information Retrieval*, Addison Wesly, 1999

[Bertino und Ferrari 1998] BERTINO, E. ; FERRARI, E.: Temporal Synchronization Models for Multimedia Data. In: *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 10 (1998), Nr. 4, S. 612–631

[Biewer 1997] BIEWER, B.: *Fuzzy Methoden - Praxisrelevante Rechenmodelle und Fuzzy-Programmiersprachen*. Springer Verlag, 1997

[Blakowski u. a. 1991] BLAKOWSKI, G. ; HÜBEL, J. ; LANGREHR, U.: Tools for Specifying and Executing Synchronized Multimedia Presentations. In: *Second International Workshop on Networking and Operating System Support for Digital Audio and Video*, 1991, S. 271–282

[Blakowski u. a. 1992] BLAKOWSKI, G. ; HÜBEL, J. ; LANGREHR, U. ; MÜHLHÄUSER, M.: Tool support for the synchronization and presentation of distributed multimedia. In: *Computer Communications* 15 (1992), Nr. 10, S. 611–618

[Blakowski und Steinmetz 1996] BLAKOWSKI, G. ; STEINMETZ, R.: A Media Synchronization Survey: Reference Model, Specification, and Case Studies. In: *IEEE Journal on Selected Areas in Communications* 14 (1996), Nr. 1, S. 5–35

[Bookstein 1978] BOOKSTEIN, A.: On the perils of merging Boolean and weighted retrieval systems. In: *Journal of the American Society for Information Science* 29 (1978), S. 156–158

[Bookstein 1980] BOOKSTEIN, A.: Fuzzy requests: An approach to weighted Boolean search. In: *Journal of the American Society for Information Science* 30 (1980), Nr. 4, S. 240–247

[Bordogna u. a. 1995] BORDOGNA, G. ; CARRARA, P. ; PASI, G.: *Studies in Fuzziness*. Bd. 5: *Fuzzy Approaches to Extend Boolean Information Retrieval*. S. 231–274. In: BOSC, P. (Hrsg.) ; KACPRZYK, J. (Hrsg.): *Fuzziness in Database Management Systems* Bd. 5, Springer Verlag, 1995

[Borzsonyi u. a. 2001] BORZSONYI, S. ; KOSSMANN, D. ; STOCKER, K.: The Skyline Operator. In: *Proceedings of the IEEE Conference on Data Engineering*, 2001, S. 421–430

[Bosshard 2000] BOSSHARD, J.: *An Open and Powerful Relevance Feedback Engine for Content-Based Retrieval*, Eidgenössische Technische Hochschule Zürich, Diplomarbeit, 2000

[Buell und Kraft 1981] BUELL, D. A. ; KRAFT, D. H.: Threshold Values and Boolean Retrieval Systems. In: *Information Processing and Management* 17 (1981), S. 127–136

[Cardenas u. a. 1993] CARDENAS, A. F. ; IEONG, I. T. ; TAIRA, R. K. ; BARKER, R. ; BREANT, C. M.: The Knowledge-Based Object-Oriented PICQUERY+ Language. In: *IEEE Transaction on Knowledge and Data Engineering* 5 (1993), Nr. 4, S. 644–657

[Carey und Kossmann 1997] CAREY, M. J. ; KOSSMANN, D.: On saying "Enough already!" in SQL. In: *Proceedings of the ACM SIGMOD Conference on Management of Data*, 1997, S. 219–230

[Carey und Kossmann 1998] CAREY, M. J. ; KOSSMANN, D.: Reducing the Braking Distance of an SQL Query Engine. In: GUPTA, A. (Hrsg.) ; SHMUELI, O. (Hrsg.) ; WIDOM, J. (Hrsg.): *Proceedings of 24th International Conference on Very Large Data Bases (VLDB'98)*, Morgan Kaufmann Publishers, 1998, S. 158–169

[Carson u. a. 1997] CARSON, C. ; BELONGIE, S. ; GREENSPAN, H. ; MALIK, J.: Region-based image querying. In: *Proc. of the IEEE Workshop CVPR '97 Workshop on Content-Based Access of Image and Video Libraries*, 1997, S. 42–49

[Catarci u. a. 1997] CATARCI, T. ; COSTABILE, M. F. ; LEVIALDI, S. ; BATINI, C.: Visual Query Systems for Databases: A Survey. In: *Journal of Visual Languages and Computing* 8 (1997), Nr. 2, S. 215–260

[Catarci und Santucci 1995] CATARCI, T. ; SANTUCCI, G.: Are Visual Query Languages Easier to Use than Traditional Ones? An Experimental Proof. In: KIRBY, M. A. R. (Hrsg.) ; DIX, A. J. (Hrsg.) ; FINLAY, J. E. (Hrsg.): *People and Computers X, Proceedings of HCI '95*, Cambridge University Press, 1995, S. 323–338

[Chambers u. a. 1983] CHAMBERS, J. M. ; CLEVELAND, W. S. ; KLEINER, B. ; TUKEY, P. A.: *Graphical Methods for Data Analysis*. Chapman & Hall, 1983

[Chang und Fu 1981] CHANG, N.-S. ; FU, K.-S.: Picture Query Languages for Pictorial Data-Base Systems. In: *IEEE Computer* 14 (1981), Nr. 11, S. 23–33

[Chang und Liu 1984] CHANG, S.-K. ; LIU, S.-H.: Picture Indexing and Abstraction Techniques for Pictorial Databases. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6 (1984), Nr. 4, S. 475–484

[Chang u. a. 1987] CHANG, S.-K. ; SHI, Q.-Y. ; YAN, C.-W.: Iconic indexing by 2D strings. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 9 (1987), Nr. 3, S. 413–428

[Chaudhuri und Gravano 1996] CHAUDHURI, S. ; GRAVANO, L.: Optimizing Queries over Multimedia Repositories. In: JAGADISH, H. V. (Hrsg.) ; MUMICK, I. S. (Hrsg.): *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, ACM Press, 1996, S. 91–102

[Chen u. a. 2000] CHEN, C. ; GAGAUDAKIS, G. ; ROSIN, P. L.: Content-Based Image Visualization. In: *Proceedings of IEEE International Conference on Information Visualization (IV'00)*, 2000, S. 13–18

[Chen u. a. 2003] CHEN, Y. ; LIN, S. ; ZHONG, H. ; XU, Y.-Q. ; GUO, B. ; SHUM, H.-Y.: Realistic Rendering and Animation of Knitwear. In: *IEEE Transactions on Visualization and Computer Graphics* 9 (2003), Nr. 1, S. 43–55

[Chomicki 2002] CHOMICKI, J.: Querying with Intrinsic Preferences. In: JEFFERY, C. S. Jensenand K. G. (Hrsg.) ; POKORNÝ, J. (Hrsg.) ; SALTENIS, S. (Hrsg.) ; BERTINO, E. (Hrsg.) ; BÖHM, K. (Hrsg.) ; JARKE, M. (Hrsg.): *Advances in Database Technology - EDBT 2002, 8th International Conference on Extending Database Technology*, Springer Verlag, 2002 (LNCS 2287), S. 34–51

[Chu u. a. 1998] CHU, W. W. ; HSU, C.-C. ; IEONG, I. T. ; TAIRA, R. K.: Content-Based Image Retrieval using Metadata and Relaxation Techniques. In: SHETH, A. P. (Hrsg.) ; KLAS, W. (Hrsg.): *Multimedia Data Management: Using Metadata to Integrate and Apply Digital Media*. McGraw-Hill, 1998, S. 149–190

[Ciaccia u. a. 2000] CIACCIA, P. ; MONTESI, D. ; PENZO, W. ; TROMBETTA, A.: Imprecision and User Preferences in Multimedia Queries: A Generic Algebraic Approach. In: *1st International Symposium on Foundations of Information and Knowledge Systems, (FoIKS 2000)*, 2000, S. 50–71

[Cinque u. a. 1998] CINQUE, L. ; LEVIALDI, S. ; MALIZIA, A. ; OLSEN, K. A.: A Browser for Image Retrieval. In: *Journal of Visual Languages & Computing* 9 (1998), Nr. 1, S. 103–117

[Codd 1972] CODD, E. F.: Relational Completeness of Data Base Sublanguages. In: *Database Systems* (1972), S. 65–98

[Cooper 1988] COOPER, W. S.: Getting beyond Boole. In: *Information Processing and Management* 24 (1988), Nr. 3, S. 243–248

[Dat 2000] Preference SQL User Manual 1.3 / Database Preference GmbH. 2000. – Manual

[Demolombe 1982] DEMOLOMBE, R.: Syntactical characterization of a subset of domain independent formulas / ONERA-CERT, Toulouse. 1982. – Technical Report

[Demolombe 1992] DEMOLOMBE, R.: Syntactical Characterization of a Subset of Domain-Independent Formulas. In: *Journal of the ACM* 39 (1992), Nr. 1, S. 71–94

[Dionisio und Cárdenas 1996] DIONISIO, J. D. N. ; CÁRDENAS, A. F.: MQuery: A Visual Query Language for Multimedia, Timeline, and Simulation Data. In: *Journal of Visual Languages and Computing* 7 (1996), Nr. 4, S. 377–401

[Dubois und Prade 1984] DUBOIS, D. ; PRADE, H.: Criteria aggregation and ranking of alternatives in the framework of fuzzy set theory. In: *Studies in Management Sciences* 20 (1984), S. 209–240

[Dubois und Prade 1986] DUBOIS, D. ; PRADE, H.: Weighted Minimum and Maximum Operations in Fuzzy Set Theory. In: *Information Science* 39 (1986), S. 205–210

[Dubois und Prade 1996] DUBOIS, D. ; PRADE, H.: Using fuzzy sets in flexible querying: Why and how? In: CHRISTIANSEN, H. (Hrsg.) ; LARSEN, H. L. (Hrsg.) ; ANDREASEN, T. (Hrsg.): *Proceedings of the Workshop on Flexible Query-Answering Systems (FQAS'96)*, Datalogiske Skrifter (Writings on Computer Science) 62, 1996, S. 89–103

[Dubois und Prade 2001] DUBOIS, D. ; PRADE, H.: Fuzzy Logic Techniques in Multimedia Database Quering: A Preliminary Investigation of the Potentials. In: *IEEE Transaction on Knowledge and Data Engineering* (2001), S. 383–392

[Egenhofer und Franzosa 1991] EGENHOFER, M. J. ; FRANZOSA, R. D.: Point-Set Topological Spatial Relations. In: *International Journal of Geographical Information Systems* 5 (1991), Nr. 2, S. 161–174

[Egenhofer 1997] EGENHOFER, M.J.: Query Processing in Spatial-Query-by-Sketch. In: *Journal of Visual Languages and Computing* 8 (1997), Nr. 4, S. 403–424

[Egenhofer und Franzosa 1995] EGENHOFER, M.J. ; FRANZOSA, R.D.: On the Equivalence of Topological Relations. In: *International Journal of Geographical Information Systems* 9 (1995), Nr. 2, S. 133–152

[Elmasri und Navathe 2002] ELMASRI, R. ; NAVATHE, S. B.: *Grundlagen von Datenbanksystemen*. 3. Pearson Studium, 2002

[Fagin 1996] FAGIN, R.: Combining fuzzy information from multiple systems. In: *Proceedings of the 15th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'96)*, 1996, S. 216–226

[Fagin 1998] FAGIN, R.: Fuzzy Queries in Multimedia Database Systems. In: *Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, (PODS'98)*, ACM Press, 1998, S. 1–10

[Fagin 1999] FAGIN, R.: Combining Fuzzy Information from Multiple Systems. In: *Journal of Computer and System Sciences* 58 (1999), Nr. 1, S. 83—99

[Fagin und Maarek 2000] FAGIN, R. ; MAAREK, Y.S.: Allowing users to weight search terms. In: *Proceedings of Computer-Assisted Information Retrieval, (RIAO'00)*, 2000, S. 682–700

[Fagin und Wimmers 1997] FAGIN, R. ; WIMMERS, E. L.: Incorporating User Preferences in Multimedia Queries. In: *Proceedings of the 6th International Conference on Database Theory*, Springer Verlag, 1997 (LNCS 1186), S. 247–261

[Fagin und Wimmers 2000] FAGIN, R. ; WIMMERS, E. L.: A Formula for Incorporating Weights into Scoring Rules. In: *Theorectical Computer Science* 239 (2000), S. 309–338

[Ferber 2003] FERBER, R.: *Information Retrieval - Suchmodelle und Data-Mining-Verfahren für Textsammlungen und das Web*. dpunkt Verlag, 2003

[Fühles-Ubach 1997] FÜHLES-UBACH, S.: *Analysen zur Unschärfe in Datenbank- und Retrievalsystemen unter besonderer Berücksichtigung der Redundanz*, Humboldt-Universität Berlin, Dissertation, 1997

[Fishburn 1999] FISHBURN, P. C.: Preference Structures and Their Numerical Representations. In: *Theoretical Computer Science* 217 (1999), Nr. 2, S. 359–383

[Frank 1992] FRANK, A.U.: Qualitative spatial reasoning about distances and directions in geographic space. In: *Journal of Visual Languages and Computing* 3 (1992), Nr. 3, S. 343–371

[Fuhr 1996] FUHR, N.: Skriptum Information Retrieval (Lecture Notes on Information Retrieval) / Universität Dortmund. 1996. – Skript zur Vorlesung

[Fuhr und Grosjohann 2001] FUHR, N. ; GROSJOHANN, K.: XIRQL: A Query Language for Information Retrieval in XML Documents. In: *Proceedings of the 24th Annual International Conference on Research and Development in Information Retrieval*, 2001, S. 172–180

[Fuhr u. a. 1998] FUHR, N. ; GÖVERT, N. ; RÖLLEKE, T.: DOLORES: A System for Logic-Based Retrieval of Multimedia Objects. In: *SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM Press, 1998, S. 257–265

[Gelder und Topor 1991] GELDER, A. V. ; TOPOR, R. W.: Safety and Translation of Relational Calculus Queries. In: *ACM Transactions on Database Systems* 16 (1991), Nr. 2, S. 235–278

[Güntzer u. a. 2000] GÜNTZER, U. ; BALKE, W.-T. ; KIESSLING, W.: Optimizing Multi-Feature Queries for Image Databases. In: *Proceedings of the 26th International Conference on Very Large Data Bases (VLDB'00)*, 2000, S. 419–428

[Gogolla 1994] GOGOLLA, M.: *An Extended Entity-Relationship Model - Fundamentals and Pragmatics*. Springer-Verlag, 1994

[Gogolla und Hohenstein 1991] GOGOLLA, M. ; HOHENSTEIN, U.: Towards a Semantic View of an Extended Entity-Relationship Model. In: *ACM Transactions on Database Systems (TODS)* 16 (1991), Nr. 3, S. 369–416

[Govindarajan u. a. 2000] GOVINDARAJAN, K. ; JAYARAMAN, B. ; MANTHA, S.: Preference Queries in Deductive Databases. In: *New Generation Computing* 19 (2000), Nr. 1, S. 57–86

[Greenblatt und Waxman 1978] GREENBLATT, D. ; WAXMAN, J.: A Study of Three Database Languages. In: SHNEIDERMAN, B. (Hrsg.): *Proceedings of the International Conference on Databases*, Academic Press, 1978, S. 77–97

[Gudivada und Raghavan 1995] GUDIVADA, V.N. ; RAGHAVAN, V.V.: Content-based Image Retrieval Systems: Guest Editors' Introduction. In: *IEEE Computer* 28 (1995), Nr. 9, S. 18–22

[Guttman 1984] GUTTMAN, A.: R-Trees: A Dynamic Index Structure for Spatial Searching. In: YORMARK, B. (Hrsg.): *SIGMOD'84, Proceedings of Annual Meeting*, ACM Press, 1984, S. 47–57

[Hebert u. a. 1999] HEBERT, J. ; MAVRIS, S. ; MULLEN, D. ; ROWELL, T. ; ANNAMALAI, M. ; CHOPRA, R. ; SHEPARD, S. ; WARD, Rod: *Oracle8i Visual Information Retrieval User's Guide and Reference*. 1999

[Henrich 1995] HENRICH, A.: POQL: an OQL-oriented Query Language for PCTE. In: *Proceedings of the 7th Conference on Software Engineering Environments*, 1995, S. 48–60

[Henrich und Robbert 2001] HENRICH, A. ; ROBBERT, G.: POQL^{MM}: A Query Language for Structured Multimedia Documents. In: *Proceedings of the 1th International Workshop on Multimedia and Document Engineering (MDDE'01)*, 2001, S. 17–26

[Henrich und Robbert 2003] HENRICH, A. ; ROBBERT, G.: Ein Ansatz zur Übertragung von Rangordnungen bei der Suche auf strukturierten Daten. In: WEIKUM, G. (Hrsg.) ; SCHÖNING, H. (Hrsg.) ; RAHM, E. (Hrsg.): *Datenbanksysteme in Büro, Technik und Wissenschaft (BTW'03), GI-Fachtagung* Bd. 26, GI, 2003, S. 167–186

[Herrera-Viedma 2001a] HERRERA-VIEDMA, E.: Controlling the Retrieval of an Ordinal Fuzzy Linguistic Information Retrieval System Using a Multi-Level Weighting Scheme in the Formulation of Queries / Department of Computer Science and A.I., University of Granada. 2001. – Technical Report, DECSAI-010101

[Herrera-Viedma 2001b] HERRERA-VIEDMA, E.: An Information Retrieval System with Ordinal Linguistic Weighted Queries Based on Two Weighting Elements. In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 9 (2001), S. 77–88

[Herrera-Viedma 2001c] HERRERA-VIEDMA, E.: Modeling the retrieval process for an information retrieval system using an ordinal fuzzy linguistic approach. In: *Journal of the American Society of Information Science* 52 (2001), Nr. 6, S. 460–475

[Herrtwich und Delgrossi 1990] HERRTWICH, R.G. ; DELGROSSI, L.: ODA-Based Data Modeling in Multimedia Systems / International Computer Science Institute,Berkley. 1990. – Technical Report, tr-90-043

[Heuer und Saake 2000] HEUER, A. ; SAAKE, G.: *Datenbanken - Konzepte und Sprachen*. 2. International Thomson Publishing, 2000

[Heuer und Scholl 1991] HEUER, A. ; SCHOLL, M. H.: Principles of Object-Oriented Query Languages. In: APPELRATH, H.-J. (Hrsg.): *Datenbanksysteme in Büro, Technik und Wissenschaft (BTW'91), GI-Fachtagung* Bd. 270, Springer Verlag, 1991, S. 178–197

[Hibino und Rundensteiner 1996] HIBINO, S. ; RUNDENSTEINER, E.A.: A Visual Multimedia Query Language for Temporal Analysis of Video Data. In: *Multimedia Database Systems* (1996), S. 123–159

[Hirzalla und Karmouch 1996] HIRZALLA, N. ; KARMOUCH, A.: A Multimedia Query Specification Language. In: NWOSU, K. C. (Hrsg.) ; THURAISINGHAM, B. M. (Hrsg.) ; BERRA, P. B. (Hrsg.): *Multimedia Database Systems: Design and Implementation Strategies*. Kluwer Academic Publishers, 1996, S. 160–184

[Hohenstein 1993] HOHENSTEIN, U.: *Formale Semantik eines erweiterten Entity-Relationship-Modells*. Teubner-Verlag, 1993

[Hohenstein und Gogolla 1988] HOHENSTEIN, U. ; GOGOLLA, M.: A Calculus for an Extended Entity-Relationship Model Incorporating Arbitrary Data Operations and Aggregate Functions. In: BATINI, C. (Hrsg.): *Entity-Relationship Approach: A Bridge to the User, Proceedings of the 7th International Conference on Enity-Relationship*, North-Holland, 1988, S. 129–148

[Hong u. a. 1995] HONG, J.-H. ; EGENHOFER, M. J. ; FRANK, A.: On the Robustness of Qualitative Distance- and Direction-Reasoning. In: PEUQUET, D. (Hrsg.): *Proceedings of the 12th International Symposium on Computer-Assisted Cartography* Bd. 4, 1995, S. 301–310

[Hristidis u. a. 2001] HRISTIDIS, V. ; KOUDAS, N. ; PAPAKONSTANTINOU, Y.: PREFER: A System for the Efficient Execution of Multi-parametric Ranked Queries. In: AREF, W. G. (Hrsg.): *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data*, ACM, 2001, S. 297–306

[Informix 1999] INFORMIX: *Excalibur Image DataBlade Module User's Guide 1.2*. 1999

[Informix 2003] INFORMIX: *Informix DataBlades modules*. http://www-3.ibm.com/software/data/informix/blades. 2003. – 29.10.2003

[Inselberg und Dimsdale 1990] INSELBERG, A. ; DIMSDALE, B.: Parallel Coordinates: A Tool for Visualizing Multi-Dimensional Geometry. In: *Proceedings of IEEE Visualization'90 Conference*, IEEE Computer Society Press, 1990, S. 361–375

[Jain 1996] JAIN, R.: *InfoScopes: Multimedia Information Systems*. Bd. 7. S. 217–254. In: FURHT, B. (Hrsg.): *Multimedia Systems and Techniques* Bd. 7, Kluwer Academic Publishers, 1996

[Joseph und Cardenas 1993] JOSEPH, T. ; CARDENAS, A.: PICQUERY: a High Level Query Language for Pictorial Database Management. In: *IEEE Transaction on Knowledge and Data Engineering* 14 (1993), Nr. 5, S. 630–638

[Kießling 2002] KIESSLING, W.: Foundations of Preferences in Database Systems. In: *Proceedings of the 28th International Conference on Very Large Databases (VLDB'02)*, 2002, S. 311–322

[Kießling und Hafenrichter 2003] KIESSLING, W. ; HAFENRICHTER, B.: Algebraic Optimization of Relational Preference Queries / Universität Augsburg. 2003 (Nr. 2003-1). – Technical Report

[Kießling und Köstler 2002] KIESSLING, W. ; KÖSTLER, G.: Preference SQL - Design, Implementation, Experiences. In: *Proceedings of the 28th International Conference on Very Large Databases (VLDB'02)*, 2002, S. 990–1001

[Klir und Yuan 1995] KLIR, G. J. ; YUAN, B.: *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall, 1995

[Koll und Srinivasan 1990] KOLL, M. ; SRINIVASAN, P.: Fuzzy versus Probabilistic Models for User Relevance Judgments. In: *Journal of the American Society for Information Science* 41 (1990), Nr. 4, S. 264–271

[Kung u. a. 1975] KUNG, H. T. ; LUCCIO, F. ; PREPARATA, F. P.: On Finding the Maxima of a Set of Vectors. In: *Journal of the ACM* 22 (1975), Nr. 4, S. 469–476

[Lacroix und Lavency 1987] LACROIX, M. ; LAVENCY, P.: Preferences; Putting More Knowledge into Queries. In: STOCKER, P. M. (Hrsg.) ; KENT, W. (Hrsg.) ; HAMMERSLEY, P. (Hrsg.): *Proceedings of 13th International Conference on Very Large Data Bases (VLDB'87)*, Morgan Kaufmann Publishers, 1987, S. 217–225

[Lee und Hsu 1990] LEE, S.Y. ; HSU, F. J.: 2D C-String: a New Spatial Knowledge Representation for Image Database Systems. In: *Pattern Recognition* 23 (1990), S. 1077–1088

[Li u. a. 1996] LI, J. Z. ; ÖZSU, M. T. ; SZAFRON, D.: Spatial Reasoning Rules in a Multimedia Management System. In: *International Conference on Multimedia Modeling*, 1996, S. 119–133

[Li und Candan 1998] LI, W.-S. ; CANDAN, K. S.: SEMCOG: A Hybrid Object-based Image Database System and its Modeling, Language, and Query Processing. In: *Proceedings of the 14th International Conference on Data Engineering*, IEEE Computer Society, 1998, S. 284–291

[Lin u. a. 1994] LIN, K.-I. ; JAGADISH, H. V. ; FALOUTSOS, C.: The TV-Tree: An Index Structure for High-Dimensional Data. In: *VLDB Journal* 3 (1994), Nr. 4, S. 517–542

[Losee 1998] LOSEE, R. M.: A Discipline Independent Definiton of Information. In: *Journal of the American Society for Information Science* 48 (1998), Nr. 3, S. 254–269

[Maier 1983] MAIER, D.: *The Theory of Relational Databases*. Computer Science Press, 1983

[Marcus 1996] MARCUS, S.: Querying Multimedia Database in SQL. In: SUBRAHMANIAN, V. S. (Hrsg.) ; JAJODIA, S. (Hrsg.): *Multimedia Database System: Issues and Research Direction*, 1996, S. 263–277

[Marcus und Subrahmanian 1996] MARCUS, S. ; SUBRAHMANIAN, V. S.: Foundations of Multimedia Database Systems. In: *Journal of the ACM* 43 (1996), Nr. 3, S. 474–523

[Massari u. a. 1997] MASSARI, A. ; SALADINI, L. ; HEMMJE, M. ; SISINNI, F.: Virgilio: A Non-Immersive VR System to Browse Multimedia Databases. In: *Proceedings of the International Conference on Multimedia Computing and Systems (ICMCS)*, 1997, S. 573–580

[Meyer-Wegener 2003] MEYER-WEGENER, K.: *Multimediale Datenbanken - Einsatz von Datenbanktechnik in Multimedialen-Systemen*. Bd. 2. Teubner-Verlag, 2003

[Montesi und Penzo 2000] MONTESI, D. ; PENZO, W.: Taking care of vagueness and user preferences for effective similarity queries on multimedia data. In: *Sistemi Evolui per Basi di Dati*, 2000, S. 303–316

[Mortensen und Barrett 1995] MORTENSEN, E. N. ; BARRETT, W. A.: Intelligent Scissors for Image Composition. In: *Proceedings of the 22nd Annual Conference on Computer Graphics (SIGGRAPH '95)*, 1995, S. 191–198

[Nakazato und Huang 2001] NAKAZATO, M. ; HUANG, T. S.: 3D MARS: Immersive Virtual Reality for Content-Based Image Retrieval. In: *Proceedings of 2001 IEEE International Conference on Multimedia and Expo (ICME2001)*, 2001

[Nepal u. a. 1999] NEPAL, S. ; RAMAKRISHNA, M. ; THOM, J.: A Fuzzy Object Language (FOQL) for Image Databases. In: CHEN, A. L. P. (Hrsg.) ; LOCHOVSKY, F. H. (Hrsg.): *In Procceedings of the 6th International Conference on Database Systems for Advanced Applications (DASFAA'99)*, IEEE Computer Society, 1999, S. 117–124

[Nepal und Ramakrishna 1999] NEPAL, S. ; RAMAKRISHNA, M. V.: Query Processing Issues in Image (Multimedia) Databases. In: *Proceedings of the 15th International Conference on Data Engineering (ICDE)*, IEEE Computer Society Press, 1999, S. 22–29

[Nievergelt u. a. 1984] NIEVERGELT, J. ; HINTERBERGER, H. ; SEVCIK, K. C.: The Grid File: An Adaptable, Symmetric Multikey File Structure. In: *ACM Transactions on Database Systems (TODS)* 9 (1984), Nr. 1, S. 38–71

[Nürnberger und Klose 2002] NÜRNBERGER, A. ; KLOSE, A.: Improving Clustering and Visualization of Multimedia Data Using Interactive User Feedback. In: *Proceedings of the 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2002)*, 2002, S. 993–999

[Oomoto und Tanaka 1993] OOMOTO, E. ; TANAKA, K.: OVID: Design and Implementation of a Video-Object Database System. In: *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 5 (1993), Nr. 4, S. 629–643

[Oracle 2003] ORACLE: *Oracle interMedia User's Guide and Reference Release 9.0.1 Part Number A88786-01*. 2003

[Oria u. a. 1999] ORIA, V. ; ÖZSU, M. T. ; XU, B. ; CHENG, L. I. ; IGLINSKI, P. J.: VisualMOQL: The DISIMA Visual Query Language. In: *Proceedings of the 6th IEEE International Conference on Multimedia Computing and Systems* Bd. 1, IEEE Computer Society, 1999, S. 536–542

[Ortega u. a. 1998] ORTEGA, M. ; RUI, Y. ; CHAKRABARTI, K. ; PORKAEW, K. ; MEHROTRA, S. ; HUANG, T. S.: Supporting Ranked Boolean Similarity Queries in MARS. In: *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 10 (1998), Nr. 6, S. 905–925

[Özsoyoglu u. a. 1989] ÖZSOYOGLU, G. ; MATOS, V. ; ÖZSOYOGLU, M.: Query processing techniques in the summary-table-by-example database query language. In: *ACM Transactions on Database Systems (TODS)* 14 (1989), Nr. 4, S. 526 – 573

[Özsoyoglu und Wang 1989] ÖZSOYOGLU, G. ; WANG, H.: A Relational Calculus with Set Operators, Its Safety and Equivalent Graphical Languages. In: *IEEE Transactions on Software Engineering (TSE)* 15 (1989), Nr. 9, S. 1038–1052

[Papadias und Sellis 1995] PAPADIAS, D. ; SELLIS, T.K.: A Pictorial Query-by-Example Language. In: *Journal of Visual Languages and Computing* 6 (1995), Nr. 1, S. 53–72

[Pasi 1999] PASI, G.: A logical formulation of the Boolean model and of weighted Boolean models. In: *Proceedings of the Workshop on Logical and Uncertainty Models for Information Systems (LUMIS 99)*, 1999, S. 1–11

[Paskamp 1999] PASKAMP, M.: *Vergleichende Analyse von Anfragesprachen in Multimedia-Datenbanken*, Otto-von-Guericke-Universität Magdeburg, Fakultät für Informatik, Diplomarbeit, 1999

[Petraglia u. a. 2001] PETRAGLIA, G. ; SEBILLO, M. ; TUCCI, M. ; TORTORA, G.: Virtual Images for Similarity Retrieval in Image Databases. In: *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 13 (2001), Nr. 1, S. 951–967

[Prabhakaran 1997] PRABHAKARAN, B.: *Multimedia Database Management Systems*. Kluwer Academic Publishers, 1997

[Radecki 1979] RADECKI, T.: Fuzzy Set Theoretical Approach to Document Retrieval. In: *Information Processing and Management* 15 (1979), S. 247–259

[Raghavan und Wong 1986] RAGHAVAN, V. V. ; WONG, S. K. M.: A critical analysis of vector space model for information retrieval. In: *Journal of the American Society for Information Science* 37 (1986), Nr. 5, S. 279–287

[Reisner 1988] REISNER, P.: *Query Languages*. Kap. 12, S. 257–280. In: *Handbook of Human-Computer Interaction*, Elsevier Science, 1988

[Robertson und Jones 1976] ROBERTSON, S. E. ; JONES, K. S.: Relevance weighting of search terms. In: *Journal of the American Society for Information Science 27* 27 (1976), Nr. 3, S. 129–146

[Rocchio 1971] ROCCHIO, J. J.: Relevance feedback in information retrieval. In: *The SMART Retrieval System – Experiments in Automatic Document Processing*, Prentice Hall, 1971, S. 313–323

[Rubner u. a. 2000] RUBNER, Y. ; TOMASI, C. ; GUIBAS, L. J.: The Earth Mover's Distance as a Metric for Image Retrieval. In: *International Journal of Computer Vision* 40 (2000), Nr. 2, S. 99–121

[Rui u. a. 1998a] RUI, Y. ; HUANG, T. S. ; ORTEGA, M. ; MEHROTRA, S.: Relevance feedback: A power tool in interactive content-based image retrieval. In: *IEEE Trans. on Circuits and Systems for Video Technology 8(5)*, 1998, S. 644–655

[Rui u. a. 1998b] RUI, Y. ; HUANG, T. S. ; S.MEHROTRA: Relevance Feedback Techniques in Interactive Content-Based Image Retrieval. In: SETHI, I. K. (Hrsg.) ; JAIN, R. (Hrsg.): *Storage and Retrieval for Image and Video Databases VI, SPIE Proceedings* Bd. 3312, 1998, S. 25–36

[Saake und Heuer 1999] SAAKE, G. ; HEUER, A.: *Datenbanken: Implementierungstechniken*. 1. International Thomson Publishing, 1999

[Saake und Sattler 2001] SAAKE, G. ; SATTLER, K.-U.: *Algorithmen & Datenstrukturen - Eine Einführung in Java*. dpunkt.verlag, 2001

[Saake u. a. 1997] SAAKE, G. ; SCHMITT, I. ; TÜRKER, C.: *Objektdatenbanken — Konzepte, Sprachen, Architekturen*. International Thomson Publishing, 1997

[Salton 1971] SALTON, G.: *The SMART Retrieval System – Experiments in Automatic Document Processing*. Prentice Hall, 1971

[Salton und Buckley 1990] SALTON, G. ; BUCKLEY, C.: Improving retrieval performance by relevance feedback. In: *Journal of the American Society for Information Science* 41 (1990), Nr. 4, S. 288–297

[Salton u. a. 1983a] SALTON, G. ; BUCKLEY, C. ; FOX, E. A.: Automatic Query Formulation in Information Retrieval. In: *Journal of the American Society for Information Science* 34 (1983), Nr. 4, S. 262–280

[Salton u. a. 1983b] SALTON, G. ; FOX, E.A. ; WU, H.: Extended Boolean Information Retrieval. In: *Communications of the ACM* 26 (1983), Nr. 11, S. 1022–1036

[Santini und Jain 1999] SANTINI, S. ; JAIN, R.: The El Niño Image Database System. In: *Proceedings of the IEEE International Conference on Multimedia Computing and Systems (ICMCS) Vol. 1*, 1999, S. 524–529

[Santini und Jain 2000] SANTINI, S. ; JAIN, R.: Integrated Browsing and Querying for Image Databases. In: *IEEE MultiMedia* 7 (2000), Nr. 3, S. 26–39

[Sattler u. a. 2002] SATTLER, K.-U. ; SCHALLEHN, E. ; SCHMITT, I. ; SCHULZ, N.: Das Projekt LostArt.De. In: *Magdeburger Wissenschaftsjournal* 7 (2002), Nr. 2, S. 11–18

[Scharnofske 1996] SCHARNOFSKE, A.: *Eine Erweiterung der relationalen Anfragsprache QBE*, Institut für Informatik, Universität Hannover, Diplomarbeit, 1996

[Scharnofske u. a. 1997] SCHARNOFSKE, A. ; LIPECK, U.W. ; GERTZ, M.: SubQuery-By-Example: Eine orthogonale Erweiterung von QBE. In: DITTRICH, K. R. (Hrsg.) ; GEPPERT, A. (Hrsg.): *Datenbanksysteme in Büro, Technik und Wissenschaft (BTW'97), GI-Fachtagung*, Springer Verlag, 1997 (Informatik Aktuell), S. 132–152

[Schmitt] SCHMITT, I.: *Multimedia-Datenbanken*. – erscheint vorraussichtlich 2004

[Schmitt und Schulz 2004a] SCHMITT, I. ; SCHULZ, N.: Safe Reduction of Similarity Calculus to Similarity Algebra / Fakultät für Informatik, Universität Magdeburg. 2004. – Technical Report. wird veröffentlicht

[Schmitt und Schulz 2004b] SCHMITT, I. ; SCHULZ, N.: Similarity Relational Calculus and its Reduction to a Similarity Algebra. In: *International Symposium on Foundations of Information and Knowledge Systems (FoIKS)*, Springer-Verlag, 2004 (LNCS). – to appear

[Schmitt u. a. 2002] SCHMITT, I. ; SCHULZ, N. ; SAAKE, G.: Multi-Level Weighting in Multimedia Retrieval Systems. In: *Proceedings of the 2nd Int. Workshop on Multimedia Data Document Engineering (MDDE'02)*, Springer Verlag, 2002 (LNCS 2490), S. 353–364

[Schäuble 1997] SCHÄUBLE, P.: *Multimedia Information Retrieval - Content-Based Information Retrieval from Large Text and Audio Databases*. Kluwer Academic Publishers, 1997

[Schulz und Schmitt 2002] SCHULZ, N. ; SCHMITT, I.: A Survey of Weighted Scoring Rules in Multimedia Database Systems / Otto-von-Guericke Univeristät, Magdeburg. 2002. – Technical Report, Preprint Nr. 7

[Schulz und Schmitt 2003a] SCHULZ, N. ; SCHMITT, I.: Logical Transformation Rules for Complex Weighted Queries / Otto-von-Guericke Univeristät, Magdeburg. 2003. – Technical Report, Preprint Nr. 3

[Schulz und Schmitt 2003b] SCHULZ, N. ; SCHMITT, I.: Relevanzgewichtung in komplexen Ähnlichkeitsanfragen. In: WEIKUM, G. (Hrsg.) ; SCHÖNING, H. (Hrsg.) ; RAHM, E. (Hrsg.): *Datenbanksysteme in Büro, Technik und Wissenschaft (BTW'03), GI-Fachtagung* Bd. 26, GI, 2003, S. 187–196

[Schulz und Schmitt 2003c] SCHULZ, N. ; SCHMITT, I.: Transforming Multi-Level Weighted Multi-Media Queries. In: *Proceedings of the 9th International Workshop on Multimedia Information Systems (MIS'03)*, 2003, S. 97–106

[Sellis u. a. 1987] SELLIS, T. K. ; ROUSSOPOULOS, N. ; FALOUTSOS, C.: The R+-Tree: A Dynamic Index for Multi-Dimensional Objects. In: STOCKER, P. M. (Hrsg.) ; KENT, W. (Hrsg.) ; HAMMERSLEY, P. (Hrsg.): *Proceedings of 13th International Conference on Very Large Data Bases (VLDB'87)*, Morgan Kaufmann Publishers, 1987, S. 507–518

[Shedroff 1999] SHEDROFF, N.: Information Interaction Design: A Unified Field Theory of Design. In: JACOBSON, Robert (Hrsg.): *Information Design*. MIT Press, 1999, Kap. 11, S. 267–292

[Smeaton und Over 2002] SMEATON, A. F. ; OVER, P.: The TREC-2002 Video Track Report. In: VOORHEES, E. M. (Hrsg.) ; BUCKLAND, L. P. (Hrsg.): *NIST Special Publication: SP 500-251 The Eleventh Text Retrieval Conference (TREC 2002)*, 2002

[Soffer u. a. 1998] SOFFER, A. ; SAMET, H. ; ZOTKIN, D.: Pictorial query trees for query specification in image databases. In: *Proceedings of the 14th International Conference on Pattern Recognition*, 1998, S. 919–921

[Steinmetz 1990] STEINMETZ, R.: Synchronization Properties in Multimedia Systems. In: *IEEE Journal on Selected Areas in Communications* 8 (1990), Nr. 3, S. 401–412

[Steinmetz 1999] STEINMETZ, R.: *Multimedia-Technologie, Grundlagen, Komponenten und Systeme*. 2. Springer Verlag, 1999

[Subrahmanian 1998] SUBRAHMANIAN, V. S.: *Principles of multimedia database systems*. Morgan Kaufmann Publishers, 1998

[Sung 1998] SUNG, S.Y.: A Linear Transform Scheme for Combining Weights into Scores / Rice University. 1998. – Technical Report

[Tan u. a. 2001] TAN, K.-L. ; ENG, P.-K. ; OOI, B. C.: Efficient Progressive Skyline Computation. In: APERS, P. M. G. (Hrsg.) ; ATZENI, P. (Hrsg.) ; CERI, S. (Hrsg.) ; PARABOSCHI, S. (Hrsg.) ; RAMAMOHANARAO, K. (Hrsg.) ; SNODGRASS, R. T. (Hrsg.): *Proceedings of the 27th International Conference on Very Large Data Bases (VLDB'01)*, Morgan Kaufmann Publishers, 2001, S. 301–310

[Thomas und Gould 1975] THOMAS, J. C. ; GOULD, J. D.: A Psychological Study of Query By Example. In: *AFIPS National Computer Conference*, AFIPS Press, 1975, S. 439–445

[Tian und Taylor 2000] TIAN, G. Y. ; TAYLOR, D.: Color Image Retrieval Using Virtual Reality. In: *Proceedings of the International Conference on Information Visualisation (IV 2000)*, IEEE Computer Society, 2000, S. 221–225

[Topor 1987] TOPOR, R.W.: Domain-independent formulas and databases. In: *Theoretical Computer Science* 52 (1987), Nr. 3, S. 281–306

[Torlone und Ciaccia 2002a] TORLONE, R. ; CIACCIA, P.: Finding the Best when it's a Matter of Preference. In: CIACCIA, P. (Hrsg.) ; RABITTI, F. (Hrsg.) ; SODA, G. (Hrsg.): *Proceedings of the 10th Italian National Conference on Advanced Data Base Systems (SEBD 2002)*, 2002, S. 347–360

[Torlone und Ciaccia 2002b] TORLONE, R. ; CIACCIA, P.: Which Are My Preferred Items? In: *Workshop on Recommendation and Personalization in eCommerce*, 2002

[Ullman 1982] ULLMAN, J.D.: *Principles of Database Systems*. Computer Science Press, 1982

[Ullman 1988] ULLMAN, J.D.: *Principles of Database and Knowledge-Base Systems*. Bd. 1: Classical database systems. Computer Science Press, 1988

[Urban 2003] URBAN, C.: *Ähnlichkeitssuche in der Lost Art Internet Database*. Studienarbeit an der Otto-von-Guericke Universität, Magdeburg. 2003

[Voorhees und Buckland 2002] VOORHEES, E. M. (Hrsg.) ; BUCKLAND, L. P. (Hrsg.): *NIST Special Publication: SP 500-251 The Eleventh Text REtrieval Conference (TREC 2002)*. Department of Commerce, National Institute of Standards and Technology, 2002

[Waller und Kraft 1979] WALLER, W. G. ; KRAFT, D. H.: A mathematical model for a weighted Boolean retrieval system. In: *Information Processing and Management* 15 (1979), Nr. 5, S. 235–245

[Womser-Hacker 1989] WOMSER-HACKER, C.: *Der PADOK-Retrievaltest.Zur Methode und Verwendung statistischer Verfahren bei der Bewertung von Information-Retrieval-Systemen*, Universität Regensburg, Linguistische Informationswissenschaft, Dissertation, 1989

[Yager 1987] YAGER, R. R.: A note on weighted queries in information retrieval systems. In: *Journal of the American Society for Information Science* 38 (1987), S. 23–24

[Yager 1988] YAGER, R. R.: On Ordered Weighted Averaging Aggregation Operators in Multicriteria Decision Making. In: *IEEE Transactions on Systems, Man, and Cybernetics* 18 (1988), Nr. 1, S. 183–190

[Yager 1991] YAGER, R. R.: Connectives and quantifiers in fuzzy sets. In: *Fuzzy Sets and Systems* 40 (1991), S. 39–75

[Yager 1993] YAGER, R. R.: Families of OWA operators. In: *Fuzzy Sets and Systems* 59 (1993), S. 125–148

[Yazici und George 1999] YAZICI, A. ; GEORGE, R.: *Studies in Fuzziness and Soft Computing*. Bd. 26: *Fuzzy Database Modeling*. Physica-Verlag, 1999

[Yen und Scamell 1993] YEN, M. Y.-M. ; SCAMELL, R. W.: A Human Factors Experimental Comparison of SQL and QBE. In: *IEEE Transactions on Software Engineering (TSE)* 19 (1993), Nr. 4, S. 390–409

[Yin 1994] YIN, R. K.: *Applied Social Research Methods Series*. Bd. 5: *Case Study Research: Design and Methods*. 2. SAGE Publications, Thousand Oaks, 1994

[Zadeh 1965] ZADEH, L.: Fuzzy Sets. In: *Information and Control* 8 (1965), S. 338–353

[Zenner u. a. 1985] ZENNER, R. B. R. ; DECALUWE, R. M. M. ; KERRE, E. E.: A new approach to information retrieval systems using fuzzy expressions. In: *Fuzzy Sets and Systems* 17 (1985), Nr. 1, S. 9–22

[Zimmermann 1990] ZIMMERMANN, H.-J.: *Fuzzy Sets Theory - and Its Applications*. Kluwer Academic Publishers, 1990

[Zimmermann und Zysno 1980] ZIMMERMANN, H.-J. ; ZYSNO, P.: Latent Connectives in Human Decision Making. In: *Fuzzy Sets and Systems* 4 (1980), S. 37–51

[Zimmermann und Zysno 1983] ZIMMERMANN, H.-J. ; ZYSNO, P.: Descisions and Evaluations by Hierarchical Aggregation of Information. In: *Fuzzy Sets and Systems* 10 (1983), S. 243–266

[Zloof 1975] ZLOOF, M. M.: Query By Example. In: *AFIPS National Computer Conference* Bd. 44, AFIPS Press, 1975, S. 431–438

[Zloof 1977] ZLOOF, M. M.: Query By Example: A data base language. In: *IBM System Journal* 16 (1977), Nr. 4, S. 324–343

Anhang

A Semantik der Ähnlichkeitsalgebra

Definition A.1. Die Semantik des *gewichteten Durchschnitts* $E = (\cap_{(\theta_1,\theta_2,\ldots,\theta_m)} E_1, E_2, \ldots, E_m)$ ist definiert als:

$$
\begin{array}{l}
I^*((\cap_{(\theta_1,\theta_2,\ldots\theta_m)} E_1, E_2, \ldots, E_m)) = \\
\{\star_1 | (v_0, v_1, \ldots, v_k) \in I^*(E_1) \wedge (w_0, v_1, \ldots, v_k) \in I^*(E_2) \wedge \ldots \wedge (u_0, v_1, \ldots, v_k) \in I^*(E_m)\} \cup \\
\{\star_2 | (v_0, v_1, \ldots, v_k) \in I^*(E_1) \wedge (w_0, v_1, \ldots, v_k) \in I^*(E_2) \wedge \ldots \wedge \forall u_0.(u_0, v_1, \ldots, v_k) \notin I^*(E_m)\} \cup \\
\{\star_3 | (v_0, v_1, \ldots, v_k) \in I^*(E_1) \wedge \forall w_0.(w_0, v_1, \ldots, v_k) \notin I^*(E_2) \wedge \ldots \wedge (u_0, v_1, \ldots, v_k) \in I^*(E_m)\} \cup \\
\ldots \cup \\
\{\star_4 | (v_0, v_1, \ldots, v_k) \in I^*(E_1) \wedge \forall w_0.(w_0, v_1, \ldots, v_k) \notin I^*(E_2) \wedge \ldots \wedge \forall u_0.(u_0, v_1, \ldots, v_k) \notin I^*(E_m)\}
\end{array}
$$

mit

$$
\begin{array}{rcl}
\star_1 & = & (I(\wedge_{(\theta_1,\theta_2,\ldots\theta_m)})(I(\theta_1), I(\theta_2), \ldots, I(\theta_m), v_0, w_0, \ldots, u_0), v_1, \ldots, v_k) \\
\star_2 & = & (I(\wedge_{(\theta_1,\theta_2,\ldots\theta_m)})(I(\theta_1), I(\theta_2), \ldots, I(\theta_m), v_0, w_0, \ldots, 0), v_1, \ldots, v_k) \\
\star_3 & = & (I(\wedge_{(\theta_1,\theta_2,\ldots\theta_m)})(I(\theta_1), I(\theta_2), \ldots, I(\theta_m), v_0, 0, \ldots, u_0), v_1, \ldots, v_k) \\
\star_4 & = & (I(\wedge_{(\theta_1,\theta_2,\ldots\theta_m)})(I(\theta_1), I(\theta_2), \ldots, I(\theta_m), v_0, 0, \ldots, 0), v_1, \ldots, v_k).
\end{array}
$$

□

Definition A.2. Die Semantik der *gewichteten Vereinigung* $E = (\cup_{(\theta_1,\theta_2,\ldots,\theta_m)} E_1, E_2, \ldots E_m)$ ist wie folgt definiert:

$$
\begin{array}{l}
I^*(\cup_{(\theta_1,\theta_2,\ldots,\theta_m)} E_1, E_2, \ldots E_m) = \\
\{\star_1 | (v_0, v_1, \ldots, v_k) \in I^*(E_1) \wedge (w_0, v_1, \ldots, v_k) \in I^*(E_2) \wedge \ldots \wedge (u_0, v_1, \ldots, v_k) \in I^*(E_m)\} \cup \\
\{\star_2 | (v_0, v_1, \ldots, v_k) \in I^*(E_1) \wedge (w_0, v_1, \ldots, v_k) \in I^*(E_2) \wedge \ldots \wedge \forall u_0.(u_0, v_1, \ldots, v_k) \notin I^*(E_m)\} \cup \\
\{\star_3 | (v_0, v_1, \ldots, v_k) \in I^*(E_1) \wedge \forall w_0.(w_0, v_1, \ldots, v_k) \notin I^*(E_2) \wedge \ldots \wedge (u_0, v_1, \ldots, v_k) \in I^*(E_m)\} \cup \\
\{\star_4 | \forall v_0.(v_0, v_1, \ldots, v_k) \notin I^*(E_1) \wedge (w_0, v_1, \ldots, v_k) \in I^*(E_2) \wedge \ldots \wedge (u_0, v_1, \ldots, v_k) \in I^*(E_m)\} \cup \\
\ldots \cup \\
\{\star_5 | (v_0, v_1, \ldots, v_k) \in I^*(E_1) \wedge \forall w_0.(w_0, v_1, \ldots, v_k) \notin I^*(E_2) \wedge \ldots \wedge \forall u_0.(u_0, v_1, \ldots, v_k) \notin I^*(E_m)\} \cup \\
\{\star_6 | \forall v_0.(v_0, v_1, \ldots, v_k) \notin I^*(E_1) \wedge (w_0, v_1, \ldots, v_k) \in I^*(E_2) \wedge \ldots \wedge \forall u_0.(u_0, v_1, \ldots, v_k) \notin I^*(E_m)\} \cup \\
\ldots \cup \\
\{\star_7 | \forall v_0.(v_0, v_1, \ldots, v_k) \notin I^*(E_1) \wedge \forall w_0.(w_0, v_1, \ldots, v_k) \notin I^*(E_2) \wedge \ldots \wedge (u_0, v_1, \ldots, v_k) \in I^*(E_m)\}
\end{array}
$$

mit

$$\begin{aligned}
\star_1 &= (I(\vee_{(\theta_1,\theta_2,\ldots,\theta_m)}))(I(\theta_1), I(\theta_2), \ldots, I(\theta_m), v_0, w_0, \ldots, u_0), v_1, \ldots, v_k) \\
\star_2 &= (I(\vee_{(\theta_1,\theta_2,\ldots,\theta_m)}))(I(\theta_1), I(\theta_2), \ldots, I(\theta_m), v_0, w_0, \ldots, 0), v_1, \ldots, v_k) \\
\star_3 &= (I(\vee_{(\theta_1,\theta_2,\ldots,\theta_m)}))(I(\theta_1), I(\theta_2), \ldots, I(\theta_m), v_0, 0, \ldots, u_0), v_1, \ldots, v_k) \\
\star_4 &= (I(\vee_{(\theta_1,\theta_2,\ldots,\theta_m)}))(I(\theta_1), I(\theta_2), \ldots, I(\theta_m), 0, w_0, \ldots, u_0), v_1, \ldots, v_k) \\
\star_5 &= (I(\vee_{(\theta_1,\theta_2,\ldots,\theta_m)}))(I(\theta_1), I(\theta_2), \ldots, I(\theta_m), v_0, 0, \ldots, 0), v_1, \ldots, v_k) \\
\star_6 &= (I(\vee_{(\theta_1,\theta_2,\ldots,\theta_m)}))(I(\theta_1), I(\theta_2), \ldots, I(\theta_m), 0, w_0, \ldots, 0), v_1, \ldots, v_k) \\
\star_7 &= (I(\vee_{(\theta_1,\theta_2,\ldots,\theta_m)}))(I(\theta_1), I(\theta_2), \ldots, I(\theta_m), 0, 0, \ldots, u_0), v_1, \ldots, v_k).
\end{aligned}$$

□

Definition A.3. Die Semantik der *Kombination* $E = (\uplus^z E_1, E_2, \ldots E_m)$ ist definiert als:

$I^*(\uplus^z E_1, E_2, \ldots E_m) =$
$\{\star_1 | (v_0, v_1, \ldots, v_k) \in I^*(E_1) \wedge (w_0, v_1, \ldots, v_k) \in I^*(E_2) \wedge \ldots \wedge (u_0, v_1, \ldots, v_k) \in I^*(E_m)\} \cup$
$\{\star_2 | (v_0, v_1, \ldots, v_k) \in I^*(E_1) \wedge (w_0, v_1, \ldots, v_k) \in I^*(E_2) \wedge \ldots \wedge \forall u_0.(u_0, v_1, \ldots, v_k) \notin I^*(E_m)\} \cup$
$\{\star_3 | (v_0, v_1, \ldots, v_k) \in I^*(E_1) \wedge \forall w_0.(w_0, v_1, \ldots, v_k) \notin I^*(E_2) \wedge \ldots \wedge (u_0, v_1, \ldots, v_k) \in I^*(E_m)\} \cup$
$\{\star_4 | \forall v_0.(v_0, v_1, \ldots, v_k) \notin I^*(E_1) \wedge (w_0, v_1, \ldots, v_k) \in I^*(E_2) \wedge \ldots \wedge (u_0, v_1, \ldots, v_k) \in I^*(E_m)\} \cup$
$\ldots \cup$
$\{\star_5 | (v_0, v_1, \ldots, v_k) \in I^*(E_1) \wedge \forall w_0.(w_0, v_1, \ldots, v_k) \notin I^*(E_2) \wedge \ldots \wedge \forall u_0.(u_0, v_1, \ldots, v_k) \notin I^*(E_m)\} \cup$
$\{\star_6 | \forall v_0.(v_0, v_1, \ldots, v_k) \notin I^*(E_1) \wedge (w_0, v_1, \ldots, v_k) \in I^*(E_2) \wedge \ldots \wedge \forall u_0.(u_0, v_1, \ldots, v_k) \notin I^*(E_m)\} \cup$
$\ldots \cup$
$\{\star_7 | \forall v_0.(v_0, v_1, \ldots, v_k) \notin I^*(E_1) \wedge \forall w_0.(w_0, v_1, \ldots, v_k) \notin I^*(E_2) \wedge \ldots \wedge (u_0, v_1, \ldots, v_k) \in I^*(E_m)\}$

mit

$$\begin{aligned}
\star_1 &= (I(\oplus^z))(v_0, w_0, \ldots, u_0), v_1, \ldots, v_k) \\
\star_2 &= (I(\oplus^z))(v_0, w_0, \ldots, 0), v_1, \ldots, v_k) \\
\star_3 &= (I(\oplus^z))(v_0, 0, \ldots, u_0), v_1, \ldots, v_k) \\
\star_4 &= (I(\oplus^z))(0, w_0, \ldots, u_0), v_1, \ldots, v_k) \\
\star_5 &= (I(\oplus^z))(v_0, 0, \ldots, 0), v_1, \ldots, v_k) \\
\star_6 &= (I(\oplus^z))(0, w_0, \ldots, 0), v_1, \ldots, v_k) \\
\star_7 &= (I(\oplus^z))(0, 0, \ldots, u_0), v_1, \ldots, v_k).
\end{aligned}$$

□

Definition A.4. Die Semantik der *gewichteten Kombination* $E = (\uplus^z_{(\theta_1,\theta_2,\ldots,\theta_m)} E_1, E_2, \ldots E_m)$ ist definiert als:

$I^*(\uplus^z_{(\theta_1,\theta_2,\ldots,\theta_m)} E_1, E_2, \ldots E_m) =$
$\{\star_1 | (v_0, v_1, \ldots, v_k) \in I^*(E_1) \wedge (w_0, v_1, \ldots, v_k) \in I^*(E_2) \wedge \ldots \wedge (u_0, v_1, \ldots, v_k) \in I^*(E_m)\} \cup$
$\{\star_2 | (v_0, v_1, \ldots, v_k) \in I^*(E_1) \wedge (w_0, v_1, \ldots, v_k) \in I^*(E_2) \wedge \ldots \wedge \forall u_0.(u_0, v_1, \ldots, v_k) \notin I^*(E_m)\} \cup$
$\{\star_3 | (v_0, v_1, \ldots, v_k) \in I^*(E_1) \wedge \forall w_0.(w_0, v_1, \ldots, v_k) \notin I^*(E_2) \wedge \ldots \wedge (u_0, v_1, \ldots, v_k) \in I^*(E_m)\} \cup$
$\{\star_4 | \forall v_0.(v_0, v_1, \ldots, v_k) \notin I^*(E_1) \wedge (w_0, v_1, \ldots, v_k) \in I^*(E_2) \wedge \ldots \wedge (u_0, v_1, \ldots, v_k) \in I^*(E_m)\} \cup$
$\ldots \cup$
$\{\star_5 | (v_0, v_1, \ldots, v_k) \in I^*(E_1) \wedge \forall w_0.(w_0, v_1, \ldots, v_k) \notin I^*(E_2) \wedge \ldots \wedge \forall u_0.(u_0, v_1, \ldots, v_k) \notin I^*(E_m)\} \cup$
$\{\star_6 | \forall v_0.(v_0, v_1, \ldots, v_k) \notin I^*(E_1) \wedge (w_0, v_1, \ldots, v_k) \in I^*(E_2) \wedge \ldots \wedge \forall u_0.(u_0, v_1, \ldots, v_k) \notin I^*(E_m)\} \cup$
$\ldots \cup$
$\{\star_7 | \forall v_0.(v_0, v_1, \ldots, v_k) \notin I^*(E_1) \wedge \forall w_0.(w_0, v_1, \ldots, v_k) \notin I^*(E_2) \wedge \ldots \wedge (u_0, v_1, \ldots, v_k) \in I^*(E_m)\}$

mit

$$
\begin{aligned}
\star_1 &= (I(\oplus^z_{(\theta_1,\theta_2,\ldots,\theta_m)})(I(\theta_1), I(\theta_2), \ldots, I(\theta_m), v_0, w_0, \ldots, u_0), v_1, \ldots, v_k) \\
\star_2 &= (I(\oplus^z_{(\theta_1,\theta_2,\ldots,\theta_m)})(I(\theta_1), I(\theta_2), \ldots, I(\theta_m), v_0, w_0, \ldots, 0), v_1, \ldots, v_k) \\
\star_3 &= (I(\oplus^z_{(\theta_1,\theta_2,\ldots,\theta_m)})(I(\theta_1), I(\theta_2), \ldots, I(\theta_m), v_0, 0, \ldots, u_0), v_1, \ldots, v_k) \\
\star_4 &= (I(\oplus^z_{(\theta_1,\theta_2,\ldots,\theta_m)})(I(\theta_1), I(\theta_2), \ldots, I(\theta_m), 0, w_0, \ldots, u_0), v_1, \ldots, v_k) \\
\star_5 &= (I(\oplus^z_{(\theta_1,\theta_2,\ldots,\theta_m)})(I(\theta_1), I(\theta_2), \ldots, I(\theta_m), v_0, 0, \ldots, 0), v_1, \ldots, v_k) \\
\star_6 &= (I(\oplus^z_{(\theta_1,\theta_2,\ldots,\theta_m)})(I(\theta_1), I(\theta_2), \ldots, I(\theta_m), 0, w_0, \ldots, 0), v_1, \ldots, v_k) \\
\star_7 &= (I(\oplus^z_{(\theta_1,\theta_2,\ldots,\theta_m)})(I(\theta_1), I(\theta_2), \ldots, I(\theta_m), 0, 0, \ldots, u_0), v_1, \ldots, v_k).
\end{aligned}
$$

□

B Anfragen zur Evaluierung

Im Folgenden sind die Anfragen A1 bis A4 als Anfrageausdruck im Ähnlichkeitskalkül dargestellt, wobei die Gewichte direkt angegeben sind. Die Datenkollektion mit den Testdaten (1288 Datensätze) sind in der Relation $Bilder(ID, Kategorie, Bild)$ abgespeichert.

A1-ungewichtet:

$$\{X_1, X_2, X_3 \quad | \quad (((Bilder(X_1, X_2, X_3) \wedge (X_3 \sim_{Farbe} C_{Bild3})) \wedge (X_2 = C_{Text})) \wedge (X_3 \sim_{Form} C_{Bild3}))\}$$

A1-gewichtet:

$$\{X_1, X_2, X_3 \quad | \quad (Bilder(X_1, X_2, X_3) \wedge (\oplus^{0.7}_{(0.5,0.4,0.1)}((X_3 \sim_{Farbe} C_{Bild3}), (X_2 = C_{Text}), (X_3 \sim_{Form} C_{Bild3}))))\}$$

A2-ungewichtet:

$$\{X_1, X_2, X_3 \quad | \quad (Bilder(X_1, X_2, X_3) \wedge (((X_3 \sim_{Farbe} C_{Bild6}) \wedge (X_3 \sim_{Textur} C_{Bild4})) \wedge ((X_3 \sim_{Form} C_{Bild5}) \vee (X_3 \sim_{Form} C_{Bild6}))))\}$$

A2-gewichtet:

$$\{X_1, X_2, X_3 \quad | \quad (\rightsquigarrow_{0.6}(Bilder(X_1, X_2, X_3) \wedge (((X_3 \sim_{Farbe} C_{Bild6}) \oplus^{0.6}_{(0.6,0.4)} (X_3 \sim_{Textur} C_{Bild4})) \oplus^{0.6}_{(0.8,0.2)}((X_3 \sim_{Form} C_{Bild5}) \oplus^{0.0}_{(0.6,0.4)} (X_3 \sim_{Form} C_{Bild6})))))\}$$

A3-ungewichtet:

$$\{X_1, X_2, X_3 \quad | \quad (Bilder(X_1, X_2, X_3) \wedge (((X_3 \sim_{Textur} C_{Bild18}) \wedge (X_3 \sim_{Farbe} C_{Bild18})) \vee ((X_3 \sim_{Textur} C_{Bild16}) \wedge (X_3 \sim_{Farbe} C_{Bild16}))))\}$$

A3-gewichtet:

$$\{X_1, X_2, X_3 \quad | \quad (Bilder(X_1, X_2, X_3) \wedge (((X_3 \sim_{Textur} C_{Bild18}) \oplus^{0.7}_{(0.8,0.2)} (X_3 \sim_{Farbe} C_{Bild18})) \oplus^{0.2}_{(0.6,0.4)}((X_3 \sim_{Textur} C_{Bild16}) \oplus^{0.7}_{(0.8,0.2)} (X_3 \sim_{Farbe} C_{Bild16}))))\}$$

A4-ungewichtet:

$$\{X_1, X_2, X_3 \quad | \quad (Bilder(X_1, X_2, X_3) \wedge (((X_3 \sim_{Farbe} C_{Bild21}) \wedge (X_3 \sim_{Textur} C_{Bild21})) \vee ((X_3 \sim_{Farbe} C_{Bild23}) \wedge (X_3 \sim_{Textur} C_{Bild23}))))\}$$

A4-gewichtet:

$$\{X_1, X_2, X_3 \quad | \quad (Bilder(X_1, X_2, X_3) \wedge (((X_3 \sim_{Farbe} C_{Bild21}) \oplus^{0.7}_{(0.7,0.3)} (X_3 \sim_{Textur} C_{Bild21})) \oplus^{0.2}_{(0.7,0.3)} ((X_3 \sim_{Farbe} C_{Bild23}) \oplus^{0.7}_{(0.7,0.3)} (X_3 \sim_{Textur} C_{Bild23}))))\}$$

C Beweise der Transformationsregeln

Aus Gründen der Übersichtlichkeit wird hier eine vereinfachte Schreibweise verwendet. Für $\mu_i(o)$ wird kurz μ_i geschrieben.

C.1 Idempotenz

Es gilt $x_1 = x_2 \rightarrow ((x_1, \theta_1) \wedge (x_2, \theta_2)) = x_1 = x_2$,
$x_1 = x_2 \rightarrow ((x_1, \theta_1) \vee (x_2, \theta_2)) = x_1 = x_2$.

Behauptung C.1.1 $(\mu_1 = \ldots = \mu_n) \rightarrow S_X^{\Theta} = \mu_1 = \ldots = \mu_n$

Beweis mit $S_X = S_{min}$ beziehungsweise $S_X = S_{max}$

$$\begin{aligned} S_X^{\Theta} &= (\theta_1 - \theta_2)\mu_1 + 2(\theta_2 - \theta_3)S_X(\mu_1, \mu_2) + \ldots + n\theta_n S_X(\mu_1, \ldots, \mu_n) \\ &= \theta_1\mu_1 - \theta_2\mu_1 + 2\theta_2\mu_1 - 2\theta_3\mu_1 + \ldots + n\theta_n\mu_1 \\ &= \theta_1\mu_1 + \theta_2\mu_1 + \ldots + \theta_n\mu_1 \\ &= (\theta_1 + \theta_2 + \ldots + \theta_n)\mu_1 \\ &\quad \text{mit: } \theta_1 + \theta_2 + \ldots + \theta_n = 1 \\ &= \mu_1 \end{aligned}$$

□

C.2 Involution

Es gilt: $\neg\neg(x_1, \theta_1) = (x_1, \theta_1)$.

Behauptung C.2.1 $\neg\neg(x_1, \theta_1) = (x_1, \theta_1)$

Beweis

$$\begin{aligned} \neg\neg(x_1, \theta_1) &= (\neg\neg x_1, \theta_1) \\ &\quad \text{mit: } \neg\neg x_1 = x_1 \\ &= (x_1, \theta_1) \end{aligned}$$

□

C.3 De Morgan

Es gilt: $\neg((x_1,\theta_1) \wedge (x_2,\theta_2)) = ((\neg x_1,\theta_1) \vee (\neg x_2,\theta_2))$,
$\neg((x_1,\theta_1) \vee (x_2,\theta_2)) = ((\neg x_1,\theta_1) \wedge (\neg x_2,\theta_2))$.

Es wird zwischen einer ursprünglichen konjunktiven Anfrage und einer disjunktiven Anfrage unterschieden. Die konjunktive Anfrage wird mit S_{min} und dementsprechend die disjunktive Anfrage mit S_{max} evaluiert.

Behauptung C.3.1 $\neg S^{\Theta}_{min}(\mu_1 \ldots, \mu_n, \theta_1, \ldots \theta_n) = S^{\Theta}_{max}(\neg\mu_1 \ldots, \neg\mu_n, \theta_1, \ldots \theta_n)$

Beweis

$$
\begin{aligned}
&S^{\Theta}_{max}(\neg\mu_1, \ldots, \neg\mu_n, \theta_1, \ldots, \theta_n) \\
&\quad = (\theta_1 - \theta_2)(1-\mu_1) + 2(\theta_2 - \theta_3)max(1-\mu_1, 1-\mu_2) + \ldots + n\theta_n max(1-\mu_1, \ldots, 1-\mu_n)
\end{aligned}
$$

$$
\begin{aligned}
&\neg S^{\Theta}_{min}(\mu_1, \ldots, \mu_n, \theta_1, \ldots, \theta_n) \\
&\quad = 1 - ((\theta_1 - \theta_2)min(\mu_1) + 2(\theta_2 - \theta_3)min(\mu_1, \mu_2) + \ldots + n\theta_n min(\mu_1, \ldots, \mu_n)) \\
&\qquad \text{mit: } min(\mu_1, \ldots, \mu_n) = 1 - max(1-\mu_1, \ldots, 1-\mu_n) \\
&\quad = 1 - \theta_1\mu_1 + \theta_2\mu_1 - 2\theta_2 + 2\theta_3 - 2(\theta_2 - \theta_3)max(1-\mu_1, 1-\mu_2) - \ldots - n\theta_n \\
&\qquad + n\theta_n max(1-\mu_1, \ldots, 1-\mu_n) \\
&\qquad \text{mit: } 1 = \theta_1 + \ldots + \theta_n \\
&\quad = \theta_1 - \theta_2 - \theta_1\mu_1 + \theta_2\mu_1 + 2(\theta_2 - \theta_3)max(1-\mu_1, 1-\mu_2) + \ldots \\
&\qquad + n\theta_n max(1-\mu_1, \ldots, 1-\mu_n) \\
&\quad = (\theta_1 - \theta_2)(1-\mu_1) + 2(\theta_2 - \theta_3)max(1-\mu_1, 1-\mu_2) + \ldots + n\theta_n max(1-\mu_1, \ldots, 1-\mu_n) \\
&\quad = S^{\Theta}_{max}(\neg\mu_1, \ldots, \neg\mu_n, \theta_1, \ldots, \theta_n)
\end{aligned}
$$

□

Behauptung C.3.2

$$\neg S^{\Theta}_{max}(\mu_1, \ldots, \mu_n, \theta_1, \ldots, \theta_n) = S^{\Theta}_{min}(\neg\mu_1, \ldots, \neg\mu_n, \theta_1, \ldots, \theta_n)$$

Beweis

$$\begin{aligned}
&\neg S^{\Theta}_{max}(\mu_1,\ldots,\mu_n,\theta_1,\ldots,\theta_n)\\
&\quad = 1-((\theta_1-\theta_2)max(\mu_1)+2(\theta_2-\theta_3)max(\mu_1,\mu_2)+\ldots+n\theta_n min(\mu_1,\ldots,\mu_n))\\
&\\
&S^{\Theta}_{min}(\neg\mu_1,\ldots,\neg\mu_n,\theta_1,\ldots,\theta_n)\\
&\quad = (\theta_1-\theta_2)min(1-\mu_1)+2(\theta_2-\theta_3)min(1-\mu_1,1-\mu_2)+\ldots\\
&\qquad +n\theta_n min(1-\mu_1,\ldots,1-\mu_n)\\
&\qquad \text{mit: } min(\mu_1,\ldots,\mu_n)=1-max(1-\mu_1,\ldots,1-\mu_n)(\theta_1-\theta_2)(1-max(x_1))+\\
&\qquad\qquad 2(\theta_2-\theta_3)(1-max(\mu_1,\mu_2))+\ldots+n\theta_n(1-max(\mu_1,\ldots,\mu_n))\\
&\quad = \theta_1-\theta_2-\theta_1\mu_1+\theta_2\mu_2+2\theta_2-2\theta_3-2(\theta_2-\theta_3)max(\mu_1,\mu_2)+\ldots+n\theta_n\\
&\qquad -n\theta_n max(\mu_1,\ldots,\mu_n)\\
&\quad = \theta_1-(\theta_1-\theta_2)max(\mu_1)+\theta_2-2(\theta_2-\theta_3)+\ldots+\theta_n-n\theta_n max(\mu_1,\ldots,\mu_n)\\
&\qquad \text{mit: } 1=\theta_1+\theta_2+\ldots+\theta_n\\
&\quad = 1-((\theta_1-\theta_2)max(\mu_1)+2(\theta_2-\theta_3)max(\mu_1,\mu_2)+\ldots+n\theta_n max(\mu_1,\ldots,\mu_n))\\
&\quad = \neg S^{\Theta}_{max}(\mu_1,\ldots,\mu_n,\theta_1,\ldots,\theta_n)
\end{aligned}$$

□

C.4 Distributivität

Es gilt:
$((((x_1,\theta_1)\wedge(x_2,\theta_2)),\theta_{1,1})\vee(x_3,\theta_3)) = ((((x_1,\theta_{1'})\vee(x_3,\theta_{3'})),\theta_{1,1'})\wedge(((x_2,\theta_{2'})\vee(x_3,\theta_{3''})),\theta_{1,2'}))$,
$((((x_1,\theta_1)\vee(x_2,\theta_2),\theta_{1,1}))\wedge(x_3,\theta_3)) = ((((x_1,\theta_{1'})\wedge(x_3,\theta_{3'}),\theta_{1,1'})\vee((x_2,\theta_{2'})\wedge(x_3,\theta_{3''})),\theta_{1,2'}))$.

Bei einer distributiven Transformation werden in Abhängigkeit von μ_1,μ_2,μ_3 die folgenden sechs Fälle unterschieden:

1. $\mu_1\geq\mu_2\geq\mu_3$
2. $\mu_1\geq\mu_3\geq\mu_2$
3. $\mu_2\geq\mu_1\geq\mu_3$
4. $\mu_2\geq\mu_3\geq\mu_1$
5. $\mu_3\geq\mu_1\geq\mu_2$
6. $\mu_3\geq\mu_2\geq\mu_1$

Des Weiteren wird zwischen einer *und/oder*-Anfrage ($X_{\wedge,\vee}$) und einer *oder/und*-Anfrage ($X_{\vee,\wedge}$) differenziert.Im Folgenden sind die Beweise für die distributive Transformation von X nach X' für jeden Fall angegeben. Da es sich bei diesen Anfragen um komplexe Anfragen handelt, muß die Gewichtung in expliziter Form Θ^E vorliegen.

Für eine übersichtlichere Darstellung werden die Scoring-Funktionen für die einzelnen Teilanfragen getrennt aufgeführt. So wird beispielsweise die Teilanfrage $s_{1,1}$ durch die Scoring-Funktion $S^{\Theta}_{s_{1,1}}$ evaluiert.

C.4.1 Beweise für Fall 1: $\mu_1 \geq \mu_2 \geq \mu_3$

Behauptung C.4.1 $S^{\Theta^E}_{X_{\wedge,\vee}} \Rightarrow S^{\Theta^E}_{X'}$

Beweis Für $S^{\Theta^E}_{X_{\wedge,\vee}}$ ergibt sich:

$$\begin{aligned} S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 min(\mu_1,\mu_2) \\ S^{\Theta^E}_{X_{\wedge,\vee}} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 max(S^{\Theta}_{s_{1,1}},\mu_3) \\ &= s_{1,1} \\ &= (1-2\theta_2)\mu_1 + 2\theta_2 min(\mu_1,\mu_2) \end{aligned}$$

und für $S^{\Theta^E}_{X'}$:

$$\begin{aligned} S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_1 + 2\theta_{3'} max(\mu_1,\mu_3) = \mu_1 \\ S^{\Theta}_{s_{1,2'}} &= (1-2\theta_{3''})\mu_2 + 2\theta_{3''} max(\mu_2,\mu_3) = \mu_2 \\ S^{\Theta^E}_{X'} &= (1-2\theta_{1,2'})S^{\Theta}_{s_{1,1'}} + 2\theta_{1,2'} min(S^{\Theta}_{s_{1,1'}}, S^{\Theta}_{s_{1,2'}}) \\ &= (1-2\theta_{1,2'})\mu_1 + 2\theta_{1,2'} min(\mu_1,\mu_2) \\ &\quad \text{mit } \theta_{3'} = \theta_3, \theta_{3''} = \theta_3, \theta_{1,2'} = \theta_2 \\ &= (1-2\theta_2)\mu_1 + 2\theta_2\mu_2 = S^{\Theta^E}_{X_{\wedge,\vee}} \end{aligned}$$

□

Behauptung C.4.2 $S^{\Theta^E}_{X_{\vee,\wedge}} \Rightarrow S^{\Theta^E}_{X'}$

Beweis Für $S^{\Theta^E}_{X_{\vee,\wedge}}$ ergibt sich:

$$\begin{aligned} S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 max(\mu_1,\mu_2) = \mu_1 \\ S^{\Theta^E}_{X_{\vee,\wedge}} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 min(S^{\Theta}_{s_{1,1}},\mu_3) \\ &= (1-2\theta_3)\mu_1 + 2\theta_3\mu_3 \end{aligned}$$

und für $S^{\Theta^E}_{X'}$:

$$\begin{aligned}
S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_1 + 2\theta_{3'} min(\mu_1,\mu_3) \\
S^{\Theta}_{s_{1,2'}} &= (1-2\theta_{3''})\mu_2 + 2\theta_{3''} min(\mu_2,\mu_3) \\
S^{\Theta^E}_{X'} &= (1-2\theta_{1,2'})S^{\Theta}_{s_{1,1'}} + 2\theta_{1,2'} max(S^{\Theta}_{s_{1,1'}}, S^{\Theta}_{s_{1,2'}}) \\
&= S^{\Theta}_{s_{1,1'}} \\
&\quad \text{mit } \theta_{3'} = \theta_3, \theta_{3''} = \theta_3, \theta_{1,2'} = \theta_2 \\
&= (1-2\theta_3)\mu_1 + 2\theta_3\mu_3 = S^{\Theta^E}_{X_{\vee,\wedge}}
\end{aligned}$$

□

C.4.2 Beweise für Fall 2: $\mu_1 \geq \mu_3 \geq \mu_2$

Behauptung C.4.3 $S^{\Theta^E}_{X_{\wedge,\vee}} \Rightarrow S^{\Theta^E}_{X'}$

Fall 1: $S^{\Theta}_{s_{1,1}} < \mu_3$ Für $S^{\Theta^E}_{X_{\wedge,\vee}}$ ergibt sich:

$$\begin{aligned}
S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 min(\mu_1,\mu_2) \\
&= (1-2\theta_2)\mu_1 + 2\theta_2\mu_2 \\
S^{\Theta^E}_{X_{\wedge,\vee}} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 max(S^{\Theta}_{s_{1,1}}, \mu_3) \\
&= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3\mu_3
\end{aligned}$$

Für X' ergibt sich mit $\theta_{3'} = \theta_3$, $\theta_{3''} = \frac{\theta_3}{1-(1-2\theta_2)(1-2\theta_3)}$, $\theta_{1,2'} = \frac{1-(1-2\theta_2)(1-2\theta_3)}{2}$:

$$\begin{aligned}
S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_1 + 2\theta_{3'} max(\mu_1,\mu_3) = \mu_1 \\
S^{\Theta}_{s_{1,2'}} &= (1-2\theta_{3''})\mu_2 + 2\theta_{3''} max(\mu_2,\mu_3) \\
&= (1-2\theta_{3''})\mu_2 + 2\theta_{3''}\mu_3 \\
&= \left(1 - 2\frac{\theta_3}{1-(1-2\theta_2)(1-2\theta_3)}\right)\mu_2 + 2\frac{\theta_3}{1-(1-2\theta_2)(1-2\theta_3)}\mu_3 \\
S^{\Theta^E}_{X'} &= (1-2\theta_{1,2'})S^{\Theta}_{s_{1,1'}} + 2\theta_{1,2'} min(S^{\Theta}_{s_{1,1'}}, S^{\Theta}_{s_{1,2'}}) \\
&= \left(1 - 2\frac{1-(1-2\theta_2)(1-2\theta_3)}{2}\right) S^{\Theta}_{s_{1,1'}} + 2\frac{1-(1-2\theta_2)(1-2\theta_3)}{2} S^{\Theta}_{s_{1,2'}} \\
&= (1-(1-(1-2\theta_2)(1-2\theta_3)))\mu_1 + (1-(1-2\theta_2)(1-2\theta_3))S^{\Theta}_{s_{1,2'}} \\
&= (1-2\theta_2)(1-2\theta_3)\mu_1 + (1-2\theta_3)(2\theta_2)\mu_2 + 2\theta_3\mu_3 \\
&= (1-2\theta_3)((1-2\theta_2)\mu_1 + (2\theta_2)\mu_2) + 2\theta_3 = S^{\Theta^E}_{X_{\vee,\wedge}}
\end{aligned}$$

□

Fall 2: $S^{\Theta}_{s_{1,1}} \geq \mu_3$ Für $S^{\Theta^E}_{X_{\wedge,\vee}}$ ergibt sich:

$$\begin{aligned}
S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 min(\mu_1,\mu_2) \\
&= (1-2\theta_2)\mu_1 + 2\theta_2\mu_2 \\
S^{\Theta^E}_{X_{\wedge,\vee}} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 max(S^{\Theta}_{s_{1,1}},\mu_3) \\
&= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 S^{\Theta}_{s_{1,1}} \\
&= S^{\Theta}_{s_{1,1}} \\
&= (1-2\theta_2)\mu_1 + 2\theta_2 min(\mu_1,\mu_2)
\end{aligned}$$

Für X' ergibt sich mit $\theta_{3'} = \theta_3$, $\theta_{3''} = 0$, $\theta_{1,2'} = \theta_2$

$$\begin{aligned}
S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_1 + 2\theta_{3'} max(\mu_1,\mu_3) = \mu_1 \\
S^{\Theta}_{s_{1,2'}} &= (1-2\theta_{3''})\mu_2 + 2\theta_{3''} max(\mu_2,\mu_3) \\
&= (1-2\theta_{3''})\mu_2 + 2\theta_{3''}\mu_3 \\
&= (1-2*0)\mu_2 + 2*0*\mu_3 = \mu_2 \\
S^{\Theta^E}_{X'} &= (1-2\theta_{1,2'})S^{\Theta}_{s_{1,1'}} + 2\theta_{1,2'} min(S^{\Theta}_{s_{1,1'}}, S^{\Theta}_{s_{1,2'}}) \\
&= (1-2\theta_2)\mu_1 + 2\theta_2 min(\mu_1,\mu_2) = S^{\Theta^E}_{X_{\vee,\wedge}}
\end{aligned}$$

□

Behauptung C.4.4 $S^{\Theta^E}_{X_{\vee,\wedge}} \Rightarrow S^{\Theta^E}_{X'}$

Beweis Für $S^{\Theta^E}_{X_{\vee,\wedge}}$ ergibt sich:

$$\begin{aligned}
S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 max(\mu_1,\mu_2) \\
&= \mu_1 \\
S^{\Theta^E}_{X_{\wedge,\vee}} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 min(S^{\Theta}_{s_{1,1}},\mu_3) \\
&= (1-2\theta_3)\mu_1 + 2\theta_3 min(\mu_1,\mu_3)
\end{aligned}$$

und für $S^{\Theta^E}_{X'}$:

$$\begin{aligned}
S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_1 + 2\theta_{3'} min(\mu_1,\mu_3) \\
S^{\Theta}_{s_{1,2'}} &= (1-2\theta_{3''})\mu_2 + 2\theta_{3''} min(\mu_2,\mu_3) \\
S^{\Theta^E}_{X'} &= (1-2\theta_{1,2'})S^{\Theta}_{s_{1,1'}} + 2\theta_{1,2'} max(S^{\Theta}_{s_{1,1'}}, S^{\Theta}_{s_{1,2'}}) \\
&= S^{\Theta}_{s_{1,1'}} \\
&\quad \text{mit } \theta_{3'} = \theta_3, \theta_{3''} = \theta_3, \theta_{1,2'} = \theta_2 \\
&= (1-2\theta_3)\mu_1 + 2\theta_3 min(\mu_1,\mu_3) = S^{\Theta^E}_{X_{\wedge,\vee}}
\end{aligned}$$

□

C.4.3 Beweise für Fall 3: $\mu_2 \geq \mu_1 \geq \mu_3$

Behauptung C.4.5 $S^{\Theta^E}_{X_{\wedge,\vee}} \Rightarrow S^{\Theta^E}_{X'}$

Beweis Für $S^{\Theta^E}_{X_{\wedge,\vee}}$ ergibt sich:

$$\begin{aligned} S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 min(\mu_1,\mu_2) = \mu_1 \\ S^{\Theta^E}_{X_{\wedge,\vee}} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 max(S^{\Theta}_{s_{1,1}},\mu_3) \\ &= ((1-2\theta_3)\mu_1 + 2\theta_3 max(\mu_1,\mu_3) = \mu_1 \end{aligned}$$

und für $S^{\Theta^E}_{X'}$:

$$\begin{aligned} S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_1 + 2\theta_{3'} max(\mu_1,\mu_3) = \mu_1 \\ S^{\Theta}_{s_{1,2'}} &= (1-2\theta_{3''})\mu_2 + 2\theta_{3''} max(\mu_2,\mu_3) = \mu_2 \\ S^{\Theta^E}_{X'} &= (1-2\theta_{1,2'})S^{\Theta}_{s_{1,1'}} + 2\theta_{1,2'} min(S^{\Theta}_{s_{1,1'}}, S^{\Theta}_{s_{1,2'}}) \\ &= (1-2\theta_{1,2'})\mu_1 + 2\theta_{1,2'} min(\mu_1,\mu_2) \\ &\quad \text{mit } \theta_{3'} = \theta_3, \theta_{3''} = \theta_3, \theta_{1,2'} = \theta_2 \\ &= \mu_1 = S^{\Theta^E}_{X_{\wedge,\vee}} \end{aligned}$$

□

Behauptung C.4.6 $S^{\Theta^E}_{X_{\vee,\wedge}} \Rightarrow S^{\Theta^E}_{X'}$

Beweis Für $S^{\Theta^E}_{X_{\vee,\wedge}}$ ergibt sich:

$$\begin{aligned} S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 max(\mu_1,\mu_2) \\ S^{\Theta^E}_{X_{\vee,\wedge}} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 min(S^{\Theta}_{s_{1,1}},\mu_3) \\ &= (1-2\theta_3)S_{s_{1,1}} + 2\theta_3\mu_3 \\ &= (1-2\theta_3)((1-2\theta_2)\mu_1 + 2\theta_2\mu_2) + 2\theta_3\mu_3 \end{aligned}$$

und für $S^{\Theta^E}_{X'}$:

$$\begin{aligned} S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_1 + 2\theta_{3'} min(\mu_1,\mu_3) \\ S^{\Theta}_{s_{1,2'}} &= (1-2\theta_{3''})\mu_2 + 2\theta_{3''} min(\mu_2,\mu_3) \\ S^{\Theta^E}_{X'} &= (1-2\theta_{1,2'})S^{\Theta}_{s_{1,1'}} + 2\theta_{1,2'} max(S^{\Theta}_{s_{1,1'}}, S^{\Theta}_{s_{1,2'}}) \\ S^{\Theta^E}_{X'} &= (1-2\theta_{1,2'})S^{\Theta}_{s_{1,1'}} + 2\theta_{1,2'} S^{\Theta}_{s_{1,2'}} \\ &\quad \text{mit } \theta_{3'} = \theta_3, \theta_{3''} = \theta_3, \theta_{1,2'} = \theta_2 \\ &= (1-2\theta_2)(1-2\theta_3)\mu_2 + (1-2\theta_2)2\theta_3\mu_3 + 2\theta_2(1-2\theta_3)\mu_2 + 2\theta_2 * 2\theta_3\mu_3 \\ &= (1-2\theta_3)((1-2\theta_2)\mu_1 + 2\theta_2\mu_2) + 2\theta_3\mu_3 = S^{\Theta^E}_{X_{\vee,\wedge}} \end{aligned}$$

□

C.4.4 Beweise für Fall 4: $\mu_2 \geq \mu_3 \geq \mu_1$

Behauptung C.4.7 $S^{\Theta^E}_{X_{\wedge,\vee}} \Rightarrow S^{\Theta^E}_{X'}$

Beweis Für $S^{\Theta^E}_{X_{\wedge,\vee}}$ ergibt sich:

$$\begin{aligned} S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 min(\mu_1,\mu_2) = \mu_1 \\ S^{\Theta^E}_{X_{\wedge,\vee}} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 max(S^{\Theta}_{s_{1,1}},\mu_3) \\ &= (1-2\theta_3)\mu_1 + 2\theta_3\mu_3 \end{aligned}$$

und für $S^{\Theta^E}_{X'}$:

$$\begin{aligned} S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_1 + 2\theta_{3'} max(\mu_1,\mu_3) \\ S^{\Theta}_{s_{1,2'}} &= (1-2\theta_{3''})\mu_2 + 2\theta_{3''} max(\mu_2,\mu_3) = \mu_2 \\ S^{\Theta^E}_{X'} &= (1-2\theta_{1,2'})S^{\Theta}_{s_{1,1'}} + 2\theta_{1,2'} min(S^{\Theta}_{s_{1,1'}}, S^{\Theta}_{s_{1,2'}}) \\ &= (1-2\theta_{1,2'})S^{\Theta}_{s_{1,1'}} + 2\theta_{1,2'} min(S^{\Theta}_{s_{1,1'}}, \mu_2) \\ &= S^{\Theta}_{s_{1,1'}} \\ &\quad \text{mit } \theta_{3'} = \theta_3, \theta_{3''} = \theta_3, \theta_{1,2'} = \theta_2 \\ &= (1-2\theta_3)\mu_1 + 2\theta_3\mu_3 = S^{\Theta^E}_{X_{\wedge,\vee}} \end{aligned}$$

□

Behauptung C.4.8 $S^{\Theta^E}_{X_{\vee,\wedge}} \Rightarrow S^{\Theta^E}_{X'}$

Fall 1: $S^{\Theta}_{s_{1,1}} \leq \mu_3$ Für $S^{\Theta^E}_{X_{\vee,\wedge}}$ ergibt sich:

$$\begin{aligned} S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 max(\mu_1,\mu_2) \\ S^{\Theta^E}_{X_{\vee,\wedge}} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 min(S^{\Theta}_{s_{1,1}},\mu_3) \\ &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 S^{\Theta}_{s_{1,1}} \\ &= S^{\Theta}_{s_{1,1}} \\ &= (1-2\theta_2)\mu_1 + 2\theta_2 max(\mu_1,\mu_2) \end{aligned}$$

Für X' ergibt sich mit $\theta_{3'} = \theta_3$, $\theta_{3''} = 0$, $\theta_{1,2'} = \theta_2$:

$$
\begin{aligned}
S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_1 + 2\theta_{3'} min(\mu_1,\mu_3) = \mu_1 \\
S^{\Theta}_{s_{1,2'}} &= (1-2\theta_{3''})\mu_2 + 2\theta_{3''} min(\mu_2,\mu_3) \\
&= (1-2\theta_{3''})\mu_2 + 2\theta_{3''}\mu_3 \\
&= (1-2*0)\mu_2 + 2*0*\mu_3 = \mu_2 \\
S^{\Theta^E}_{X'} &= (1-2\theta_{1,2'})S^{\Theta}_{s_{1,1'}} + 2\theta_{1,2'} max(S^{\Theta}_{s_{1,1'}}, S^{\Theta}_{s_{1,2'}}) \\
&= (1-2\theta_2)\mu_1 + 2\theta_2 max(\mu_1,\mu_2) = S^{\Theta^E}_{X_{\vee,\wedge}}
\end{aligned}
$$

□

Fall 2: $S^{\Theta}_{s_{1,1}} > \mu_3$ Für $S^{\Theta^E}_{X_{\vee,\wedge}}$ ergibt sich:

$$
\begin{aligned}
S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 max(\mu_1,\mu_2) \\
S^{\Theta^E}_{X_{\vee,\wedge}} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 min(S^{\Theta}_{s_{1,1}},\mu_3) \\
&= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3\mu_3 \\
&= (1-2\theta_3)((1-2\theta_2)\mu_1 + 2\theta_2\mu_2) + 2\theta_3\mu_3
\end{aligned}
$$

Für X' ergibt sich mit $\theta_{3'} = \theta_3$, $\theta_{3''} = \frac{\theta_3}{1-(1-2\theta_2)(1-2\theta_3)}$, $\theta_{1,2'} = \frac{1-(1-2\theta_2)(1-2\theta_3)}{2}$:

$$
\begin{aligned}
S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_1 + 2\theta_{3'} min(\mu_1,\mu_3) = \mu_1 \\
S^{\Theta}_{s_{1,2'}} &= (1-2\theta_{3''})\mu_2 + 2\theta_{3''} min(\mu_2,\mu_3) \\
&= (1-2\theta_{3''})\mu_2 + 2\theta_{3''}\mu_3 \\
&= \left(1 - 2\frac{\theta_3}{1-(1-2\theta_2)(1-2\theta_3)}\right)\mu_2 + 2\frac{\theta_3}{1-(1-2\theta_2)(1-2\theta_3)}\mu_3 \\
S^{\Theta^E}_{X'} &= (1-2\theta_{1,2'})S^{\Theta}_{s_{1,1'}} + 2\theta_{1,2'} max(S^{\Theta}_{s_{1,1'}}, S^{\Theta}_{s_{1,2'}}) \\
&= (1-2\theta_{1,2'})\mu_1 + 2\theta_{1,2'} S^{\Theta}_{s_{1,2}} \\
&= \left(1 - 2\frac{1-(1-2\theta_2)(1-2\theta_3)}{2}\right)\mu_1 + 2\frac{1-(1-2\theta_2)(1-2\theta_3)}{2} S^{\Theta}_{s_{1,2'}} \\
&= (1-1-(1-2\theta_2)(1-2\theta_3))\mu_1 + 1 - (1-2\theta_2)(1-2\theta_3) S^{\Theta}_{s_{1,2'}} \\
&= (1-2\theta_2)(1-2\theta_3)\mu_1 + (1-2\theta_3)(2\theta_2)\mu_2 + 2\theta_3\mu_3 \\
&= (1-2\theta_3)((1-2\theta_2)\mu_1 + 2\theta_2\mu_2) + 2\theta_3 = S^{\Theta^E}_{X_{\vee,\wedge}}
\end{aligned}
$$

□

C.4.5 Beweise für Fall 5: $\mu_3 \geq \mu_1 \geq \mu_2$

Behauptung C.4.9 $S^{\Theta^E}_{X_{\wedge,\vee}} \Rightarrow S^{\Theta^E}_{X'}$

Beweis Für $S^{\Theta^E}_{X_{\wedge,\vee}}$ ergibt sich:

$$\begin{aligned} S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 min(\mu_1,\mu_2) \\ S^{\Theta^E}_{X_{\wedge,\vee}} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 max(S^{\Theta}_{s_{1,1}},\mu_3) \\ &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3\mu_3 \end{aligned}$$

und für $S^{\Theta^E}_{X'}$:

$$\begin{aligned} S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_1 + 2\theta_{3'} max(\mu_1,\mu_3) \\ S^{\Theta}_{s_{1,2'}} &= (1-2\theta_{3''})\mu_2 + 2\theta_{3''} max(\mu_2,\mu_3) \\ S^{\Theta^E}_{X'} &= (1-2\theta_{1,2'})S^{\Theta}_{s_{1,1'}} + 2\theta_{1,2'} min(S^{\Theta}_{s_{1,1'}}, S^{\Theta}_{s_{1,2'}}) \\ &= (1-2\theta_{1,2'})S^{\Theta}_{s_{1,1'}} + 2\theta_{1,2'} S^{\Theta}_{s_{1,2'}} \\ &\quad \text{mit } \theta_{3'} = \theta_3, \theta_{3''} = \theta_3, \theta_{1,2'} = \theta_2 \\ &= (1-2\theta_2)(1-2\theta_3)\mu_1 + (1-2\theta_2) * 2\theta_3\mu_3 + 2\theta_2(1-2\theta_3)\mu_2 + 2\theta_2 * 2\theta_3\mu_3 \\ &= (1-2\theta_3)((1-2\theta_2)\mu_1 + 2\theta_2\mu_2) + 2\theta_2\mu_3 = S^{\Theta^E}_{X_{\wedge,\vee}} \end{aligned}$$

□

Behauptung C.4.10 $S^{\Theta^E}_{X_{\vee,\wedge}} \Rightarrow S^{\Theta^E}_{X'}$

Beweis Für $S^{\Theta^E}_{X_{\vee,\wedge}}$ ergibt sich:

$$\begin{aligned} S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 max(\mu_1,\mu_2) = \mu_1 \\ S^{\Theta^E}_{X_{\vee,\wedge}} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 min(S^{\Theta}_{s_{1,1}},\mu_3) \\ &= S^{\Theta}_{s_{1,1}} = \mu_1 \end{aligned}$$

und für $S^{\Theta^E}_{X'}$:

$$\begin{aligned} S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_1 + 2\theta_{3'} min(\mu_1,\mu_3) = \mu_1 \\ S^{\Theta}_{s_{1,2'}} &= (1-2\theta_{3''})\mu_2 + 2\theta_{3''} min(\mu_2,\mu_3) = \mu_2 \\ S^{\Theta^E}_{X'} &= (1-2\theta_{1,2'})S^{\Theta}_{s_{1,1'}} + 2\theta_{1,2'} max(S^{\Theta}_{s_{1,1'}}, S^{\Theta}_{s_{1,2'}}) \\ &= (1-2\theta_{1,2'})\mu_1 + 2\theta_{1,2'} max(\mu_1,\mu_2) \\ &\quad \text{mit } \theta_{3'} = \theta_3, \theta_{3''} = \theta_3, \theta_{1,2'} = \theta_2 \\ &= (1-2\theta_3)\mu_1 + 2\theta_3\mu_1 = \mu_1 = S^{\Theta^E}_{X_{\vee,\wedge}} \end{aligned}$$

□

C.4.6 Beweise für Fall 6: $\mu_3 \geq \mu_2 \geq \mu_1$

Behauptung C.4.11 $S^{\Theta^E}_{X_{\wedge,\vee}} \Rightarrow S^{\Theta^E}_{X'}$

Beweis Für $S^{\Theta^E}_{X_{\wedge,\vee}}$ ergibt sich:

$$\begin{aligned} S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 min(\mu_1,\mu_2) = \mu_1 \\ S^{\Theta^E}_{X_{\wedge,\vee}} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 max(S^{\Theta}_{s_{1,1}},\mu_3) \\ &= (1-2\theta_3)\mu_1 + 2\theta_3 max(\mu_1,\mu_3) \end{aligned}$$

und für $S^{\Theta^E}_{X'}$:

$$\begin{aligned} S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_1 + 2\theta_{3'} max(\mu_1,\mu_3) \\ S^{\Theta}_{s_{1,2'}} &= (1-2\theta_{3''})\mu_2 + 2\theta_{3''} max(\mu_2,\mu_3) \\ S^{\Theta^E}_{X'} &= (1-2\theta_{1,2'})S^{\Theta}_{s_{1,1'}} + 2\theta_{1,2'} min(S^{\Theta}_{s_{1,1'}}, S^{\Theta}_{s_{1,2'}}) \\ &= S^{\Theta}_{s_{1,1'}} \\ &\quad \text{mit } \theta_{3'} = \theta_3, \theta_{3''} = \theta_3, \theta_{1,2'} = \theta_2 \\ &= (1-2\theta_3)\mu_1 + 2\theta_3 max(\mu_1,\mu_3) = S^{\Theta^E}_{X_{\wedge,\vee}} \end{aligned}$$

□

Behauptung C.4.12 $S^{\Theta^E}_{X_{\vee,\wedge}} \Rightarrow S^{\Theta^E}_{X'}$

Beweis Für $S^{\Theta^E}_{X_{\vee,\wedge}}$ ergibt sich:

$$\begin{aligned} S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 max(\mu_1,\mu_2) \\ S^{\Theta^E}_{X_{\vee,\wedge}} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 min(S^{\Theta}_{s_{1,1}},\mu_3) \\ &= S^{\Theta}_{s_{1,1}} \\ &= (1-2\theta_2)\mu_1 + 2\theta_2 max(\mu_1,\mu_2) \end{aligned}$$

und für $S^{\Theta^E}_{X'}$:

$$\begin{aligned} S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_1 + 2\theta_{3'} min(\mu_1,\mu_3) = \mu_1 \\ S^{\Theta}_{s_{1,2'}} &= (1-2\theta_{3''})\mu_2 + 2\theta_{3''} min(\mu_2,\mu_3) = \mu_2 \\ S^{\Theta^E}_{X'} &= (1-2\theta_{1,2'})S^{\Theta}_{s_{1,1'}} + 2\theta_{1,2'} max(S^{\Theta}_{s_{1,1'}}, S^{\Theta}_{s_{1,2'}}) \\ &= (1-2\theta_{1,2'})\mu_1 + 2\theta_{1,2'} max(\mu_1,\mu_2) \\ &\quad \text{mit } \theta_{3'} = \theta_3, \theta_{3''} = \theta_3, \theta_{1,2'} = \theta_2 \\ &= (1-2\theta_3)\mu_1 + 2\theta_3 max(\mu_1,\mu_2) = S^{\Theta^E}_{X_{\vee,\wedge}} \end{aligned}$$

□

C.5 Assoziativität

Es gilt:
$((((x_1,\theta_1) \wedge (x_2,\theta_2)),\theta_{1,1}) \wedge (x_3,\theta_3)) = ((x_1,\theta_{1'}) \wedge (((x_2,\theta_{2'}) \wedge (x_3,\theta_{3'})),\theta_{1,1'}))$,
$((((x_1,\theta_1) \vee (x_2,\theta_2)),\theta_{1,1}) \vee (x_3,\theta_3)) = ((x_1,\theta_{1'}) \vee (((x_2,\theta_{2'}) \vee (x_3,\theta_{3'})),\theta_{1,1'}))$.

Wie bei der distributiven Transformation werden in Abhängigkeit von μ_1, μ_2, μ_3 die folgenden sechs Fälle unterschieden:

1. $\mu_1 \geq \mu_2 \geq \mu_3$
2. $\mu_1 \geq \mu_3 \geq \mu_2$
3. $\mu_2 \geq \mu_1 \geq \mu_3$
4. $\mu_2 \geq \mu_3 \geq \mu_1$
5. $\mu_3 \geq \mu_1 \geq \mu_2$
6. $\mu_3 \geq \mu_2 \geq \mu_1$

Des Weiteren wird zwischen einer *und*-Anfrage ($X_\wedge$) und einer *oder*-Anfrage ($X_\vee$) differenziert. Im Folgenden sind die Beweise für die assoziativ Transformation von X nach X' und umgekehrt für jeden Fall angegeben. Da es sich bei diesen Anfragen um komplexe Anfragen handelt, muß die Gewichtung in expliziter Form Θ^E vorliegen.

Für eine übersichtlichere Darstellung werden die Scoring-Funktionen für die einzelnen Teilanfragen getrennt aufgeführt. So wird beispielsweise die Teilanfrage $s_{1,1}$ durch die Scoring-Funktion $S^\Theta_{s_{1,1}}$ evaluiert.

C.5.1 Beweise für Fall 1: $\mu_1 \geq \mu_2 \geq \mu_3$

Behauptung C.5.1 $S^{\Theta^E}_{X_\wedge} \Rightarrow S^{\Theta^E}_{X'_\wedge}$

Beweis Für $S^{\Theta^E}_{X_\wedge}$ ergibt sich:

$$\begin{aligned} S^\Theta_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 min(\mu_1,\mu_2) \\ S^{\Theta^E}_{X_\wedge} &= (1-2\theta_3)S^\Theta_{s_{1,1}} + 2\theta_3 min(S^\Theta_{s_{1,1}},\mu_3) \\ &= (1-2\theta_3)((1-2\theta_2)\mu_1 + 2\theta_2\mu_2) + 2\theta_3\mu_3 \end{aligned}$$

und für $S^{\Theta^E}_{X'_\wedge}$:

$$\begin{aligned}
S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_2 + 2\theta_{3'} min(\mu_2,\mu_3) \\
S^{\Theta^E}_{X'} &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'} min(\mu_1, S^{\Theta}_{s_{1,1'}}) \\
&= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'} S^{\Theta}_{s_{1,1'}} \\
&= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'}(1-2\theta_{3'})\mu_2 + 2\theta_{1,1'}2\theta_{3'}\mu_3 \\
&\quad \text{mit } \theta_{3'} = \frac{\theta_3}{2\theta_{1,1'}}, \theta_{1,1'} = \frac{1-(1-2\theta_2)(1-2\theta_3)}{2} \\
&= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'}\left(1-\frac{2\theta_3}{2\theta_{1,1'}}\right)\mu_2 + 2\theta_{1,1'}\left(\frac{2\theta_3}{2\theta_{1,1'}}\right)\mu_3 \\
&= (1-2\theta_{1,1'})\mu_1 + (2\theta_{1,1'} - 2\theta_3)\,\mu_2 + 2\theta_3\mu_3 \\
&= (1-2\theta_3)(1-2\theta_2)\mu_1 + (1-2\theta_3)(1-(1-2\theta_2)))\mu_2 + 2\theta_3\mu_3 \\
&= (1-2\theta_3)(1-2\theta_2)\mu_1 + (1-2\theta_3)2\theta_2\mu_2 + 2\theta_3\mu_3 \\
&= (1-2\theta_3)((1-2\theta_2)\mu_1 + 2\theta_2\mu_2) + 2\theta_3\mu_3 = S^{\Theta^E}_{X_\wedge}
\end{aligned}$$

□

Behauptung C.5.2 $S^{\Theta^E}_{X_\vee} \Rightarrow S^{\Theta^E}_{X'_\vee}$

Beweis Für $S^{\Theta^E}_{X_\vee}$ ergibt sich:

$$\begin{aligned}
S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 max(\mu_1,\mu_2) = \mu_1 \\
S^{\Theta^E}_{X_\vee} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 max(S^{\Theta}_{s_{1,1}},\mu_3) = S^{\Theta}_{s_{1,1}} = \mu_1
\end{aligned}$$

und für X':

$$\begin{aligned}
S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_2 + 2\theta_{3'} max(\mu_2,\mu_3) = \mu_2 \\
S^{\Theta^E}_{X'} &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'} max(\mu_1, S^{\Theta}_{s_{1,1'}}) \\
&\quad \text{mit } \theta_{3'} = \theta_2\ \theta_{1,1'} = \theta_2 \\
&= (1-2\theta_2)\mu_1 + 2\theta_2 max(\mu_1,\mu_2) = \mu_1 = S^{\Theta^E}_{X_\vee}
\end{aligned}$$

□

Behauptung C.5.3 $S^{\Theta^E}_{X_\wedge} \Leftarrow S^{\Theta^E}_{X'_\wedge}$

Fall 1: $\theta_{3'} \neq \theta_{1,1'} \vee \theta_{3'} = \theta_{1,1'} \neq 0.5$ Für $S^{\Theta^E}_{X'_\wedge}$ ergibt sich:

$$\begin{aligned}
S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_2 + 2\theta_{3'} min(\mu_2, \mu_3) \\
&= (1-2\theta_{3'})\mu_2 + 2\theta_{3'}\mu_3) \\
S^{\Theta^E}_{X'_\wedge} &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'} min(\mu_1, S^{\Theta}_{s_{1,1'}}) \\
&= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'} S^{\Theta}_{s_{1,1'}} \\
&= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'}(1-2\theta_{3'})\mu_2 + 2\theta_{1,1'}2\theta_{3'}
\end{aligned}$$

und für $S^{\Theta^E}_{X_\wedge}$:

$$\begin{aligned}
S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 min(\mu_1, \mu_2) \\
&= (1-2\theta_2)\mu_1 + 2\theta_2\mu_2 \\
S^{\Theta^E}_{X_\wedge} &= (1-2\theta_3) S^{\Theta}_{s_{1,1}} + 2\theta_3 min(S^{\Theta}_{s_{1,1}}, \mu_3) \\
&= (1-2\theta_3)(1-2\theta_2)\mu_1 + (1-2\theta_3)2\theta_2\mu_2 + 2\theta_3\mu_3 \\
&\quad \text{mit } \theta_3 = 2\theta_{1,1'}\theta_{3'}, \theta_2 = \frac{(1-2\theta_{3'})\theta_{1,1'}}{(1-2\theta_3)} \\
&= (1-2\theta_3)\left(1 - 2\frac{(1-2\theta_{3'})\theta_{1,1'}}{(1-2\theta_3)}\right)\mu_1 + (1-2\theta_3)2\frac{(1-2\theta_{3'})\theta_{1,1'}}{(1-2\theta_3)}\mu_2 + 2\theta_3\mu_3 \\
&= (1-2(2\theta_{1,1'}\theta_{3'})) - (2(1-2\theta_{3'})\theta_{1,1'})\mu_1 + 2(1-2\theta_{3'})\theta_{1,1'}\mu_2 + 2(2\theta_{1,1'}\theta_{3'})\mu_3 \\
&= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'}(1-2\theta_{3'})\mu_2 + 2\theta_{1,1'}2\theta_{3'}\mu_3 \\
&= S^{\Theta^E}_{X'_\wedge}
\end{aligned}$$

□

Fall 2: $\theta_{3'} = \theta_{1,1'} \neq 0.5$ Für $S^{\Theta^E}_{X'_\wedge}$ ergibt sich:

$$\begin{aligned}
S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_2 + 2\theta_{3'} min(\mu_2, \mu_3) \\
&= \mu_3 \\
S^{\Theta^E}_{X'_\wedge} &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'} min(\mu_1, S^{\Theta}_{s_{1,1'}}) \\
&= S^{\Theta}_{s_{1,1'}} = \mu_3
\end{aligned}$$

und für $S^{\Theta^E}_{X_\wedge}$:

$$\begin{aligned}
S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 min(\mu_1, \mu_2) \\
S^{\Theta^E}_{X_\wedge} &= (1-2\theta_3) S^{\Theta}_{s_{1,1}} + 2\theta_3 min(S^{\Theta}_{s_{1,1}}, \mu_3) \\
&= (1-2\theta_3)(1-2\theta_2)\mu_1 + (1-2\theta_3)2\theta_2\mu_2 + 2\theta_3\mu_3 \\
&\quad \text{mit } \theta_3 = \theta_{1,1'} = 0.5, \theta_2 = \theta_{1,1'} = 0.5 \\
&= \mu_3 = S^{\Theta^E}_{X'_\wedge}
\end{aligned}$$

□

Behauptung C.5.4 $S^{\Theta^E}_{X_\vee} \Leftarrow S^{\Theta^E}_{X'_\vee}$

Beweis Für $S^{\Theta^E}_{X'_\vee}$ ergibt sich:

$$\begin{aligned} S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_2 + 2\theta_{3'} max(\mu_2,\mu_3) \\ &= \mu_2 \\ S^{\Theta^E}_{X'_\vee} &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'} max(\mu_1, S^{\Theta}_{s_{1,1'}}) \\ &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'} max(\mu_1,\mu_2) \\ &= \mu_1 \end{aligned}$$

und für $S^{\Theta^E}_{X_\vee}$:

$$\begin{aligned} S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 max(\mu_1,\mu_2) \\ &= \mu_1 \\ S^{\Theta^E}_{X_\vee} &= (1-2\theta_3) S^{\Theta}_{s_{1,1}} + 2\theta_3 max(S^{\Theta}_{s_{1,1}}, \mu_3) = S^{\Theta}_{s_{1,1}} \\ &\quad \text{mit } \theta_3 = \theta_{1,1'}, \theta_2 = \theta_{1,1'} \\ &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'} max(\mu_1,\mu_2) \\ &= \mu_1 = S^{\Theta^E}_{X'_\vee} \end{aligned}$$

□

C.5.2 Beweise für Fall 2: $\mu_1 \geq \mu_3 \geq \mu_2$

Behauptung C.5.5 $S^{\Theta^E}_{X_\wedge} \Rightarrow S^{\Theta^E}_{X'_\wedge}$

Fall 1: $S^{\Theta}_{s_{1,1}} > \mu_3$; Für $S^{\Theta^E}_{X_\wedge}$ ergibt sich:

$$\begin{aligned} S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 min(\mu_1,\mu_2) \\ S^{\Theta^E}_{X_\wedge} &= (1-2\theta_3) S^{\Theta}_{s_{1,1}} + 2\theta_3 min(S^{\Theta}_{s_{1,1}}, \mu_3) \\ &= (1-2\theta_3) S^{\Theta}_{s_{1,1}} + 2\theta_3 \mu_3 \\ &= (1-2\theta_3)((1-2\theta_2)\mu_1 + 2\theta_2\mu_2) + 2\theta_3\mu_3 \end{aligned}$$

Für $S^{\Theta^E}_{X'_\wedge}$ ergibt sich mit $\theta_{3'} = 1 - \frac{(1-2\theta_3)\theta_2}{2\theta_{1,1'}} \rightarrow \theta_{2'} = \frac{(1-2\theta_3)\theta_2}{2\theta_{1,1'}}, \theta_{1,1'} = \frac{1-(1-2\theta_3)(1-2\theta_2)}{2}$:

$$
\begin{aligned}
S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{2'})\mu_3 + 2\theta_{2'}\mu_2 \\
S^{\Theta^E}_{X'} &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'}S^{\Theta}_{s_{1,1'}} \\
&= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'}\left(1-2\frac{(1-2\theta_3)\theta_2}{2\theta_{1,1'}}\right)\mu_3 + 2\theta_{1,1'}2\frac{(1-2\theta_3)\theta_2}{2\theta_{1,1'}}\mu_2 \\
&= (1-2\theta_{1,1'})\mu_1 + (2\theta_{1,1'} - 2(1-2\theta_3)\theta_2)\,\mu_3 + (1-2\theta_3)2\theta_2\mu_2 \\
&= (1-2\theta_2)(1-2\theta_3)\mu_1 + (1-(1-2\theta_2)(1-2\theta_3) - 2\theta_2(1-2\theta_3))\mu_3 + (1-2\theta_3)2\theta_2\mu_2 \\
&= (1-2\theta_3)((1-2\theta_2)\mu_1 + 2\theta_2\mu_2) + (1-(1-2\theta_3)(1-2\theta_2+2\theta_2))\mu_3 \\
&= (1-2\theta_3)((1-2\theta_2)\mu_1 + 2\theta_2\mu_2) + 2\theta_3\mu_3 = S^{\Theta^E}_{X_\wedge}
\end{aligned}
$$

□

Fall 2: $S^{\Theta}_{s_{1,1}} \leq \mu_3$; Für $S^{\Theta^E}_{X_\wedge}$ ergibt sich:

$$
\begin{aligned}
S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 min(\mu_1,\mu_2) \\
S^{\Theta^E}_{X_\wedge} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 min(S^{\Theta}_{s_{1,1}},\mu_3) \\
&= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 S^{\Theta}_{s_{1,1}} \\
&= S^{\Theta}_{s_{1,1}} \\
&= (1-2\theta_2)\mu_1 + 2\theta_2\mu_2
\end{aligned}
$$

Für $S^{\Theta^E}_{X'_\wedge}$ ergibt sich mit $\theta_{3'} = \theta_2, \theta_{1,1'} = \theta_2$:

$$
\begin{aligned}
S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_2 + 2\theta_{3'} min(\mu_2,\mu_3) \\
S^{\Theta^E}_{X'} &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'} min(\mu_1, S^{\Theta}_{s_{1,1'}}) \\
&= (1-2\theta_2)\mu_1 + 2\theta_2 min(\mu_1,\mu_2) \\
&= (1-2\theta_2)\mu_1 + 2\theta_2\mu_2 = S^{\Theta^E}_{X_\wedge}
\end{aligned}
$$

□

Behauptung C.5.6 $S^{\Theta^E}_{X_\vee} \Rightarrow S^{\Theta^E}_{X'_\vee}$

Beweis Für $S^{\Theta^E}_{X_\vee}$ ergibt sich:

$$
\begin{aligned}
S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 max(\mu_1,\mu_2) = \mu_1 \\
S^{\Theta^E}_{X_\vee} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 max(S^{\Theta}_{s_{1,1}},\mu_3) = S^{\Theta}_{s_{1,1}} = \mu_1
\end{aligned}
$$

und für $S^{\Theta^E}_{X'_\vee}$:

$$\begin{aligned} S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_2 + 2\theta_{3'} max(\mu_2,\mu_3) \\ S^{\Theta^E}_{X'} &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'} max(\mu_1, S^{\Theta}_{s_{1,1'}}) \\ &\quad \text{mit } \theta_{3'} = \theta_2, \theta_{1,1'} = \theta_2 \\ &= (1-2\theta_2)\mu_1 + 2\theta_2 max(\mu_1,\mu_2) = \mu_1 = S^{\Theta^E}_{X_\vee} \end{aligned}$$

□

Behauptung C.5.7 $S^{\Theta^E}_{X_\wedge} \Leftarrow S^{\Theta^E}_{X'_\wedge}$

Beweis Für $S^{\Theta^E}_{X'_\wedge}$ ergibt sich:

$$\begin{aligned} S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_2 + 2\theta_{3'} min(\mu_2,\mu_3) \\ &= \mu_2 \\ S^{\Theta^E}_{X'_\wedge} &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'} min(S^{\Theta}_{s_{1,1}}, \mu_1) \\ &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'}\mu_2 \end{aligned}$$

Für $S^{\Theta^E}_{X_\wedge}$ ergibt sich:

$$\begin{aligned} S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 min(\mu_1,\mu_2) \\ &= (1-2\theta_2)\mu_1 + 2\theta_2\mu_2 \\ S^{\Theta^E}_{X_\wedge} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 min(S^{\Theta}_{s_{1,1}}, \mu_3) \\ &\quad \text{mit } \theta_3 = 0, \theta_2 = \theta_{1,1'} \\ &= (1-2*0)S^{\Theta}_{s_{1,1}} + 2*0 min(S^{\Theta}_{s_{1,1}}, \mu_3) \\ &= S^{\Theta}_{s_{1,1}} \\ &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'}\mu_2 = S^{\Theta^E}_{X'_\wedge} \end{aligned}$$

□

Behauptung C.5.8 $S^{\Theta^E}_{X_\vee} \Leftarrow S^{\Theta^E}_{X'_\vee}$

Beweis Für $S^{\Theta^E}_{X'_\vee}$ ergibt sich:

$$\begin{aligned} S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_2 + 2\theta_{3'} max(\mu_2,\mu_3) \\ S^{\Theta^E}_{X'_\vee} &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'} max(\mu_1, S^{\Theta}_{s_{1,1'}}) \\ &= \mu_1 \end{aligned}$$

und für $S^{\Theta^E}_{X_\wedge}$ mit $\theta_3 = \theta_{1,1'}, \theta_2 = \theta_{1,1'}$:

$$\begin{aligned} S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 max(\mu_1,\mu_2) = \mu_1 \\ S^{\Theta^E}_{X_\vee} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 max(S^{\Theta}_{s_{1,1}},\mu_3) = S^{\Theta}_{s_{1,1}} \\ &= \mu_1 = S^{\Theta^E}_{X'_\vee} \end{aligned}$$

□

C.5.3 Beweise für Fall 3: $\mu_2 \geq \mu_1 \geq \mu_3$

Behauptung C.5.9 $S^{\Theta^E}_{X_\wedge} \Rightarrow S^{\Theta^E}_{X'_\wedge}$

Beweis Für $S^{\Theta^E}_{X_\wedge}$ ergibt sich:

$$\begin{aligned} S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 min(\mu_1,\mu_2) = \mu_1 \\ S^{\Theta^E}_{X_\wedge} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 min(S^{\Theta}_{s_{1,1}},\mu_3) \\ &= (1-2\theta_3)\mu_1 + 2\theta_3\mu_3 \end{aligned}$$

und für $S^{\Theta^E}_{X'_\wedge}$:

$$\begin{aligned} S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_2 + 2\theta_{3'} min(\mu_2,\mu_3) \\ S^{\Theta^E}_{X'} &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'} min(\mu_1, S^{\Theta}_{s_{1,1'}}) \\ &\quad \text{mit } \theta_{3'} = 1 \rightarrow \theta_{3'} > \theta_{2'}, \theta_{1,1'} = \theta_3 \\ S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{2'})\mu_3 + 2\theta_{2'} min(\mu_2,\mu_3) = \mu_3 \\ S^{\Theta^E}_{X'} &= (1-2\theta_3)\mu_1 + 2\theta_3 min(\mu_1,\mu_3) \\ &= (1-2\theta_3)\mu_1 + 2\theta_3\mu_3 = S^{\Theta^E}_{X_\wedge} \end{aligned}$$

□

Behauptung C.5.10 $S^{\Theta^E}_{X_\vee} \Rightarrow S^{\Theta^E}_{X'_\vee}$

Beweis Für $S^{\Theta^E}_{X_\vee}$ ergibt sich:

$$\begin{aligned} S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 max(\mu_1,\mu_2) \\ S^{\Theta^E}_{X_\vee} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 max(S^{\Theta}_{s_{1,1}},\mu_3) = S^{\Theta}_{s_{1,1}} \\ &= (1-2\theta_2)\mu_1 + 2\theta_2 max(\mu_1,\mu_2) \end{aligned}$$

und für $S^{\Theta^E}_{X'_\vee}$:

$$\begin{aligned} S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_2 + 2\theta_{3'} max(\mu_2,\mu_3) = \mu_2 \\ S^{\Theta^E}_{X'} &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'} max(\mu_1, S^{\Theta}_{s_{1,1'}}) \\ &\quad \text{mit } \theta_{3'} = \theta_2, \theta_{1,1'} = \theta_2 \\ &= (1-2\theta_2)\mu_1 + 2\theta_2 max(\mu_1,\mu_2) = S^{\Theta^E}_{X_\vee} \end{aligned}$$

□

Behauptung C.5.11 $S^{\Theta^E}_{X_\wedge} \Rightarrow S^{\Theta^E}_{X'_\wedge}$

Fall 1: $S^{\Theta}_{s_{1,1'}} \geq \mu_1$ Für $S^{\Theta^E}_{X'_\wedge}$ ergibt sich:

$$\begin{aligned} S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_2 + 2\theta_{3'} min(\mu_2,\mu_3) \\ S^{\Theta^E}_{X'_\wedge} &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'} min(S^{\Theta}_{s_{1,1}}, \mu_1) \\ &= \mu_1 \end{aligned}$$

Für $S^{\Theta^E}_{X_\wedge}$ ergibt sich:

$$\begin{aligned} S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 min(\mu_1,\mu_2) \\ &= \mu_1 \\ S^{\Theta^E}_{X_\wedge} &= (1-2\theta_3) S^{\Theta}_{s_{1,1}} + 2\theta_3 min(S^{\Theta}_{s_{1,1}}, \mu_3) \\ &= (1-2\theta_3)\mu_1 + 2\theta_3\mu_3 \\ &\quad \text{mit } \theta_3 = 0, \theta_2 = \theta_{1,1'} \\ &= (1-2*0)\mu_1 + 2*0\mu_3 \\ &= \mu_1 = S^{\Theta^E}_{X'_\wedge} \end{aligned}$$

□

Fall 2.1: $S^{\Theta}_{s_{1,1'}} < \mu_1 \wedge (\theta_{3'} \neq \theta_{1,1'} \vee \theta_{3'} = \theta_{1,1'} \neq 0.5)$ Für $S^{\Theta^E}_{X'_\wedge}$ ergibt sich:

$$\begin{aligned} S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_2 + 2\theta_{3'} min(\mu_2,\mu_3) \\ S^{\Theta^E}_{X'_\wedge} &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'} min(S^{\Theta}_{s_{1,1}}, \mu_1) \\ &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'}(1-2\theta_{3'})\mu_2 + 2\theta_{1,1'}2\theta_{3'}\mu_3 \end{aligned}$$

Für $S^{\Theta^E}_{X_\wedge}$ ergibt sich:

$$\begin{aligned}
S^{\Theta}_{s_{1,1}} &= (1-2\theta_1)\mu_2 + 2\theta_1 min(\mu_1,\mu_2)\\
&= (1-2\theta_1)\mu_2 + 2\theta_1\mu_1\\
S^{\Theta^E}_{X_\wedge} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 min(S^{\Theta}_{s_{1,1}},\mu_3)\\
&= (1-2\theta_3)(1-2\theta_1)\mu_2 + (1-2\theta_3)2\theta_1\mu_1 + 2\theta_3\mu_3\\
&\quad \text{mit } \theta_3 = 2\theta_{1,1'}\theta_{3'}, \theta_2 = 1 - \frac{(1-2\theta_{1,1'})}{2(1-2\theta_3)} \rightarrow \theta_1 = \frac{(1-2\theta_{1,1'})}{2(1-2\theta_3)}\\
&= (1-2\theta_3)\left(1 - 2\frac{(1-2\theta_{1,1'})}{2(1-2\theta_3)}\right)\mu_2 + (1-2\theta_3)2\frac{(1-2\theta_{1,1'})}{2(1-2\theta_3)}\mu_1 + 2\theta_3\mu_3\\
&= (1-2\theta_3) - (1-2\theta_{1,1'})\mu_2 + (1-2\theta_{1,1})\mu_1 + 2\theta_3\mu_3\\
&= (1-2\theta_{1,1})\mu_1 + (1-2(2\theta_{1,1'}\theta_{3'})) - (1-2\theta_{1,1'})\mu_2 + 2\theta_{1,1'}2\theta_{3'}\mu_3\\
&= (1-2\theta_{1,1})\mu_1 + 2\theta_{1,1'}(1-2\theta_{3'})\mu_2 + 2\theta_{1,1'}2\theta_{3'}\mu_3 = S^{\Theta^E}_{X'_\wedge}
\end{aligned}$$

□

Fall 2.2: $S^{\Theta}_{s_{1,1'}} < \mu_1 \wedge \theta_{1,1'} = \theta_{3'} = 0.5$) Für $S^{\Theta^E}_{X'_\wedge}$ ergibt sich:

$$\begin{aligned}
S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_2 + 2\theta_{3'} min(\mu_2,\mu_3)\\
&= \mu_3\\
S^{\Theta^E}_{X'_\wedge} &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'} min(S^{\Theta}_{s_{1,1}},\mu_1)\\
&= \mu_3
\end{aligned}$$

Für $S^{\Theta^E}_{X_\wedge}$ ergibt sich:

$$\begin{aligned}
S^{\Theta}_{s_{1,1}} &= (1-2\theta_1)\mu_2 + 2\theta_1 min(\mu_1,\mu_2)\\
&= \mu_1\\
S^{\Theta^E}_{X_\wedge} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 min(S^{\Theta}_{s_{1,1}},\mu_3)\\
&= (1-2\theta_3)\mu_1 + 2\theta_3\mu_3)\\
&\quad \text{mit } \theta_3 = \theta_{1,1'} = 0.5, \theta_2 = \theta_{1,1'} = 0.5\\
&= \mu_3 = S^{\Theta^E}_{X'_\wedge}
\end{aligned}$$

□

Behauptung C.5.12 $S^{\Theta^E}_{X_\vee} \Leftarrow S^{\Theta^E}_{X'_\vee}$

Beweis Für $S^{\Theta^E}_{X'_\vee}$ ergibt sich:

$$\begin{aligned}
S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_2 + 2\theta_{3'} max(\mu_2,\mu_3)\\
&= \mu_2\\
S^{\Theta^E}_{X'_\vee} &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'} max(\mu_1, S^{\Theta}_{s_{1,1'}})\\
&= \mu_(1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'}\mu_2
\end{aligned}$$

und für $S^{\Theta^E}_{X_\vee}$:

$$\begin{aligned} S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 max(\mu_1,\mu_2) \\ S^{\Theta^E}_{X_\vee} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 max(S^{\Theta}_{s_{1,1}},\mu_3) = S^{\Theta}_{s_{1,1}} \\ &\quad \text{mit } \theta_3 = \theta_{1,1'}, \theta_2 = \theta_{1,1'} \\ &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'}\mu_2 = S^{\Theta^E}_{X'_\vee} \end{aligned}$$

□

C.5.4 Beweise für Fall 4: $\mu_2 \geq \mu_3 \geq \mu_1$

Behauptung C.5.13 $S^{\Theta^E}_{X_\wedge} \Rightarrow S^{\Theta^E}_{X'_\wedge}$

Beweis Für $S^{\Theta^E}_{X_\wedge}$ ergibt sich:

$$\begin{aligned} S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 min(\mu_1,\mu_2) = \mu_1 \\ S^{\Theta^E}_{X_\wedge} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 min(S^{\Theta}_{s_{1,1}},\mu_3) \\ &= (1-2\theta_3)\mu_1 + 2\theta_3\mu_1 = \mu_1 \end{aligned}$$

und für $S^{\Theta^E}_{X'_\wedge}$:

$$\begin{aligned} S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_2 + 2\theta_{3'} min(\mu_2,\mu_3) \\ S^{\Theta^E}_{X'} &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'} min(\mu_1, S^{\Theta}_{s_{1,1'}}) \\ &\quad \text{mit } \theta_{3'} = \theta_2, \theta_{1,1'} = \theta_2 \\ S^{\Theta^E}_{X'} &= (1-2\theta_2)\mu_1 + 2\theta_2 min(\mu_1,\mu_3) = \mu_1 = S^{\Theta^E}_{X_\wedge} \end{aligned}$$

□

Behauptung C.5.14 $S^{\Theta^E}_{X_\vee} \Rightarrow S^{\Theta^E}_{X'_\vee}$

Fall 1: $S^{\Theta}_{s_{1,1}} \geq \mu_3$ Für $S^{\Theta^E}_{X_\vee}$ ergibt sich:

$$\begin{aligned} S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 max(\mu_1,\mu_2) \\ S^{\Theta^E}_{X_\vee} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 max(S^{\Theta}_{s_{1,1}},\mu_3) \\ &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 S^{\Theta}_{s_{1,1}} \\ &= (1-2\theta_2)\mu_1 + 2\theta_2 max(\mu_1,\mu_2) \end{aligned}$$

Für $S^{\Theta^E}_{X'_\vee}$ ergibt sich mit mit $\theta_{3'} = \theta_2, \theta_{1,1'} = \theta_2$:

$$\begin{aligned} S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_2 + 2\theta_{3'} max(\mu_2, \mu_3) = \mu_2 \\ S^{\Theta^E}_{X'} &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'} max(\mu_1, S^{\Theta}_{s_{1,1'}}) \\ &= (1-2\theta_2)\mu_1 + 2\theta_2 max(\mu_1, \mu_2) = S^{\Theta^E}_{X_\vee} \end{aligned}$$

□

Fall 2: $S^{\Theta}_{s_{1,1}} < \mu_3$ Für $S^{\Theta^E}_{X_\vee}$ ergibt sich:

$$\begin{aligned} S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 max(\mu_1, \mu_2) \\ S^{\Theta^E}_{X_\vee} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 max(S^{\Theta}_{s_{1,1}}, \mu_3) \\ &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3\mu_3 \\ &= (1-2\theta_3)((1-2\theta_2)\mu_1 + 2\theta_2\mu_2) + 2\theta_3\mu_3 \end{aligned}$$

Für $S^{\Theta^E}_{X'_\vee}$ ergibt sich mit $\theta_{3'} = 1 - \frac{(1-2\theta_3)\theta_2}{2\theta_{1,1'}} \rightarrow \theta_{2'} = \frac{(1-2\theta_3)\theta_2}{2\theta_{1,1'}}, \theta_{1,1'} = \frac{1-(1-2\theta_3)(1-2\theta_2)}{2}$:

$$\begin{aligned} S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{2'})\mu_3 + 2\theta_{2'}\mu_2 \\ S^{\Theta^E}_{X'} &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'} max(\mu_1, S^{\Theta}_{s_{1,1'}}) \\ &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'} S^{\Theta}_{s_{1,1'}} \\ &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'}\left(1 - 2\frac{(1-2\theta_3)\theta_2}{2\theta_{1,1'}}\right)\mu_3 + 2\theta_{1,1'}2\frac{(1-2\theta_3)\theta_2}{2\theta_{1,1'}}\mu_2 \\ &= (1-2\theta_{1,1'})\mu_1 + (2\theta_{1,1'} - 2(1-2\theta_3)\theta_2)\,\mu_3 + (1-2\theta_3)2\theta_2\mu_2 \\ &= (1-2\theta_2)(1-2\theta_3)\mu_1 + (1-(1-2\theta_2)(1-2\theta_3) - 2\theta_2(1-2\theta_3))\mu_3 + (1-2\theta_3)2\theta_2\mu_2 \\ &= (1-2\theta_3)((1-2\theta_2)\mu_1 + 2\theta_2\mu_2) + (1-(1-2\theta_3)(1-2\theta_2+2\theta_2))\mu_3 \\ &= (1-2\theta_3)((1-2\theta_2)\mu_1 + 2\theta_2\mu_2) + 2\theta_3\mu_3 = S^{\Theta^E}_{X_\wedge} \end{aligned}$$

□

Behauptung C.5.15 $S^{\Theta^E}_{X_\wedge} \Leftarrow S^{\Theta^E}_{X'_\wedge}$

Beweis Für $S^{\Theta^E}_{X'_\wedge}$ ergibt sich:

$$\begin{aligned} S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_2 + 2\theta_{3'} min(\mu_2, \mu_3) \\ S^{\Theta^E}_{X'_\wedge} &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'} min(\mu_1, S^{\Theta}_{s_{1,1'}}) \\ &= \mu_1 \end{aligned}$$

und für $S^{\Theta^E}_{X_\wedge}$ mit $\theta_{3'} = \theta_2, \theta_{1,1'} = \theta_2$:

$$\begin{aligned} S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 min(\mu_1,\mu_2) \\ &= \mu_1 \\ S^{\Theta^E}_{X_\wedge} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 min(S^{\Theta}_{s_{1,1}},\mu_3) \\ &= S^{\Theta}_{s_{1,1}} \\ &= \mu_1 = S^{\Theta^E}_{X'_\wedge} \end{aligned}$$

□

Behauptung C.5.16 $S^{\Theta^E}_{X_\vee} \Leftarrow S^{\Theta^E}_{X'_\vee}$

Beweis Für $S^{\Theta^E}_{X'_\vee}$ ergibt sich:

$$\begin{aligned} S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_2 + 2\theta_{3'} max(\mu_2,\mu_3) \\ &= \mu_2 \\ S^{\Theta^E}_{X'_\wedge} &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'} max(S^{\Theta}_{s_{1,1}},\mu_1) \\ &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'}\mu_2 \end{aligned}$$

Für $S^{\Theta^E}_{X_\wedge}$ ergibt sich:

$$\begin{aligned} S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 max(\mu_1,\mu_2) \\ &= (1-2\theta_2)\mu_1 + 2\theta_2\mu_2 \\ S^{\Theta^E}_{X_\vee} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 max(S^{\Theta}_{s_{1,1}},\mu_3) \\ &\quad \text{mit } \theta_3 = 0, \theta_2 = \theta_{1,1'} \\ &= (1-2*0)S^{\Theta}_{s_{1,1}} + 2*0*max(S^{\Theta}_{s_{1,1}},\mu_3) \\ &= S^{\Theta}_{s_{1,1}} \\ &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'}\mu_2 = S^{\Theta^E}_{X'_\vee} \end{aligned}$$

□

C.5.5 Beweise für Fall 5: $\mu_3 \geq \mu_1 \geq \mu_2$

Behauptung C.5.17 $S^{\Theta^E}_{X_\wedge} \Rightarrow S^{\Theta^E}_{X'_\wedge}$

Beweis Für $S^{\Theta^E}_{X_\wedge}$ ergibt sich:

$$\begin{aligned} S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 min(\mu_1,\mu_2) \\ S^{\Theta^E}_{X_\wedge} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 min(S^{\Theta}_{s_{1,1}},\mu_3) \\ &= S^{\Theta}_{s_{1,1}} \\ &= (1-2\theta_2)\mu_1 + 2\theta_2 min(\mu_1,\mu_2) \end{aligned}$$

und für $S^{\Theta^E}_{X'_\wedge}$:

$$\begin{aligned} S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_2 + 2\theta_{3'} min(\mu_2,\mu_3) = \mu_2 \\ S^{\Theta^E}_{X'} &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'} min(\mu_1, S^{\Theta}_{s_{1,1'}}) \\ &\quad \text{mit } \theta_{3'} = \theta_2,\ \theta_{1,1'} = \theta_2 \\ S^{\Theta^E}_{X'} &= (1-2\theta_2)\mu_1 + 2\theta_2 min(\mu_1,\mu_2) = S^{\Theta^E}_{X_\wedge} \end{aligned}$$

□

Behauptung C.5.18 $S^{\Theta^E}_{X_\vee} \Rightarrow S^{\Theta^E}_{X'_\vee}$

Beweis Für $S^{\Theta^E}_{X_\vee}$ ergibt sich:

$$\begin{aligned} S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 max(\mu_1,\mu_2) = \mu_1 \\ S^{\Theta^E}_{X_\vee} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 max(S^{\Theta}_{s_{1,1}},\mu_3) \\ &= (1-2\theta_3)\mu_1 + 2\theta_3\mu_3 \end{aligned}$$

und für $S^{\Theta^E}_{X'_\vee}$:

$$\begin{aligned} S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_2 + 2\theta_{3'} max(\mu_2,\mu_3) \\ S^{\Theta^E}_{X'} &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'} max(\mu_1, S^{\Theta}_{s_{1,1'}}) \\ &\quad \text{mit } \theta_{3'} = 1 \rightarrow \theta_{3'} > \theta_{2'},\theta_{1,1'} = \theta_3 \\ S^{\Theta}_{s_{1,1}} &= (1-2\theta_{2'})\mu_3 + 2\theta_{2'} max(\mu_2,\mu_3) = \mu_3 \\ S^{\Theta^E}_{X'} &= (1-2\theta_3)\mu_1 + 2\theta_3 max(\mu_1,\mu_3) = S^{\Theta^E}_{X_\vee} \end{aligned}$$

□

Behauptung C.5.19 $S^{\Theta^E}_{X_\wedge} \Leftarrow S^{\Theta^E}_{X'_\wedge}$

Beweis Für $S^{\Theta^E}_{X'_\wedge}$ ergibt sich:

$$\begin{aligned}
S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_2 + 2\theta_{3'} min(\mu_2, \mu_3) \\
&= \mu_2 \\
S^{\Theta^E}_{X'_\wedge} &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'} min(S^{\Theta}_{s_{1,1}}, \mu_1) \\
&= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'}\mu_2
\end{aligned}$$

Für $S^{\Theta^E}_{X_\wedge}$ ergibt sich:

$$\begin{aligned}
S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 min(\mu_1, \mu_2) \\
&= (1-2\theta_2)\mu_1 + 2\theta_2\mu_2 \\
S^{\Theta^E}_{X_\wedge} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 min(S^{\Theta}_{s_{1,1}}, \mu_3) \\
&= S^{\Theta}_{s_{1,1}} \\
&\quad \text{mit } \theta_3 = \theta_{1,1'}, \theta_2 = \theta_{1,1'} \\
&= 1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'}\mu_2 = S^{\Theta^E}_{X'_\wedge}
\end{aligned}$$

□

Behauptung C.5.20 $S^{\Theta^E}_{X_\vee} \Leftarrow S^{\Theta^E}_{X'_\vee}$

Fall 1: $S^{\Theta}_{s_{1,1'}} \leq \mu_1$ Für $S^{\Theta^E}_{X'_\vee}$ ergibt sich:

$$\begin{aligned}
S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_2 + 2\theta_{3'} max(\mu_2, \mu_3) \\
S^{\Theta^E}_{X'_\vee} &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'} max(S^{\Theta}_{s_{1,1}}, \mu_1) \\
&= \mu_1
\end{aligned}$$

Für $S^{\Theta^E}_{X_\vee}$ ergibt sich:

$$\begin{aligned}
S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 max(\mu_1, \mu_2) \\
&= \mu_1 \\
S^{\Theta^E}_{X_\vee} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 max(S^{\Theta}_{s_{1,1}}, \mu_3) \\
&= (1-2\theta_3)\mu_1 + 2\theta_3\mu_3 \\
&\quad \text{mit } \theta_3 = 0, \theta_2 = \theta_{1,1'} \\
&= (1-2*0)\mu_1 + 2*0\mu_3 \\
&= \mu_1 = S^{\Theta^E}_{X'_\wedge}
\end{aligned}$$

□

Fall 2.1: $S^{\Theta}_{s_{1,1'}} > \mu_1 \wedge (\theta_{3'} \neq \theta_{1,1'} \vee \theta_{3'} = \theta_{1,1'} \neq 0.5)$ Für $S^{\Theta^E}_{X'_\vee}$ ergibt sich:

$$\begin{aligned} S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_2 + 2\theta_{3'} max(\mu_2, \mu_3) \\ S^{\Theta^E}_{X'_\vee} &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'} max(S^{\Theta}_{s_{1,1}}, \mu_1) \\ &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'}(1-2\theta_{3'})\mu_2 + 2\theta_{1,1'}2\theta_{3'}\mu_3 \end{aligned}$$

Für $S^{\Theta^E}_{X_\vee}$ ergibt sich:

$$\begin{aligned} S^{\Theta}_{s_{1,1}} &= (1-2\theta_1)\mu_2 + 2\theta_1 max(\mu_1, \mu_2) \\ &= (1-2\theta_1)\mu_2 + 2\theta_1\mu_1 \\ S^{\Theta^E}_{X_\vee} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 max(S^{\Theta}_{s_{1,1}}, \mu_3) \\ &= (1-2\theta_3)(1-2\theta_1)\mu_2 + (1-2\theta_3)2\theta_1\mu_1 + 2\theta_3\mu_3 \\ &\quad \text{mit } \theta_3 = 2\theta_{1,1'}\theta_{3'}, \theta_2 = 1 - \frac{(1-2\theta_{1,1'})}{2(1-2\theta_3)} \rightarrow \theta_1 = \frac{(1-2\theta_{1,1'})}{2(1-2\theta_3)} \\ &= (1-2\theta_3)\left(1 - 2\frac{(1-2\theta_{1,1'})}{2(1-2\theta_3)}\right)\mu_2 + (1-2\theta_3)2\frac{(1-2\theta_{1,1'})}{2(1-2\theta_3)}\mu_1 + 2\theta_3\mu_3 \\ &= (1-2\theta_3) - (1-2\theta_{1,1'})\mu_2 + (1-2\theta_{1,1})\mu_1 + 2\theta_3\mu_3 \\ &= (1-2\theta_{1,1})\mu_1 + (1-2(2\theta_{1,1'}\theta_{3'})) - (1-2\theta_{1,1'})\mu_2 + 2\theta_{1,1'}2\theta_{3'}\mu_3 \\ &= (1-2\theta_{1,1})\mu_1 + 2\theta_{1,1'}(1-2\theta_{3'})\mu_2 + 2\theta_{1,1'}2\theta_{3'}\mu_3 = S^{\Theta^E}_{X'_\vee} \end{aligned}$$

□

Fall 2.2: $S^{\Theta}_{s_{1,1'}} > \mu_1 \wedge \theta_{1,1'} = \theta_{3'} = 0.5)$ Für $S^{\Theta^E}_{X'_\vee}$ ergibt sich:

$$\begin{aligned} S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_2 + 2\theta_{3'} max(\mu_2, \mu_3) \\ &= \mu_3 \\ S^{\Theta^E}_{X'_\vee} &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'} max(S^{\Theta}_{s_{1,1}}, \mu_1) \\ &= \mu_3 \end{aligned}$$

Für $S^{\Theta^E}_{X_\vee}$ ergibt sich:

$$\begin{aligned} S^{\Theta}_{s_{1,1}} &= (1-2\theta_1)\mu_2 + 2\theta_1 max(\mu_1, \mu_2) \\ &= \mu_1 \\ S^{\Theta^E}_{X_\vee} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 max(S^{\Theta}_{s_{1,1}}, \mu_3) \\ &= (1-2\theta_3)\mu_1 + 2\theta_3\mu_3) \\ &\quad \text{mit } \theta_3 = \theta_{1,1'} = 0.5, \theta_2 = \theta_{1,1'} = 0.5 \\ &= \mu_3 = S^{\Theta^E}_{X'_\vee} \end{aligned}$$

□

C.5.6 Beweise für Fall 6: $\mu_3 \geq \mu_2 \geq \mu_1$

Behauptung C.5.21 $S^{\Theta^E}_{X_\wedge} \Rightarrow S^{\Theta^E}_{X'_\wedge}$

Beweis Für $S^{\Theta^E}_{X_\wedge}$ ergibt sich:

$$\begin{aligned} S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 min(\mu_1,\mu_2) \\ S^{\Theta^E}_{X_\wedge} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 min(S^{\Theta}_{s_{1,1}},\mu_3) \\ &= S^{\Theta}_{s_{1,1}} \\ &= (1-2\theta_2)\mu_1 + 2\theta_2 min(\mu_1,\mu_2) \end{aligned}$$

und für $S^{\Theta^E}_{X'_\wedge}$:

$$\begin{aligned} S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_2 + 2\theta_{3'} min(\mu_2,\mu_3) = \mu_2 \\ S^{\Theta^E}_{X'} &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'} min(\mu_1, S^{\Theta}_{s_{1,1'}}) \\ &\quad \text{mit } \theta_{3'} = \theta_2, \theta_{1,1'} = \theta_2 \\ S^{\Theta^E}_{X'} &= (1-2\theta_2)\mu_1 + 2\theta_2 min(\mu_1,\mu_2) = S^{\Theta^E}_{X_\wedge} \end{aligned}$$

□

Behauptung C.5.22 $S^{\Theta^E}_{X_\vee} \Rightarrow S^{\Theta^E}_{X'_\vee}$

Beweis Für $S^{\Theta^E}_{X_\vee}$ ergibt sich:

$$\begin{aligned} S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 max(\mu_1,\mu_2) \\ S^{\Theta^E}_{X_\vee} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 max(S^{\Theta}_{s_{1,1}},\mu_3) \\ &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3\mu_3 \\ &= (1-2\theta_3)((1-2\theta_2)\mu_1 + 2\theta_2) + 2\theta_3\mu_3 \end{aligned}$$

und für $S^{\Theta^E}_{X'_\vee}$:

$$\begin{aligned}
S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_2 + 2\theta_{3'} max(\mu_2,\mu_3) \\
&= (1-2\theta_{3'})\mu_2 + 2\theta_{3'}\mu_3 \\
S^{\Theta^E}_{X'} &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'} max(\mu_1, S^{\Theta}_{s_{1,1'}}) \\
&= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'}(1-2\theta_{3'})\mu_2 + 2\theta_{1,1'}2\theta_{3'}\mu_3 \\
&\quad \text{mit } \theta_{3'} = \frac{\theta_3}{2\theta_{1,1'}}, \theta_{1,1'} = \frac{1-(1-2\theta_2)(1-2\theta_3)}{2} \\
&= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'}\left(1-\frac{2\theta_3}{2\theta_{1,1'}}\right)\mu_2 + 2\theta_{1,1'}\left(\frac{2\theta_3}{2\theta_{1,1'}}\right)\mu_3 \\
&= (1-2\theta_{1,1'})\mu_1 + (2\theta_{1,1'}-2\theta_3)\,\mu_2 + 2\theta_3\mu_3 \\
&= (1-2\theta_3)(1-2\theta_2)\mu_1 + (1-2\theta_3)(1-(1-2\theta_2)))\mu_2 + 2\theta_3\mu_3 \\
&= (1-2\theta_3)(1-2\theta_2)\mu_1 + (1-2\theta_3)2\theta_2\mu_2 + 2\theta_3\mu_3 \\
&= (1-2\theta_3)((1-2\theta_2)\mu_1 + 2\theta_2\mu_2) + 2\theta_3\mu_3 = S^{\Theta^E}_{X_\vee}
\end{aligned}$$

□

Behauptung C.5.23 $S^{\Theta^E}_{X_\wedge} \Leftarrow S^{\Theta^E}_{X'_\wedge}$

Beweis Für $S^{\Theta^E}_{X'_\wedge}$ ergibt sich:

$$\begin{aligned}
S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_2 + 2\theta_{3'} min(\mu_2,\mu_3) \\
&= \mu_2 \\
S^{\Theta^E}_{X'_\wedge} &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'} min(\mu_1, S^{\Theta}_{s_{1,1'}}) \\
&= \mu_1
\end{aligned}$$

und für $S^{\Theta^E}_{X_\wedge}$ mit $\theta_{3'} = \theta_2, \theta_{1,1'} = \theta_2$:

$$\begin{aligned}
S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 min(\mu_1,\mu_2) \\
&= \mu_1 \\
S^{\Theta^E}_{X_\wedge} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 min(S^{\Theta}_{s_{1,1}}, \mu_3) \\
&= S^{\Theta}_{s_{1,1}} \\
&= \mu_1 = S^{\Theta^E}_{X'_\wedge}
\end{aligned}$$

□

Behauptung C.5.24 $S^{\Theta^E}_{X_\vee} \Leftarrow S^{\Theta^E}_{X'_\vee}$

Fall 1: $\theta_{3'} \neq \theta_{1,1'} \vee \theta_{3'} = \theta_{1,1'} \neq 0.5$ Für $S^{\Theta^E}_{X'_\vee}$ ergibt sich:

$$\begin{aligned}
*S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_2 + 2\theta_{3'} max(\mu_2,\mu_3) \\
&= (1-2\theta_{3'})\mu_2 + 2\theta_{3'}\mu_3) \\
S^{\Theta^E}_{X'_\vee} &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'} max(\mu_1, S^{\Theta}_{s_{1,1'}}) \\
&= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'} S^{\Theta}_{s_{1,1'}} \\
&= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'}(1-2\theta_{3'})\mu_2 + 2\theta_{1,1'}2\theta_{3'}
\end{aligned}$$

und für $S^{\Theta^E}_{X_\wedge}$:

$$\begin{aligned}
S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 max(\mu_1,\mu_2) \\
&= (1-2\theta_2)\mu_1 + 2\theta_2\mu_2 \\
S^{\Theta^E}_{X_\vee} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 max(S^{\Theta}_{s_{1,1}}, \mu_3) \\
&= (1-2\theta_3)(1-2\theta_2)\mu_1 + (1-2\theta_3)2\theta_2\mu_2 + 2\theta_3\mu_3 \\
&\quad \text{mit } \theta_3 = 2\theta_{1,1'}\theta_{3'}, \theta_2 = \frac{(1-2\theta_{3'})\theta_{1,1'}}{(1-2\theta_3)} \\
&= (1-2\theta_3)\left(1 - 2\frac{(1-2\theta_{3'})\theta_{1,1'}}{(1-2\theta_3)}\right)\mu_1 + (1-2\theta_3)2\frac{(1-2\theta_{3'})\theta_{1,1'}}{(1-2\theta_3)}\mu_2 + 2\theta_3\mu_3 \\
&= (1-2(2\theta_{1,1'}\theta_{3'})) - (2(1-2\theta_{3'})\theta_{1,1'})\mu_1 + 2(1-2\theta_{3'})\theta_{1,1'}\mu_2 + 2(2\theta_{1,1'}\theta_{3'})\mu_3 \\
&= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'}(1-2\theta_{3'})\mu_2 + 2\theta_{1,1'}2\theta_{3'}\mu_3 \\
&= S^{\Theta^E}_{X'_\vee}
\end{aligned}$$

□

Fall 2: $\theta_{3'} = \theta_{1,1'} \neq 0.5$ Für $S^{\Theta^E}_{X'_\vee}$ ergibt sich:

$$\begin{aligned}
S^{\Theta}_{s_{1,1'}} &= (1-2\theta_{3'})\mu_2 + 2\theta_{3'} max(\mu_2,\mu_3) \\
&= \mu_3 \\
S^{\Theta^E}_{X'_\vee} &= (1-2\theta_{1,1'})\mu_1 + 2\theta_{1,1'} max(\mu_1, S^{\Theta}_{s_{1,1'}}) \\
&= S^{\Theta}_{s_{1,1'}} = \mu_3
\end{aligned}$$

und für $S^{\Theta^E}_{X_\wedge}$:

$$\begin{aligned}
S^{\Theta}_{s_{1,1}} &= (1-2\theta_2)\mu_1 + 2\theta_2 max(\mu_1,\mu_2) \\
S^{\Theta^E}_{X_\vee} &= (1-2\theta_3)S^{\Theta}_{s_{1,1}} + 2\theta_3 max(S^{\Theta}_{s_{1,1}}, \mu_3) \\
&= (1-2\theta_3)(1-2\theta_2)\mu_1 + (1-2\theta_3)2\theta_2\mu_2 + 2\theta_3\mu_3 \\
&\quad \text{mit } \theta_3 = \theta_{1,1'} = 0.5, \theta_2 = \theta_{1,1'} = 0.5 \\
&= \mu_3 = S^{\Theta^E}_{X'_\vee}
\end{aligned}$$

□