Erschienen in: Dittrich, K. R.; Geppert, A., (Hrsg.), Datenbanksysteme in Büro, Technik und Wissenschaft, BTW'97, GI-Fachtagung, Ulm, März 1997, Seiten 352-369, Informatik aktuell, Springer-Verlag, Berlin, 1997.

Behandlung von Integritätsbedingungen bei Schemarestrukturierung und Schemaintegration

Stefan Conrad Ingo Schmitt Can Türker

Institut für Technische Informationssysteme
Otto-von-Guericke-Universität Magdeburg
Postfach 4120, D–39016 Magdeburg, Germany

E-mail: {conrad|schmitt|tuerker}@iti.cs.uni-magdeburg.de

Zusammenfassung Die Behandlung von Integritätsbedingungen bei Schematransformation, Schemaintegration, Restrukturierung von Schemata und Sichtenbildung ist ein wichtiger Bereich, der bislang nur exemplarisch behandelt wurde. In diesem Beitrag zeigen wir für die wichtigsten Operationen der Schemazerlegung und –zusammensetzung wie begleitend Integritätsbedingungen zu behandeln sind. Dazu stellen wir einige elementare Regeln auf und diskutieren, für welche Arten von Integritätsbedingungen welche Regeln angewendet werden können. Anhand eines einfachen Beispiels demonstrieren wir die Anwendung dieser Regeln.

Schlüsselworte: Integritätsbedingungen, Schemaintegration, Schemarestrukturierung, Schematransformation, Sichtenbildung.

1 Einleitung

In vielen Bereichen werden Datenbankschemata manipuliert. Hierbei handelt es sich zum Beispiel um Ableitung von Sichten aus einem gegebenen Schema, um Schemarestrukturierung etwa zu Optimierungszwecken und um Schematransformation und Schemaintegration, wie sie etwa im Zusammenhang mit föderierten Datenbanken auftreten. Auf den ersten Blick erscheinen diese Bereiche sehr unterschiedlich. Auch wenn zum Beispiel die Schemarestrukturierung als ein Spezialfall der Sichtableitung aufgefaßt werden kann, werden in diesen Bereichen doch zumeist unterschiedliche Ziele verfolgt.

Allen Bereichen gemeinsam sind aber einige zentrale Operationen. Dabei handelt es sich um Vereinigung, Schnitt und Differenz von Mengen, bzw. hier von Objektmengen (Klassen/Extensionen). Hinzu kommt in der Regel noch die Selektionsoperation zur Bildung einer bestimmten Teilmenge. Während die Transformation von Anfragen an Sichten oder integrierte Schemata in Anfragen auf

[*] Diese Arbeit wurde teilweise gefördert vom Land Sachsen-Anhalt unter FKZ: 1987A/0025 "Föderierung heterogener Datenbanksysteme und lokaler Datenhaltungskomponenten zur systemübergreifenden Integritätssicherung" – "**SIGMA**$_{FDB}$ (**S***chema* **I***ntegration and* **G***lobal integrity* **M***aintenance* **A***pproach* $_{for}$ $\mathbf{F}_{ederated}$ $\mathbf{D}_{ata}$ $\mathbf{B}_{ases}$)".

die ursprünglichen Schemata ein schon ausgiebig bearbeitetes Gebiet darstellt, wird andererseits eine entsprechende Behandlung von Integritätsbedingungen nicht oder nur bruchstückweise angegeben.

Im Bereich der Schemaintegration gibt es zwar vielfältige Integrationsmethoden (etwa [SPD92, RPRG94, SS96a]), doch werden, wenn überhaupt, nur einzelne Arten von einfachen Integritätsbedingungen dabei berücksichtigt (vgl. z.B. [RPG95]). In [VA96] wird die Bedeutung von Integritätsbedingungen für die Interoperation von Datenbanksystemen deutlich hervorgehoben und einige zentrale Probleme der Integration von Integritätsbedingungen rudimentär diskutiert. In [CHJ+96] haben wir anhand einer Fallstudie bei der Integration verschiedener Datenbestände die Behandlung von Integritätsbedingungen untersucht. Daraus lassen sich einige allgemeine Gesetzmäßigkeiten ableiten, die nicht nur für die Schemaintegration gelten, sondern sich auch auf Sichtenbildung oder andere Schemaveränderungen wie Transformation, Restrukturierung und teilweise auch Schemaevolution übertragen lassen.

In diesem Beitrag zeigen wir, welche Regeln für die Behandlung von Integritätsbedingungen bei den zuvor erwähnten Basisoperationen zur Schemarestrukturierung (zur Bildung von Sicht, zur Schematransformation und -integration) gelten. Dabei lassen sich nicht alle Arten von Integritätsbedingungen gleich behandeln. Daher unterscheiden wir verschiedene Arten von Integritätsbedingungen und diskutieren diese. Wir beschränken uns aus Platzgründen auf einige Arten von Integritätsbedingungen, die in der Praxis häufig vorkommen.

Im nächsten Abschnitt führen wir zunächst die Basisoperationen ein, die für die Restrukturierung von Schemata benötigt werden. Für diese Basisoperationen stellen wir in Abschnitt 3 Regeln für die Behandlung von Integritätsbedingungen auf. In Abschnitt 4 diskutieren wir, inwieweit diese Regeln auf verschiedene Klassen von Integritätsbedingungen anwendbar sind. Anhand eines Beispieles zeigen wir dann die Anwendung der von uns aufgestellten Regeln (Abschnitt 5) und diskutieren an einem konkreten Problem, wie der Verlust oder die unnötige Abschwächung von bestimmten Integritätsbedingungen vermieden werden kann. Abschließend fassen wir unsere Ergebnisse zusammen und geben einen Ausblick auf weitere Arbeiten.

2 Basisoperationen für Schemarestrukturierung

In diesem Abschnitt sollen kurz die Operationen vorgestellt werden, die zur Restrukturierung von Schemata verwendet werden können. Dabei beschränken wir uns hier auf die extensionale Restrukturierung. Mit einer Extension einer Klasse ist nicht die Menge ihrer aktuellen Objekte gemeint. Statt dessen verwenden wir hier diesen Begriff, um die Menge von möglichen Datenbankobjekten einer Klasse zu bezeichnen. Die Menge der aktuell existierenden Datenbankobjekte stellt damit immer eine Untermenge der Klassenextension dar.

Mit Hilfe der im weiteren diskutierten Operationen ist es möglich, neue Klassen mit Extensionen abzuleiten, die es vorher nicht gab. Gerade dieser Aspekt ist in bestimmte Situationen wichtig:

– Schematransformation aufgrund des Wechsels des zugrundeliegenden Datenmodells
– Definition von Sichten
– Sichtenintegration beim Datenbankentwurf
– Schemaintegration zum Beispiel beim Entwurf föderierter Datenbanken oder beim Aufbau eines 'data warehouse'

Gerade die Betrachtung von Integritätsbedingungen bei der Restrukturierung ist von großer Wichtigkeit, da sie wesentlich die Semantik eines Schemas beeinflussen. Im folgenden werden wir Restrukturierungsoperationen vorstellen, ohne dabei auf die Behandlung von zugeordneten Integritätsbedingungen einzugehen. Diesem Punkt sind die darauffolgenden Abschnitte gewidmet.

Wir schlagen hier eine Restrukturierung vor, die in zwei Phasen abläuft (siehe dazu auch [SS96a]). Es werden die vorhandenen Klassenextensionen in Basisextensionen zerlegt, die danach als Grundlage für die Erzeugung neuer Klassenextensionen durch Zusammensetzen dienen sollen. Wir werden die ursprünglichen Klassenextensionen ab jetzt *Ursprungsextensionen* nennen. Analog dazu werden wir den Begriff *Ursprungsintegritätsbedingung* verwenden, wenn wir uns auf Bedingungen beziehen, die auf Ursprungsextensionen definiert sind.

Um die Zerlegung durchführen zu können, müssen vorher die unterschiedlichen Beziehungen zwischen Ursprungsextensionen betrachtet werden. Die verschiedenen Beziehungen ergeben sich aus der Möglichkeit, daß unterschiedliche Extensionen Informationen über dieselben Realweltsobjekte enthalten können. Aufgrund dieser semantischen Äquivalenz werden nach [SPD92] zwischen zwei Klassenextensionen folgende Beziehungen unterschieden:

– *disjunkte* Klassenextensionen
– *gleiche* Klassenextensionen
– *einschließende* Klassenextensionen
– *überlappende* Klassenextensionen

Leider reichen die Informationen aller Extensionsbeziehungen zwischen jeweils zwei Klassen nicht aus, um alle Beziehungen zwischen Ursprungsextensionen zu bestimmen (siehe [SS96b]). Dies folgt unmittelbar aus der allgemein bekannten Tatsache, daß eine ternäre Beziehung nicht durch drei binäre Beziehungen ersetzt werden kann. Statt dessen müssen alle Klassenextensionen gegenseitig in Beziehung gesetzt werden. Das Ergebnis einer solchen Analyse kann, wie in Abbildung 1 beispielhaft dargestellt, grafisch präsentiert werden. Die horizontalen Striche der vorgegebenen Klassen A, B, C, D und E stellen deren Extensionen dar. Die Klassenextensionen B und C überlappen sich gegenseitig. Die Klasse E ist extensional in Klasse D eingeschlossen, weil Klasse E eine Subklasse der Klasse D ist. Auf der Grundlage der Information über diese Überlappungen ist nun eine Zerlegung in Basisextensionen möglich.

Durch die Mengenoperationen *Schnitt*, *Differenz* und *Selektion* können Basisextensionen aus den Ursprungsextensionen abgeleitet werden. Um bei der Zerlegung einen Informationsverlust zu vermeiden, muß darauf geachtet werden, daß alle möglichen Basisextensionen gefunden werden.

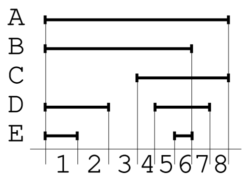

Abbildung1: Extensionale Überlappungen

Die Selektion ist aufgrund der beiden erstgenannten Operationen für die Erzeugung von Basisextensionen nicht unbedingt nötig. Jedoch spielt diese Operation bei der Ableitung von Sichten eine wichtige Rolle. Entsprechend der Forderung nach vollständigen Basisextensionen müssen bei der Anwendung der Selektion auf einer Klassenextension zwei kleinere Extensionen gebildet werden: Eine Extension, die genau die Objekte erhält, die die Selektionsbedingung erfüllen und eine Extension für die übrigen Objekte (Selektion mit der negierten Selektionsbedingung).

Als Ergebnis der Anwendung dieser Operationen entstehen Basisextensionen, die, wie in Abbildung 1 dargestellt, durchnumeriert werden können. Nachdem nun alle Basisextensionen durch die genannten Zerlegungsoperationen erzeugt wurden, können neue Klassenextensionen durch *Vereinigung* von Basisextensionen erzeugt werden. Welche Basisextensionen zu welchen neuen Klassen zusammengesetzt werden, wird beim Entwurf festgelegt. Ein hilfreicher Algorithmus im Zusammenhang mit integrierten Sichten oder Schemata ist in [SS96a] gegeben. Die Operation zum Zusammensetzen neuer Klassenextensionen entspricht damit der Mengenvereinigung.

Zusammenfassend werden für die extensionale Restrukturierung die folgenden Mengenoperationen benötigt: Selektion, Differenz, Schnitt und Vereinigung. Der Behandlung von Integritätsbedingungen bei der Anwendung dieser Operationen sind die nächsten Kapitel gewidmet.

3 Regeln für die Behandlung von Integritätsbedingungen

Im folgenden führen wir zunächst eine spezielle Notation ein, die wir bei der Formulierung der Integrationsregeln für Integritätsbedingungen nutzen werden:

$\Theta_\phi(E)$ bedeutet, daß die Integritätsbedingung ϕ für die Menge E von Objekten gilt. Für alle nicht explizit anders gebundenen Variablen in ϕ ist damit E der Definitionsbereich.

Neben dieser Notation benötigen wir noch eine einfache Klassifikation von Integritätsbedingungen für die Integrationsregeln. Wir unterscheiden im folgenden zwischen *Objektbedingungen* und *Klassenbedingungen*:

Objektbedingungen: Diese Art von Integritätsbedingungen zeichnet sich dadurch aus, daß ihre Erfüllung jeweils durch eine Überprüfung an den einzelnen Objekten einer Extension unabhängig voneinander festgestellt werden kann. Beim Einfügen oder Ändern kann eine solche Integritätsbedingung damit nur durch Betrachtung der neuen Objekte bzw. der geänderten Objekte überprüft werden. Ein typisches Beispiel für solche Integritätsbedingungen ist die Einschränkung des Wertebereiches einzelner Attribute für die Objekte einer Klasse.

Klassenbedingungen: Im Gegensatz zu den Objektbedingungen sind Klassenbedingungen dadurch charakterisiert, daß für ihre Überwachung immer mehrere oder alle Objekte einer Klasse überprüft werden müssen. Typische Beispiele sind Schlüsseleigenschaften, referentielle Integrität und Bedingungen, die Aggregationsfunktionen verwenden (also z.B. die Summe der Gehälter von Mitarbeitern einer Abteilung beschränken). Für diese Bedingungen reicht es bei Änderungsoperationen nicht aus, nur das neue, gelöschte oder veränderte Objekt zu betrachten. Wenn wir hier von „Klassen" sprechen, erlauben wir auch, verschiedene „Klassen" zu einer zusammenzufassen, so daß z.B. referentielle Integritätsbedingungen immer auch als Klassenbedingungen formuliert werden können.

Im folgenden geben wir für die vier im vorherigen Abschnitt vorgestellten Basisoperationen Regeln an, die die Gesetzmäßigkeiten beschreiben, die bei diesen Operationen für Integritätsbedingungen gelten. Dabei beschränken wir uns hier zunächst auf Objektbedingungen. Inwieweit diese Regeln auch auf Klassenbedingungen übertragbar sind, diskutieren wir später.

Selektion. Die Anwendung einer Selektion σ_ψ bedeutet vom logischen Standpunkt, daß wir genau die Objekte einer Extension E in einer Teilmenge $(\sigma_\psi(E))$ zusammenfassen, die die vorgegebene Selektionsbedingung ψ erfüllen. Eine Selektionsbedingung ist dabei immer eine Objektbedingung, die für die Ergebnismenge $\sigma_\psi(E)$ auch als Integritätsbedingung aufgefaßt werden kann. Gilt für die Ursprungsextension E bereits eine Objektbedingung ϕ, so folgt, daß für $\sigma_\psi(E)$ die logische Konjunktion $\phi \wedge \psi$ gelten muß:

$$\Theta_\phi(E) \to \Theta_{\phi\wedge\psi}(\sigma_\psi(E))$$

Differenz/Teilmenge. Die Operationen zur Bildung einer Mengendifferenz und zur Bildung einer Teilmenge unterliegen hinsichtlich der Behandlung von Integritätsbedingungen denselben Gesetzmäßigkeiten, da das Ergebnis einer Mengendifferenz zweier Extensionen immer Teilmenge einer dieser Extensionen ist. Wenn nun eine Objektbedingung ϕ für eine Extension gilt, gilt sie natürlich auch für jede Teilmenge dieser Extension. Dies liegt daran, daß die Bedingung für jedes einzelne Objekt der Extension unabhängig von anderen Objekten gilt. Als Regel können wir dies nun wie folgt ausdrücken (dabei nutzen wir in der Formulierung der Regel die implizite Eigenschaft aus, daß E_1 immer Teilmenge von $E_1 \cup E_2$ ist):

$$\Theta_\phi(E_1 \cup E_2) \to \Theta_\phi(E_1)$$

Für die Anwendung dieser Regel ist wichtig, daß die Ursprungsextension, für die ϕ gilt, sich als Vereinigung zweier Extensionen darstellen läßt. Wenn wir eine

Ursprungsextension E mit einer Teilmenge E_1 haben, dann können wir E auch als $E_1 \cup (E - E_1)$ darstellen und die Regel anwenden.

Wollen wir diese Regel für die Mengendifferenz $E' - E''$ anwenden, so müssen wir E_1 durch $E' - E''$ und E_2 durch $E' \cap E''$ ersetzen. Da immer $E' = (E' - E'') \cup (E' \cap E'')$ gilt, erhalten wir:

$$\Theta_\phi(E') \rightarrow \Theta_\phi(E' - E'')$$

Schnittmenge. Im Gegensatz zur Vereinigung von Extensionen kann für die Schnittmenge von Extensionen die Konjunktion der jeweiligen Integritätsbedingungen gebildet werden. Da jedes Objekt in $E_1 \cap E_2$ sowohl in E_1 als auch in E_2 enthalten ist, erfüllt es natürlich gleichzeitig die Integritätsbedingung ϕ der Extension E_1 und ψ der Extension E_2:

$$\Theta_\phi(E_1), \Theta_\psi(E_2) \rightarrow \Theta_{\phi \wedge \psi}(E_1 \cap E_2)$$

Vereinigung. Bei der Vereinigung von Extensionen E_1 und E_2 können wir im allgemeinen nur das Wissen ausnutzen, daß jedes Objekt der Vereinigung $E_1 \cup E_2$ auch Objekt der Extension E_1 *oder* der Extension E_2 ist (wobei dieses *Oder* ein logisches Oder und kein Entweder-Oder ist). Wenn also für die Objekte in E_1 eine Objektbedingung ϕ gilt und ψ für die Objekte in E_2, dann wissen wir von jedem einzelnen Objekt in $E_1 \cup E_2$ nur, daß es mindestens eine dieser Objektbedingungen erfüllt:

$$\Theta_\phi(E_1), \Theta_\psi(E_2) \rightarrow \Theta_{\phi \vee \psi}(E_1 \cup E_2)$$

In Abbildung 2 haben wir die zuvor besprochenen Regeln zusammengefaßt (Regeln (1) bis (4)). Die Regeln (5) und (6) behandeln noch einige elementare logische Eigenschaften, die durch die vorhergehenden Regeln nicht erfaßt werden. Dabei handelt es darum, daß jeder einzelne Teil einer Konjunktion für sich alleine auch als Integritätsbedingung aufgefaßt werden kann (Regel (5)) und daß jede Integritätsbedingung durch disjunktives Hinzufügen einer beliebigen anderen Bedingung abgeschwächt werden kann (Regel (6)). Regel (7) drückt aus, daß zwei Integritätsbedingungen, die für dieselbe Extension gelten, auch als Konjunktion zusammengefaßt werden dürfen. Die Regel (8) gehört aus Vollständigkeitsgründen auch dazu, da trivialerweise für jede beliebige Extension die Bedingung, die immer erfüllt ist, gilt.

Regel (7) ist eigentlich nur ein Spezialfall der Regel (3), der sich daraus ergibt, daß E_1 und E_2 gleich gesetzt werden, also dieselbe Menge von Objekten bezeichnen. Man kann in diesem Fall auch die Regel (4) anwenden, da die Vereinigung einer Menge mit sich selbst auch wieder dieselbe Menge ergibt. Die Regel (4) liefert uns aber eine Disjunktion der beiden Integritätsbedingungen zurück. Dieses Ergebnis ist formal korrekt, aber nicht das gewünschte. Ziel ist es, eine möglichst scharfe Integritätsbedingung zu erhalten, in diesem Fall also die Konjunktion von ϕ und ψ, wie sie Regel (7) liefert.

Die Regeln (5), (6) und (7) repräsentieren allgemeine logische Eigenschaften. Daher gelten sie nicht nur für Objektbedingungen sondern für beliebige Integritätsbedingungen. Dies trifft auf die Regeln (1) bis (4) nicht zu.

Seien ϕ und ψ Objektbedingungen, dann gilt:

$$\Theta_\phi(E) \rightarrow \Theta_{\phi\wedge\psi}(\sigma_\psi(E)) \tag{1}$$

$$\Theta_\phi(E_1 \cup E_2) \rightarrow \Theta_\phi(E_1) \tag{2}$$

$$\Theta_\phi(E_1), \Theta_\psi(E_2) \rightarrow \Theta_{\phi\wedge\psi}(E_1 \cap E_2) \tag{3}$$

$$\Theta_\phi(E_1), \Theta_\psi(E_2) \rightarrow \Theta_{\phi\vee\psi}(E_1 \cup E_2) \tag{4}$$

$$\Theta_{\psi\wedge\phi}(E) \rightarrow \Theta_\psi(E) \tag{5}$$

$$\Theta_\psi(E) \rightarrow \Theta_{\psi\vee\phi}(E) \tag{6}$$

$$\Theta_\phi(E), \Theta_\psi(E) \rightarrow \Theta_{\phi\wedge\psi}(E) \tag{7}$$

$$\rightarrow \Theta_{\mathsf{true}}(E) \tag{8}$$

Abbildung2: Zusammenstellung der Regeln für *Objektbedingungen.*

4 Anwendbarkeit der Regeln auf Klassenbedingungen

Die im vorangegangenen Kapitel diskutierten Regeln gelten für Objektbedingungen. Damit stellt sich natürlich die Frage, ob sie auf Klassenbedingungen übertragbar sind oder ob für diese Art von Integritätsbedingungen ganz andere Regeln gelten. In diesem Papier beschränken wir uns auf die Betrachtung einiger spezieller Arten von Klassenbedingungen, die in der Praxis häufig auftreten. Hierzu zählen sicher *Eindeutigkeitsbedingungen*, wie etwa *Schlüsselbedingungen*, und Bedingungen für *referentielle Integrität.* Ferner betrachten wir Bedingungen, die über die bekannten Aggregationsfunktionen max und min definiert sind.

Eindeutigkeitsbedingungen. Die Regeln (1) bis (3) gelten auch für Eindeutigkeitsbedingungen (unique). Aus der Selektionsregel folgt, daß die Objekte der Ergebnismenge neben dem Selektionsprädikat auch die Eindeutigkeitsbedingung erfüllen müssen. Man kann sich einfach klar machen, daß die Eindeutigkeitsbedingung auf alle Untermengen übertragbar ist. Das heißt, wenn ein Objekt eindeutig in einer Menge identifiziert werden kann, dann kann es auch eindeutig in jeder Untermenge dieser Menge identifiziert werden. Diese Tatsache kann auch zur Begründung der Erfüllung der Differenz- und die Schnittregel herangezogen werden.

Eines jedoch muß bei allen drei Regeln beachtet werden: Eine ursprünglich auf der Obermenge definierte Bedingung wird durch die Anwendung der Regeln (1) bis (3) abgeschwächt. Dieser Aspekt wird durch das folgende Beispiel verdeutlicht: Wenn eine Extension Mitarbeiter, auf der eine Eindeutigkeitsbedingung definiert ist, in zwei Basisextensionen Projektgruppe1 und Projektgruppe2 zerlegt wird, können wir davon ausgehen, daß für beide Basisextensionen die Eindeutigkeitsbedingung gilt. Jedoch haben wir durch die Zerlegung einen Informationsverlust, der darauf beruht, daß die Beziehung zwischen den Basisextensionen nicht berücksichtigt wird. In diesem Fall ist es möglich, daß in beiden Basisextensionen jeweils ein Objekt mit denselben Schlüsselattributwerten eingefügt werden kann, ohne daß eine der beiden abgeleiteten Eindeutigkeitsbedingungen verletzt wird. Aus Sicht der Ursprungsextension ist hier aber die Ursprungsintegritätsbedingung verletzt.

Daher benötigen wir eine zusätzliche Regel, die sozusagen die Ursprungsintegritätsbedingung weiterhin aufrecht erhält. Daraus läßt sich die folgende Aquivalenzregel ableiten, die besagt, daß eine Bedingung ϕ, die für eine Extension E gilt, auch für die Vereinigung der beiden zerlegten Mengen E_1 und E_2 ($E \equiv E_1 \cup E_2$) gelten muß und umgekehrt:

$$\Theta_\phi(E) \leftrightarrow \Theta_\phi(E_1 \cup E_2) \tag{9}$$

Diese Aquivalenzregel gilt nicht nur für Eindeutigkeitkeitsbedingungen, sondern auch für die später beschriebenen referentiellen Integritätsbedingungen sowie für über Aggregatfunktionen definierte Bedingungen.

Die Vereinigungsregel (4) gilt jedoch nicht für Eindeutigkeitsbedingungen. Der Grund hierfür ist, daß von einer Eindeutigkeitsbedingung, die auf einer Ursprungsextension gilt, nicht geschlossen werden kann, daß diese Bedingung auch für die vereinigte Extension gilt, auch dann nicht, wenn auf beiden Ursprungsextensionen die gleiche Eindeutigkeitsbedingung (das heißt unique auf demselben Attribut bzw. auf derselben Attributkombination) definiert ist.

Betrachten wir auch hierzu ein Beispiel: Nehmen wir an, es existieren zwei Extensionen Projektgruppe1 und Projektgruppe2, auf denen jeweils eine Eindeutigkeitsbedingung definiert ist. Die Mitarbeiter der Projektgruppe1 werden über ihre Versicherungsnummer identifiziert, während die Mitarbeiter der Projektgruppe2 anhand ihrer Vor- und Nachnamen unterschieden werden. Wenn wir nun diese beiden Extensionen zu einer Extension Mitarbeiter vereinigen, müssen wir uns überlegen, wie wir mit den Ursprungsbedingungen verfahren. Wir können auf keinen Fall annehmen, daß eine dieser beiden Eindeutigkeitsbedingungen für die gesamte Vereinigung gelten muß. Andererseits dürfen wir diese Bedingungen nicht einfach unterschlagen, weil dadurch das Prinzip nach Vollständigkeit verletzt wird und die Semantik der Ursprungsextensionen verloren geht.

Um dies zu vermeiden, kann für die Objekte der vereinigten Extension eine sogenannte *Diskriminante* [GCS95] eingeführt werden, die es ermöglicht, die Zuordnung der einzelnen Objekte zu den Ursprungsextensionen zu wahren. Unter Einbeziehung einer Diskriminante, die in unserem Beispiel etwa ein Attribut gruppe sein könnte, können dann für die vereinigte Extension Bedingungen abgeleitet werden, die zu den Ursprungsbedingungen korrespondieren. Für das obige Beispiel muß in diesem Fall die folgende Bedingung für die Extension Mitarbeiter abgeleitet werden: Für alle Mitarbeiter, deren Diskriminante gleich "Projektgruppe1" ist, gilt, daß sie unterschiedliche Versicherungsnummern aufweisen müssen. Analog gilt für alle Mitarbeiter, deren Diskriminante gleich "Projektgruppe2" ist, daß sie unterschiedliche Vor- und Nachnamenkombinationen haben müssen. Für die vereinigte Extension muß dann die *konjunktive* Verknüpfung dieser beiden Bedingungen gelten.

Referentielle Integrität innerhalb einer Klasse. Bei der Betrachtung der referentiellen Integritätsbedingungen (ref) beschränken wir uns auf Bedingungen, die sich auf Referenzen innerhalb einer Extension beziehen. In unserem Beispiel könnte es für die Ursprungsextension Mitarbeiter eine Bedingung geben, die besagt, daß das Attribut Vorgesetzter auf einen existierenden Mitarbeiter verweisen muß.

Für solche Bedingungen gilt die Selektionsregel nicht. Der Grund dafür ist, daß in unserem Ansatz die Ursprungsextensionen komplett durch die Anwendung der Selektionsoperation in Basisextensionen zerlegt werden, wodurch ehemals gültige Referenzen ungültig werden können, wenn die referenzierten Objekte in anderen Basisextensionen liegen. Anders formuliert, wir können nicht davon ausgehen, daß alle referentiellen Integritätsbedingungen eingehalten werden, wenn möglicherweise durch die Selektionsbedingung referenzierte Objekte nicht in der Ergebnismenge enthalten sind. Aus demselben Grund gelten die Differenz- und die Schnittregel nicht für referentielle Integritätsbedingungen.

Im Gegensatz dazu gilt die Vereinigungsregel für referentielle Integritätsbedingungen. Dies liegt daran, daß die Vereinigungsextension mindestens eine Bedingung der Ursprungsextensionen erfüllt und somit die logische Oder-Verknüpfung der beiden Bedingungen immer erfüllt ist. Wenn wir die Ursprungsextensionen Projektgruppe1 und Projektgruppe2 zur Extension Mitarbeiter vereinigen, können wir davon ausgehen, daß alle Referenzen gültig sind, da alle referenzierten Mitarbeiter durch die Vereinigung in der Ergebnismenge enthalten sind. Es ist zu beachten, daß die logische Oder-Verknüpfung der beiden Ursprungsbedingungen insgesamt gesehen zu einer Abschwächung führt.

Bedingungen definiert über die Aggregatfunktionen max und min. Zum Abschluß dieses Kapitels diskutieren wir die Anwendbarkeit unseres Regelsatzes auf sogenannten Maximum- bzw. Minimumbedingungen der Gestalt 'max(x) θ z' (bzw. 'min(x) θ z'), wobei x ein Attribut ist, z eine Konstante darstellt und θ ein Vergleichsoperator ist.

Bei den Regeln (1) bis (3) hängt es davon ab, mit welchen Vergleichsoperator die Funktion max (bzw. min) und verknüpft wird, damit die zugehörige Maximum- bzw. Minimumbedingung gilt. Für Maximumbedingungen läßt sich zeigen, daß die Regeln (1) bis (3) nur dann gelten, wenn für die Vergleichsoperatoren folgendes gilt: $\theta \in \{\leq, <\}$. Wenn also das Maximum einer Ursprungsextension kleiner (bzw. kleiner gleich) einem bestimmten Wert ist, dann ist auch in jedem Fall das Maximum jeder Untermenge dieser Extension kleiner (bzw. kleiner gleich) diesem bestimmten Wert. Wenn jedoch für die Ursprungsextension etwa die Bedingung gilt, daß das Maximum eines bestimmten Wertes größer als eine bestimmte Konstante sein muß (max(x)>10), können wir nicht garantieren, daß das auch für jede Untermenge dieser Extension gilt. Es kann der Fall eintreten, in dem gerade das Element, das die Bedingung wahr werden läßt, nicht in der Untermenge vorhanden ist. Analog gelten bei Minimumbedingungen die Regeln (1) bis (3) nur dann, wenn für die Vergleichsoperatoren gilt, daß sie aus $\{\geq, >\}$ sind.

Die Frage, ob die Vereinigungsregel für Maximum- und Minimumbedingungen gilt, ist nicht ganz einfach zu beantworten. Offensichtlich funktioniert diese Regel für Maximumbedingungen, wenn die zugehörigen Vergleichsoperatoren aus $\{\geq, >\}$ sind. Analog muß für Minimumbedingungen gelten, daß die zugehörigen Vergleichsoperatoren aus $\{\leq, <\}$ sind.

Das folgende Beispiel soll diese beiden Aussagen verdeutlichen. Nehmen wir an, daß auf einer der Ursprungsextensionen die Bedingung max(x)>10 gilt. Dann können wir davon ausgehen, daß diese Bedingung auch für die Vereinigung gelten muß, da diese Bedingung bereits durch die Objekte der entsprechenden Ursprungsextension erfüllt wird. Durch die Vereinigung kann max(x) nur noch größer werden, weil Objekte der zweiten Ursprungsextension hinzukommen können, deren Attribut x den Wert max(x) der ersten Ursprungsextension übersteigt. Analog wird auch zum Beispiel eine Minimumbedingung min(x)<5, die auf einer Ursprungsextension definiert ist, auch von der vereinigten Extension eingehalten. In diesem Fall kann min(x) durch die Vereinigung nur kleiner werden, wenn eben in der anderen Ursprungsextension ein Objekt mit einem kleineren x-Wert enthalten ist.

Bei genauerer Betrachtung der obigen Beispiele stellen wir jedoch fest, daß die Anwendung der Vereinigungsregel zu einer Abschwächung der Ursprungsintegritätsbedingungen führt. Wenn wir von dem in Abbildung 3 dargestellten Beispielszenario ausgehen und das Objekt “Objekt1” löschen wollen, ergibt sich das Problem, daß dieses Objekt global, das heißt auf der vereinigten Extension gelöscht werden darf, da die Maximumbedingung hier nicht verletzt wird. Auf lokaler Ebene, das heißt auf dem Blickwinkel der Ursprungsextension, wird aber die Ursprungsintegritätsbedingung (max(x)>10) verletzt. Dieses Problem kann, wie bei der Eindeutigkeitsbedingung, mit der “Diskriminanten”-Lösung behoben werden.

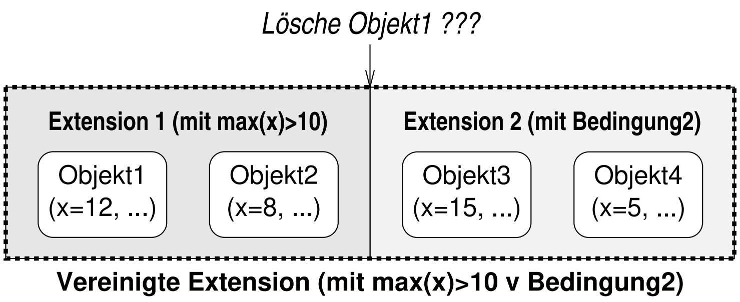

Abbildung3: Beispielszenario

Wir haben nun gezeigt, daß die Vereinigungsregel für Maximum- und Minimumbedingungen gilt, wenn der Vergleichsoperator aus $\{\geq, >\}$ bzw. aus $\{\leq, <\}$ ist. Nun zeigen wir, daß die Vereinigungsregel darüber hinaus auch für die folgenden Fälle gilt: (1) Maximumbedingung mit einem Vergleichsoperator aus $\{\leq, <\}$ und (2) Minimumbedingung mit einem Vergleichsoperator aus $\{\geq, >\}$. Dazu ist folgende Überlegung notwendig: Eine Maximumbedingung der Form 'max(x) θ_1 z' mit θ_1 aus $\{\leq, <\}$ kann in eine Objektbedingung der Form 'x θ_1 z' umformuliert werden. Analog kann eine Minimumbedingung der Form 'min(x) θ_2 z' mit θ_2 aus $\{\geq, >\}$ in eine Objektbedingung der Form 'x θ_2 z' übertragen werden. Aufgrund der Tatsache, daß für Objektbedingungen die Vereinigungsregel immer gilt, können wir hier schließen, daß sie auch für Maximum- bzw. Minimumbedingungen der obigen Form immer gilt.

Zum Abschluß dieses Kapitel sind die bisher gewonnenen Erkenntnisse in Abbildung 4 zusammengefaßt.

Regel	**unique**	**ref**	$\mathbf{max}_\theta$	$\mathbf{min}_\theta$
(1) *Selektion*	$\surd$	—	$\theta \in \{\leq, <\}$	$\theta \in \{\geq, >\}$
(2) *Differenz*	$\surd$	—	$\theta \in \{\leq, <\}$	$\theta \in \{\geq, >\}$
(3) *Schnitt*	$\surd$	—	$\theta \in \{\leq, <\}$	$\theta \in \{\geq, >\}$
(4) *Vereinigung*	—	$\surd$	$\theta \in \{\geq, >, <, \leq\}$	$\theta \in \{\leq, <, \geq, >\}$

Abbildung4: Anwendbarkeit der Regeln auf Klassenbedingungen

5 Beispielhafte Anwendung der Regeln

Im Abschnitt 3 und 4 haben wir die einzelnen Regeln zur Behandlung von Integritätsbedingungen diskutiert. Zusammen bilden sie einen Regelsatz. Wir wollen an dieser Stelle einige Richtlinien zur Verwendung dieses Regelsatzes geben, damit mit möglichst wenig Aufwand eine Restrukturierung ohne Informationsverlust durchgeführt werden kann. Das folgende Beispiel soll dazu dienen, die Anwendung der Regeln entsprechend den Richtlinien zu veranschaulichen. Die Restrukturierung findet in zwei Etappen statt: (1) Zerlegung der Ursprungsextensionen in Basisextensionen und (2) Vereinigung der Basisextensionen zu abgeleiteten Extensionen.

Zerlegung von Extensionen

Die Zerlegung der Ursprungsextensionen basiert auf der extensionalen Analyse. Aufgrund dieser Analyse ist bekannt, mit welchen Mengenoperationen die Basisextensionen erzeugt werden können.

In unserem Beispiel betrachten wir zwei Extensionen, die Mitarbeiter jeweils eines bestimmten Projektes beinhalten. Das Ergebnis der extensionalen Analyse ist in Abbildung 5 dargestellt. Es kann Mitarbeiter geben, die jeweils nur einem Projekt zugeordnet sind. Zusätzlich können aber auch Mitarbeiter existieren, die beiden Projekten angehören (horizontaler Überlappungsbereich).

Abbildung5: Extensionale Überlappungen

Entsprechend dieser Informationen kann die Zerlegung der Ursprungsextensionen in drei Basisextensionen durchgeführt werden. Das Ergebnis der Zerlegung ist im mittleren Teil der Abbildung 6 dargestellt.

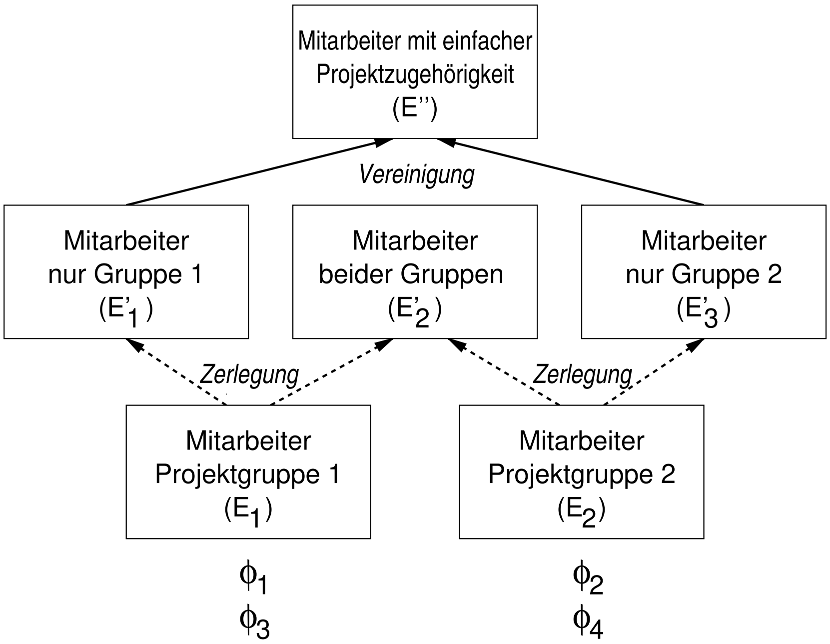

Abbildung6: Zerlegung und Vereinigung von Extensionen am Beispiel

In unserem Beispiel gehen wir davon aus, daß den beiden Ursprungsextensionen verschiedene Integritätsbedingungen zugeordnet sind:

$$\begin{aligned}
&\Theta_{\phi_1}(E_1),\ \text{mit } \phi_1 \equiv \forall e : e.\text{qualifikation} > 4 \\
&\Theta_{\phi_3}(E_1),\ \text{mit } \phi_3 \equiv \forall e_1, e_2 : e_1.\text{pid} = e_2.\text{pid} \Longrightarrow e_1 = e_2 \\
&\Theta_{\phi_2}(E_2),\ \text{mit } \phi_2 \equiv \forall e : e.\text{qualifikation} > 3 \\
&\Theta_{\phi_4}(E_2),\ \text{mit } \phi_4 \equiv \exists e_1 : \forall e_2 :\ e_1.\text{gehaltsstufe} \geq e_2.\text{gehaltsstufe} \\
&\qquad\qquad\qquad\qquad\qquad\qquad \wedge\ e_1.\text{gehaltsstufe} > 3
\end{aligned}$$

Die Integritätsbedingungen ϕ_1 und ϕ_3 sind Objektbedingungen während ϕ_2 und ϕ_4 Klassenbedingungen darstellen; ϕ_2 ist eine Eindeutigkeitsbedingung über das Attribut pid (in der Anwendung verwendete Mitarbeiteridentifikatoren) während ϕ_4 eine Maximumbedingung (entspricht max(gehaltstufe)>3) darstellt. Die letzte Bedingung sagt aus, daß es in der Projektgruppe 2 mindestens einen Mitarbeiter geben muß, dessen Gehaltsstufe die Stufe 3 übersteigt.

Es gelten zwischen den Basisextensionen und den Ursprungsextensionen folgende Mengenbeziehungen:

$$\begin{aligned}
E'_1 &= E_1 - E_2 \\
E'_2 &= E_1 \cap E_2 \\
E'_3 &= E_2 - E_1
\end{aligned}$$

Nachdem die Mengenbeziehungen zwischen den Basisextensionen und Ursprungsextensionen bekannt sind, können die Integritätsbedingungen der Basisextensionen entsprechend den im vorherigen Abschnitt genannten Regeln abgeleitet werden. Dazu werden die Integritätsbedingungen bei der Zerlegung getrennt betrachtet. Wenig Probleme bereiten die Objektbedingungen. Sie können durch Anwenden der Regel (2) und (9) abgeleitet werden:

$$\begin{aligned}
\Theta_{\phi_1}(E_1) &\stackrel{\text{Regel } 9}{\longrightarrow} \Theta_{\phi_1}(E'_1 \cup E'_2) \stackrel{\text{Regel } 2}{\longrightarrow} \Theta_{\phi_1}(E'_1) \\
\Theta_{\phi_1}(E_1) &\stackrel{\text{Regel } 9}{\longrightarrow} \Theta_{\phi_1}(E'_2 \cup E'_1) \stackrel{\text{Regel } 2}{\longrightarrow} \Theta_{\phi_1}(E'_2) \\
\Theta_{\phi_2}(E_2) &\stackrel{\text{Regel } 9}{\longrightarrow} \Theta_{\phi_2}(E'_2 \cup E'_3) \stackrel{\text{Regel } 2}{\longrightarrow} \Theta_{\phi_2}(E'_2) \\
\Theta_{\phi_2}(E_2) &\stackrel{\text{Regel } 9}{\longrightarrow} \Theta_{\phi_2}(E'_3 \cup E'_2) \stackrel{\text{Regel } 2}{\longrightarrow} \Theta_{\phi_2}(E'_3)
\end{aligned}$$

Die Übertragung der Klassenbedingungen hingegen bereitet mehr Probleme. Die Regel (2) angewendet auf eine Eindeutigkeitsbedingung führt zu einer Abschwächung. Diese Abschwächung kann dadurch abgefangen werden, daß das Ergebnis der Regel (9) als eigenständige Bedingung betrachtet wird.

Nun könnte man fordern, daß immer nur die Regeln angewendet werden sollen, die zu keiner Abschwächung der Integritätsbedingung führen. Jedoch könnte es passieren, daß in selektiven Sichten solche umgewandelten starken Integritätsbedingungen im Gegensatz zu abgeschwächten Integritätsbedingungen nicht übernommen werden können. Um den Semantikverlust so gering wie möglich zu halten, empfehlen wir daher die gemeinsame Anwendung der Regeln, die zu keiner Abschwächung der Integritätsbedingung führen und die Anwendung abschwächender Regeln. Daher wird bei der Behandlung der Eindeutigkeitsbedingung die Regel (2) und die Regel (9) angewendet.

$$\Theta_{\phi_3}(E_1) \xrightarrow{\text{Regel 9}} \Theta_{\phi_3}(E_1' \cup E_2') \xrightarrow{\text{Regel 2}} \Theta_{\phi_3}(E_1')$$
$$\Theta_{\phi_3}(E_1) \xrightarrow{\text{Regel 9}} \Theta_{\phi_3}(E_2' \cup E_1') \xrightarrow{\text{Regel 2}} \Theta_{\phi_3}(E_2')$$
$$\Theta_{\phi_3}(E_1) \xrightarrow{\text{Regel 9}} \Theta_{\phi_3}(E_1' \cup E_2')$$

Im Gegensatz zu der Eindeutigkeitsbedingung können Bedingungen wie max(x)>y und min(x)<y nicht in abgeschwächter Form für die einzelnen entstehenden Basisextensionen hergeleitet werden. Daher kann hier nur Regel (9) genutzt werden:

$$\Theta_{\phi_4}(E_2) \xrightarrow{\text{Regel 9}} \Theta_{\phi_4}(E_2' \cup E_3')$$

Vereinigung von Extensionen

Im Gegensatz zu den Zerlegungsregeln ist es nicht sinnvoll, die Vereinigungsregel auf einzelne Integritätsbedingungen anzuwenden. Um eine Abschwächung der Integritätsbedingung zu vermeiden, müßten nämlich alle Kombination von Integritätsbedingungen gebildet werden und die Regel (4) dementsprechend oft ausgeführt werden, um dann als letzte Aktion die entstehenden Integritätsregeln durch die Anwendung der Regel (7) zusammenzufassen. Statt dessen empfehlen wir vor der Vereinigung die Anwendung der Regel (7), die die einzelnen Integritätsbedingungen einer Basisextension konjunktiv verknüpft.

In unserem Beispiel wollen wir eine Extension konstruieren, die alle Mitarbeiter umfaßt, die jeweils nur einem der beiden Projekte zugeordnet sind. Wie aus der Abbildung 6 ersichtlich, wird diese Extension durch die Vereinigung der Basisextensionen E_1' und E_3' erzeugt.

$$E'' = E_1' \cup E_3'$$

Die konjunktive Verknüpfung durch Regel (7) wird in unserem Beispiel auf den Basisextensionen E_1' und E_2' durchgeführt, bevor Regel (4) angewendet wird. Auf E_3' ist nur eine Bedingung definiert, so daß eine konjunktive Verknüpfung für E_3' nicht durchgeführt werden braucht.

$$\Theta_{\phi_1}(E_1'), \Theta_{\phi_3}(E_1') \xrightarrow{\text{Regel 7}} \Theta_{\phi_1 \wedge \phi_3}(E_1')$$
$$\Theta_{\phi_1}(E_2'), \Theta_{\phi_2}(E_2'), \Theta_{\phi_3}(E_2') \xrightarrow{\text{Regel 7}} \Theta_{\phi_1 \wedge \phi_2 \wedge \phi_3}(E_2') = \Theta_{\phi_1 \wedge \phi_3}(E_2')$$

Da die Bedingung ϕ_2 schwächer als die Bedingung ϕ_1 ist, fällt sie bei der konjunktiven Verknüpfung weg.

Nun wird zur Ableitung der Integritätsbedingungen für die Extension E'' die Regel (4) angewendet.

$$\Theta_{\phi_1 \wedge \phi_3}(E_1'), \Theta_{\phi_2}(E_3') \xrightarrow{\text{Regel 4}} \Theta_{(\phi_1 \wedge \phi_3) \vee \phi_2}(E_1' \cup E_3') = \Theta_{\phi_2}(E'')$$

In diesem Fall konnte die Bedingung $(\phi_1 \wedge \phi_3) \vee \phi_2$ vereinfacht werden, da $\phi_1 \Rightarrow \phi_2$ gilt. Auf der Extension E'' ist nun weder eine Eindeutigkeitsbedingung noch eine Maximumbedingung definiert. Es kann ein Objekt in E'' eingefügt werden, daß weder in E_1 noch in E_2 eingefügt werden kann, da es zum Beispiel sowohl ϕ_3 als auch ϕ_4 verletzt. Die Bedingungen ϕ_3 und ϕ_4 können jedoch nicht auf der Extension E'' überprüft werden, da sie das Vorhandensein weiterer Extensionen voraussetzen. Das bedeutet konkret, daß ein Scheitern einer Einfügeoperation eines Objektes in E'' nicht mit dem Verletzen einer auf dem abgeleiteten Schema sichtbaren Bedingung erklärt werden kann. Dieses Problem ist eng verwandt mit dem *View-Update*-Problem bei selektiven Sichten.

Interessant an dieser Vereinigung ist weiterhin die Tatsache, daß durch die logische Disjunktion sogar die abgeschwächte Eindeutigkeitsregel verschwindet. Dies kann verhindert werden, wenn bei der Erzeugung der Basisextensionen ein künstliches Diskriminantenattribut eingefügt wird. In unserem Beispiel würde dieses Attribut (zum Beispiel gruppe) die Information beinhalten, ob ein Mitarbeiter der Projektgruppe 1 oder 2 oder beiden Gruppen zugeordnet ist. Diese Information kann nun innerhalb der Integritätsbedingungen der Basisextensionen in der Form ausgenutzt werden, daß die ursprünglichen Bedingungen als rechter Teil einer Implikation auftreten und im linken Teil jeweils der Diskriminantentest durchgeführt wird.

$$\Theta_{\phi'_1 \wedge \phi'_3}(E'_1) \text{ mit}$$

$$\begin{aligned} \phi'_1 &\equiv \forall e : e.\text{gruppe} = 1 \Longrightarrow e.\text{qualifikation} > 4 \\ \phi'_3 &\equiv \forall e_1, e_2 : (e_1.\text{gruppe} = 1 \wedge e_2.\text{gruppe} = 1) \\ &\qquad \Longrightarrow (e_1.\text{pid} = e_2.\text{pid} \Longrightarrow e_1 = e_2) \end{aligned}$$

$$\Theta_{\phi'_2}(E'_3) \text{ mit}$$

$$\phi'_2 \equiv \forall e : e.\text{gruppe} = 2 \Longrightarrow e.\text{qualifikation} > 3$$

Solche umgewandelten Integritätsbedingungen können nun entgegen der Regel (4) bei der Vereinigung konjunktiv verknüpft werden.

$$\Theta_{\phi'_1 \wedge \phi'_3}(E'_1), \Theta_{\phi'_2}(E'_3) \longrightarrow \Theta_{\phi'_1 \wedge \phi'_2 \wedge \phi'_3}(E'_1 \cup E'_3) = \Theta_{\phi'_1 \wedge \phi'_2 \wedge \phi'_3}(E'')$$

Durch diese Umwandlung kann die Abschwächung von Integritätsbedingungen bei der Restrukturierung reduziert werden.

Grundsätzlich ist bei der Ableitung der externen Integritätsbedingungen darauf zu achten, daß diese durch Anwenden der Vereinigungsregeln auf den Basisextensionen abgeleitet werden müssen, da bei einer Ableitung auf der Basis anderer schon abgeleiteter Klassenextensionen Integritätsbedingungen verloren gehen können. Damit man in solchen Fällen weiß, auf welche Basisextensionen zurückgegriffen werden muß, ist es sinnvoll, einer jeden abgeleiteten Klassenextension deren Ableitungsvorschrift auf der Grundlage der Basisextensionen zuzuordnen.

6 Zusammenfassung

Daß die Betrachtung von Integritätsbedingungen in vielen Bereichen wichtig und sogar notwendig sein kann, ist offensichtlich. Als Beispiel möchten wir hier kurz auf den Aspekt der Sichtenmaterialisierung [SJ96, KR95] eingehen, dem gerade in Hinblick auf mobile Datenbanken [WSD$^+$95] eine entscheidene Bedeutung zukommt. Wenn mit der materialisierten Sicht auch die für diese Sicht geltenden Integritätsbedingungen, die aus den ursprünglichen Integritätsbedingungen abgeleitet werden können, mobil verfügbar sind, können viele Integritätsverletzungen auch off-line erkannt werden und so viele Probleme bei dem Abgleich der mobilen veränderten Datenbank mit der zentralen Datenbank vermieden werden. Darüber hinaus können Integritätsbedingungen den Benutzern auch dabei helfen, die Semantik eines Schemas bzw. einer Sicht zu verstehen.

In diesem Beitrag haben wir zunächst, basierend auf einfachen Basisoperationen, einen allgemeinen Rahmen für die Restrukturierung von Schemata, sei es zur Bildung von Sichten oder zur Integration verschiedener Schemata, eingeführt, der zunächst eine extensionale Zerlegung und Zusammensetzung vorsieht. Die Zerlegung in Basisextensionen ist dabei eine grundlegende Technik, die auch in anderen Arbeiten für andere Zwecke diskutiert wurde (z.B. in [Bus91]). Für die dabei verwendeten Basisoperationen haben wir allgemeine Regeln angegeben, mit denen auch Integritätsbedingungen entsprechend behandelt werden können. Während diese Regeln auf Objektbedingungen ohne Einschränkungen angewendet werden können, gilt es, bei Klassenbedingungen einige besondere Fälle zu berücksichtigen. Daher haben wir exemplarisch einige in der Praxis häufig vorkommende Arten von Klassenbedingungen analysiert und dabei bereits einige nicht unbedingt zu erwartende Ergebnisse erhalten. Für weitere Arten von Klassenbedingungen, die wir hier aus Platzgründen nicht behandeln konnten, haben wir entsprechende Resultate erhalten.

Bei der Anwendung der Regeln im Beispiel ist als Problem deutlich geworden, daß in einigen Fällen nur schwächere Integritätsbedingungen abgeleitet werden können, als intuitiv zu erwarten wäre. Eine schwäche Integritätsbedingung würde aber bedeuten, daß etwa in der Sicht oder im integrierten Schema Objekte eingefügt werden dürften, die in den ursprünglichen Klassen aufgrund von Integritätsbedingungen nicht zulässig sind. Damit würde die Sicht oder das integrierte Schema semantisch nicht dem Ursprungsschema oder den Ursprungsschemata entsprechen. Für diese Fälle schlagen wir die Einführung einer Diskrimanten und eine entsprechende Veränderung der Integritätsbedingungen vor. Wie wir gezeigt haben, kann damit dann das gewünschte Ergebnis erzielt werden.

Als Fortführung der hier vorgestellten Ergebnisse untersuchen wir die Anwendbarkeit der Regeln auf einige spezielle Arten von Integritätsbedingungen, die wir bislang noch nicht abgedeckt haben. Derzeit entwickeln wir einen Prototyp eines Entwurfs- und Integrationswerkzeuges, der die Berücksichtigung von Integritätsbedingungen bei Schematransformation und -integration sowie Sichtenbildung unterstützen soll. Hierzu werden wir auch die in der Diskussion des Beispiels angesprochene Methodik der Regelanwendung weiter verfeinern, um möglichst effizient optimale Ergebnisse zu erhalten. Für die ebenfalls schon angesprochene Möglichkeit, Diskriminanten einzuführen, um in bestimmten Fällen eine Abschwächung der Integritätsbedingungen zu vermeiden, arbeiten wir gegenwärtig daran, die Einführung von Diskriminanten in Integritätsbedingungen formal korrekt und vollständig zu definieren.

References

[Bus91] R. Busse. Einordnung von Anfrageergebnissen der Objektalgebra in einen Klassenverband und ihre Implementierung. Diplomarbeit, Technische Universität Clausthal, 1991.

[CHJ+96] S. Conrad, M. Höding, S. Janssen, G. Saake, I. Schmitt und C. Türker. Integrity Constraints in Federated Database Design. Preprint 2, Fakultät für Informatik, Universität Magdeburg, April 1996.

[GCS95] M. Garcia-Solaco, M. Castellanos und F. Saltor. A Semantic-Discriminated Approach to Integration in Federated Databases. In S. Laufmann, S. Spaccapietra und T. Yokoi, Herausgeber, *Proc. of the 3rd Int. Conf. on Cooperative Information Systems (CoopIS'95), Vienna, Austria*, S. 19–31, Mai 1995.

[KR95] H. A. Kuno und E. A. Rundensteiner. Materialized Object-Oriented Views in MultiView. In O. Bukhres, M. T. Özsu und M.-C. Shan, Herausgeber, *RIDE-DOM'95, Proc. of the 5th Int. Workshop on Research Issues in Data Engineering: Distributed Object Management, March 6–7, 1995, Taipei, Taiwan*, S. 78–85, Los Alamitos, CA, 1995. IEEE Computer Society Press.

[RPG95] M. P. Reddy, B. E. Prasad und A. Gupta. Formulating Global Integrity Constraints during Derivation of Global Schema. *Data & Knowledge Engineering*, 16(3):241–268, Juli 1995.

[RPRG94] M. P. Reddy, B. E. Prasad, P. G. Reddy und A. Gupta. A Methodology for Integration of Heterogeneous Databases. *IEEE Transactions on Knowledge and Data Engineering*, 6(6):920–933, Dezember 1994.

[SJ96] M. Staudt und M. Jarke. Incremental Maintenance of Externally Materialized Views. In T. M. Vijayaraman, A. P. Buchmann, C. Mohan und N. L. Sarda, Herausgeber, *Proc. of the 22nd Int. Conf. on Very Large Data Bases (VLDB'96)*, S. 75–86, San Francisco, CA, 1996. Morgan Kaufmann Publishers.

[SPD92] S. Spaccapietra, C. Parent und Y. Dupont. Model Independent Assertions for Integration of Heterogeneous Schemas. *The VLDB Journal*, 1(1):81–126, Juli 1992.

[SS96a] I. Schmitt und G. Saake. Integration of Inheritance Trees as Part of View Generation for Database Federations. In B. Thalheim, Herausgeber, *Conceptual Modelling — ER'96, Proc. of the 15th Int. Conf., Cottbus, Germany, October 1996*, *Lecture Notes in Computer Science*, Band 1157, S. 195–210, Berlin, 1996. Springer-Verlag.

[SS96b] I. Schmitt und G. Saake. Schema Integration and View Generation by Resolving Intensional and Extensional Overlappings. In K. Yetongnon und S. Hariri, Herausgeber, *Proc. of the 9th ISCA Int. Conf. on Parallel and Distributed Computing Systems (PDCS'96), Dijon, France, September 1996*, S. 751–758, Six Forks Road, Releigh, NC, 1996. International Society for Computers and Their Application.

[VA96] M. W. W. Vermeer und P. M. G. Apers. The Role of Integrity Constraints in Database Interoperation. In T. M. Vijayaraman, A. P. Buchmann, C. Mohan und N. L. Sarda, Herausgeber, *Proc. of the 22nd Int. Conf. on Very Large Data Bases (VLDB'96), Bombay, India*, S. 425–435, San Francisco, CA, September 1996. Morgan Kaufmann Publishers.

[WSD+95] O. Wolfson, P. Sistla, S. Dao, K. Narayanan und R. Raj. View Maintenance in Mobile Computing. *ACM SIGMOD Record*, 24(4):22–27, 1995.