Compiling State Constraints

Hussien Oakasha and Gunter Saake

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

Email: {oakasha|saake}@iti.cs.uni-magdeburg.de

Tel.: ++49-391-67-12816
Fax: ++49-391-67-12020

May 1998

Abstract

The evaluation of constraints needs in many cases that a large portion of the database to be accessed. This makes the process of integrity checking very difficult to implement efficiently. Thus finding methods to improve the evaluation of integrity constraints is an important area of research in the field of database integrity. Most of these methods are based on simplification principles. One of these methods is presented by Nicolas in [Nic82]. In this method the simplified form of a constraint depends on updating operations performed on database states. For that reason, the simplified form is obtained at update time. In this report we show that, for a given constraint W and an update that is to be performed to a relation R, it is not necessary to do all the steps of the method at run time, but we can do most of these steps at compile time. We do that by developing a representation that stores simplified instances of W together with other information about occurrences of R in W into *meta relations*. The simplified instances stored in the meta relations are obtained form W by applying the same simplification steps of the method, but here we use *generic constants* instead of specific update values. When an update is performed to the relation R, the generic constants in the meta relations are replaced with the update values and a relational algebra expression is performed on the obtained relation, resulting in a set of formulas. We will show that by only applying the third step of the method to the conjunctions of these formulas we can get the simplified form obatined by the simplification method at run time.

Keywords: state constraints, integrity maintenance, constraints compilation, constraints representation.

Contents

Chapter 1

Introduction

In the simplification method presented in [Nic82], the simplified form of a constraint depends on updating operations performed on database states. The type of each of these operations and the update values involved in it have to be known in advance. For that reason, the simplified form is obtained at update time. This report addresses the following problem: The steps of the method include analyzing the quantifier structure of the constraint to define substitutions and search for atomic formulas (i.e., pre-valued literals) that have to be eliminated from the instances of the constraint. Therefore, obtaining the simplified form for the constraint at update time leads to a significant increase in response time of updating operations performed to database states, particularly for transactions. This is a serious drawback of the simplification method [BM88, MH89].

The objective of our work presented in this report is to show that, for a given constraint W and an update that is to be performed to a relation R, it is not necessary to do all the steps of the method at update time. We do that by developing a representation that stores simplified instances of W together with other information about occurrences of R in W into *meta-relations*. The simplified instances stored in the meta-relations are obtained form W by applying the same simplification steps of the method, but here we use *generic constants* instead of specific update values. When an update is performed to the relation R, the generic constants in the meta-relations are replaced with the update values and a relational algebra expression is performed on the obtained relation, resulting in a set of formulas. We will show that in the case of atomic modifying operation by only applying the third step of the method to the conjunctions of these formulas we can get the simplified form obatined by the simplification method at run time.

This report is organized as follows. The representation consists of two meta-relations, denoted by $\mathcal{T}_R^W$ and $\widetilde{\mathcal{T}}_R^W$, that are developed through three stages. Chapter 4 presents the first stage, in which, for a given constraint W and a relation R, substitutions are defined according to the occurrences of R in W. This is done in the same way as the substitutions of the simplification method defined, but using generic constants instead of specific updates values. The formulas obtained by applying these substitutions to W are stored in pair of meta-relations denoted by $\mathcal{F}_R^W$ and $\widetilde{\mathcal{F}}_R^W$. Chapter 5 presents the second stage of developing the representation. In this stage, a pair of meta-relations denoted by $\mathcal{S}_R^W$ and $\tilde{\mathcal{S}}_R^W$, is obtained from $\mathcal{F}_R^W$ and $\tilde{\mathcal{F}}_R^W$ respectively by simplifying formulas stored in $\mathcal{F}_R^W$ and $\tilde{\mathcal{F}}_R^W$ in a process analogous to simplifying the instances of W in the third step of the simplification method. Chapter 6 presents the third stage of developing the representation. In this stage, the two meta-relations $\mathcal{T}_R^W$ and $\tilde{\mathcal{T}}_R^W$ are obtained respectively from $\mathcal{S}_R^W$ and $\tilde{\mathcal{S}}_R^W$ by applying the fourth step of the simplification method to formulas stored in $\mathcal{S}_R^W$ and $\tilde{\mathcal{S}}_R^W$. In all these sections we consider the operations of inserting a tuple into a relation extension and deleting a tuple from a relation extension. In Chapter 7, we will consider transactions of these operations.

Chapter 2

Basic Notation and Definitions

Throughout the rest of the report we assume the following. R is a relation with arity n, $Art(R) = n$. The i^{th} component of R will be denoted by $\$i$. $\{g_1 \ldots g_n\}$ is a set of distinct elements called *generic constants* such that $Dom(R) \cap \{g_1 \ldots g_n\} = \emptyset$, where $Dom(R)$ is the union of all underlying domains of all attributes of the database. W is an integrity constraint that is a closed well-formed form in prenex conjunctive normal form and which satisfy the range-restricted property [Nic82]. S is the current database state, in which W is satisfied. $O(R, u)$ is either an inserting operation, $I(R, u)$, or a deleting operation, $D(R, u)$. Tr is a transaction, i.e., a set of insert and/or delete operations. S' is a new database state obtained from S by $O(R, u)$ or Tr. We will use the symbol 'e' to denote '$-$' (resp. '$+$'), when $O(R, u)$ is an inserting (resp. deleting) operation.

$Arg(\ell)$ is a tuple whose components are arguments of a literal ℓ. The i^{th} argument of ℓ will be denoted as $Arg(\ell)_i$. For example, if ℓ is the literal $\neg R(x, y, z, c)$, then $Arg(\ell) =< x, y, z, c >$ and $Arg(\ell)_3 = z$. $EQ(W)$ is the set of all variables x in the constraint W such that x is either a universally quantified variable governed by an existentially quantified variable in the constraint W, or an existentially quantified variable. $Eq(\alpha, W)$ is the set of all elements (x/g) in the substitution α such that $x \in EQ(W)$. $mod(A)$ is the set of all database states that are models to wffs of the set A. $R\gamma$ is the relation obtained from the relation R by applying the substitution γ to each tuple u in R. $L^-_{R,W}$ (resp.$L^+_{R,W}$) is the set of all negative (resp. positive) occurrences of the relation R in the constraint W. $L_{R,W}$ is the union of $L^-_{R,W}$ and $L^+_{R,W}$. L_W is the set of all literals in W. $UQ(\ell)$ is the set of all variables x in the literal $\ell \in L_W$ such that x is a universally quantified variable not governed by an existentially quantified variable in the constraint W. Finally, The symbol '$\circ$'will be used for the operation of substitutions composition.

Chapter 3

The Simplification Method

The evaluation of constraints needs in many cases that a large portion of the database to be accessed. Thus it can be time consuming. This makes the process of integrity checking very difficult to implement efficiently. For that reason finding methods to improve the evaluation of integrity constraints is an important area of research in the field of database integrity [Nic82, HMN84, Dec87, Llo87, KSS87, QS87, Qia88, LL93]. Most of these methods are based on simplification principles [GMN84]: Given an integrity constraint that is satisfied in the current database state, these methods derive an equivalent but simplified form to the constraint. Except in some special cases, the evaluation cost of the simplified form is less than the evaluation cost of the initial constraint. In this chaper we will introduce the method proposed in [Nic82] by Nicolas.

The simplification method applies to databases that correspond to the model-theoretic view of relational databases and to integrity constraints that satisfy the range restricted property. The simplification method derives a simplified form to the constraint dependent on the type of update operation that will change the current state to a new one and which may violate the constraint.

The simplified form depends on the type of the updating operation which leads to a state change. The method considers the cases in which database states are modified by simple atomic operations of inserting, deleting a tuple in a relation and transactions of such operations. The operation of updating a tuple is considered as a special transaction that consists of a deletion followed conditionally by an insertion.

In this chapter we present how the simplified forms can be obtained for atomic modifications operations and then for transactions of these operations.

3.1 Basic Definitions

The simplified form of W is mainly built by applying to W substitutions defined according to the quantifier structure of W, occurrences of R in W and the components of the tuple u. The following definitions are a slight modification for the characterization of substitutions given procedurally by Nicolas.

Definition 3.1 (The substitution α_ℓ^u)
Let $\ell \in L_{R,W}$. α_ℓ^u is a substitution defined according to the components of the tuple u and the variables of literal ℓ as follows.

$$\alpha_\ell^u = \{(x/u_i) | Arg(\ell)_i = x \in Var(W) \text{ and } Arg(\ell)_j \neq x \text{ for every } j < i\}.$$

□

Informally, for every $i \in [1, n]$, $(x/u_i) \in \alpha_\ell^u$ iff the i^{th} argument of literal ℓ is a variable x that does not appear in any argument of ℓ that precedes the i^{th} argument.

Definition 3.2 (Sets of substitutions $A_{R,W}^{e,u}$ and $G_{R,W}^{e,u}$)

$$A_{R,W}^{e,u} = \{\alpha_\ell^u | \ell \in L_{R,W}^e \text{ and } Arg(\ell\alpha_\ell^u) = u\}.$$

$$G_{R,W}^{e,u} = \{\gamma_\ell^u \mid \gamma_\ell^u = \alpha_\ell^u - Eq(\alpha_\ell^u, W) \text{ and } \alpha_\ell^u \in A_{R,W}^{e,u}\}.$$

□

Remark 3.1 Several remarks can be made about the definitions of $A_{R,W}^{\neg,u}$, and $G_{R,W}^{\neg,u}$:

- $Arg(\ell\alpha_\ell^u) = u$ iff:
  - whenever $Arg(\ell)_i$ $(1 \leq i \leq n)$ is a constant then $Arg(\ell)_i = u_i$; and
  - whenever there exists i and j $(1 \leq i, j \leq n)$ such that $Arg(\ell)_i = Arg(\ell)_j$ then $u_i = u_j$. For example, consider the literal $\ell : \neg R(x, y, x, e)$. For the tuple $u =< a, b, a, e >$, $\alpha_\ell^u = \{(x/a), (y/b)\}$ and $\ell\alpha_\ell^u$ is $\neg R(a, b, a, e)$. Thus $Arg(\ell\alpha_\ell^u) = u$. Contrarily, for the tuple $u =< a, b, c, e >$, $Arg(\alpha_\ell^u) \neq u$.
- The substitution γ_ℓ^u is obtained from $\alpha_\ell^u \in A_{R,W}^{\neg,u}$ by deleting every element (x/c) in α_ℓ^u such that x is either an existentially quantified, or it is a universally quantified variable governed by an existentially quantified one. Thus, $\gamma_\ell^u = \varepsilon$, the identity substitution, iff $\alpha_\ell^u = Eq(\alpha_\ell^u, W)$ For example, let the quantifier structure of W be: $\forall x \exists z \forall y$. Let ℓ be : $\neg R(x, y, z)$. For the tuple $u =< a, b, c >$, $\alpha_\ell^u = \{x/a, y/b, z/c\}$ and $\gamma_\ell^u = \{x/a\}$. If $\forall x$ is replaced by $\exists x$, then $Eq(\alpha_\ell^u, W) = \alpha_\ell^u$, and hence $\gamma_\ell^u = \varepsilon$.
- $A_{R,W}^{\neg,u} = \emptyset$ iff at least one of the following cases occurs:
  - There does not exist any occurrence of the relation R in W.
  - There does not exist any negative occurrence of the relation R in W.
  - For every negative occurrence ℓ of the relation R in the constraint W, $Arg(\ell\alpha_\ell^u) \neq u$.

- $G^{-,u}_{R,W} = \emptyset$ iff $A^{-,u}_{R,W} = \emptyset$.

Assume that the two substitutions γ^u_ℓ and $\gamma^u_{\ell'}$ exist in $G^{-,u}_{R,W}$ such that γ^u_ℓ subsumes $\gamma^u_{\ell'}$. Nicolas has proved that if $W\gamma^u_\ell$ is true in S' then so is $W\gamma^u_{\ell'}$. In this case $W\gamma^u_{\ell'}$ is redundant with respect to $W\gamma^u_\ell$. This point motivates the next definition.

Definition 3.3 (The set of wffs $\Gamma^{e,u}_{R,W}$)
$\Gamma^{e,u}_{R,W}$ is a set of instances of the constraint W defined as follows.

$$\Gamma^{e,u}_{R,W} = \{W\gamma^u_\ell | \gamma^u_\ell \in G^{e,u}_{R,W} \text{ and for every } \gamma^u_{\ell'} \in G^{e,u}_{R,W}, \gamma^u_{\ell'} \not\subset \gamma^u_\ell\}.$$

□

The simplified form of the constraint W for inserting (resp. deleting) a tuple u into the extension of R is denoted by $C^{-,u}_{R,W}$ (resp. $C^{+,u}_{R,W}$). The simplified form maily derived from the instances of set $\Gamma^{e,u}_{R,W}$. Now we will present the steps of the algorithm given by Nicolas to obtain $C^{e,u}_{R,W}$. The algorithm as presented here is reformulated using our notation.

3.2 The Simplified Form $C^{e,u}_{R,W}$

Given the operation $O(R,u)$, the simplified form $C^{e,u}_{R,W}$ is derived from W by doing the following steps:

Step 1. Construct the set of substitutions $A^{e,u}_{R,W}$ using Definition 3.2. **If** $A^{e,u}_{R,W} = \emptyset$, **then** W is unaffected by the $O(R,u)$, let $C^{e,u}_{R,W} = \boldsymbol{T}$.

Step 2. Construct the set of formulas $\Gamma^{e,u}_{R,W}$ using Definition 3.3. **If** $\Gamma^{e,u}_{R,W} = \{W\}$, **then** $C^{e,u}_{R,W} = W$.

Step 3. Let $V = W_1 \wedge \cdots \wedge W_n$, where $\Gamma^{e,u}_{R,W} = \{W_1 \cdots W_n\}$. Replace in V each pre-valued literal by its truth value in the new state S' and apply as much as possible the absorption rules. Let V' be the obtained formula. **If** $V' = \boldsymbol{T}$ (resp. $V' = \boldsymbol{F}$), **then** $C^{e,u}_{R,W} = \boldsymbol{T}$ (resp. $C^{e,u}_{R,W} = \boldsymbol{F}$).

Step 4. Let $V' = W'_1 \wedge \cdots \wedge W'_n$. Remove from V' any W'_i such that there is W'_j $(i \neq j)$ identical to W'_i up to the permutation of the disjunctions, a permutation of the atomic formulas and a renaming of variables. $C^{e,u}_{R,W}$ is the obtained formula.

3.3 Transactions

For a transaction Tr, the simplified form for W is denoted by C^*_W. To present how C^*_W is derived from W we need the following definitions.

Definition 3.4 (The set of substitutions A^*_W and G^*_W)

$$A^*_W = \bigcup_{O(R,u)\in Tr} A^{e,u}_{R,W} \qquad G^*_W = \bigcup_{O(R,u)\in Tr} G^{e,u}_{R,W}.$$

□

Definition 3.5 (Set of wffs Γ^*_W)
Γ^*_W is a set of instances of the constraint W defined as follows.

$$\Gamma^*_W = \{W\gamma | \gamma \in G^*_W \text{ and for every } \gamma' \in G^*_W, \gamma' \not\subset \gamma\}.$$

□

The simplified form C^*_W is derived in the same way from W as the simplified form $C^{e,u}_{R,W}$ was derived from W by steps 1 – 4 presented in previous subsection, but this time in steps 1–2, sets $A^{e,u}_{R,W}$, and $\Gamma^{e,u}_{R,W}$ are replaced by A^*_W and Γ^*_W respectively.

As a main step towards proving that the evaluation of C^*_W in the new state S' can be substituted for the evaluation of W, Nicolas proves the next theorem.

Theorem 3.1 If $\Gamma^*_W = \emptyset$ then $S' \in mod(W)$; otherwise

$$S' \in mod(W) \text{ iff } S' \in mod(\Gamma^*_W).$$

□

In other words, if for every operation $I(R,u) \in Tr$ and $D(R,u) \in Tr$, $\Gamma^{-,u}_{R,W} = \emptyset$ and $\Gamma^{+,u}_{R,W} = \emptyset$ respectively, then W is not affected by any operation of Tr and hence it remains satisfied in the new state S'. If $\Gamma^*_W \neq \emptyset$ then W could be falsified in the new state S' and W remain satisfied in S' iff every instance of W in Γ^*_W is satisfied in S'. As consequence for the previous theorem is the following corollary.

Corollary 3.1 If $\Gamma^\star_W = \emptyset$ then $S' \in mod(W)$; otherwise

$$S' \in mod(W) \text{ iff } S' \in mod(\Gamma^\star_W).$$

Where

$$\Gamma^\star_W = \bigcup_{O(R,u)\in Tr} \Gamma^{e,u}_{R,W},$$

□

Chapter 4

Meta Relations

In this Chapter, we define the first pair of meta relations $\mathcal{F}_R^W$ and $\widetilde{\mathcal{F}}_R^W$. Then we show that the set of wffs $\Gamma_{R,W}^{e,u}$, defined by Definition 3.3, can be obtained by applying a general, but simple, substitution to the tuples of the meta relations $\mathcal{F}_R^W$ and $\widetilde{\mathcal{F}}_R^W$ and then applying a relational algebra expression, that consists of selection and projection operations, to the obtained relation.

We start with definitions of substitutions that will be used to define the tuples of meta relations $\mathcal{F}_R^W$ and $\widetilde{\mathcal{F}}_R^W$.

Definition 4.1 (Substitutions β_ℓ^W, δ_ℓ^W and $\delta_{\ell,\ell'}^W$)
Let $\ell', \ell \in L_{R,W}$. β_ℓ^W, δ_ℓ^W and $\delta_{\ell,\ell'}^W$ are substitutions defined as follows.

$$\beta_\ell^W = \{(x/g_i) | Arg(\ell)_i = x \in Var(W) \text{ and for every } j < i, Arg(\ell)_j \neq x\}.$$

$$\delta_\ell^W = \{(x/g_i) | (x/g_i) \in \beta_\ell^W \text{ and } x \notin EQ(W)\}.$$

$$\delta_{\ell,\ell'}^W = \delta_\ell^W \cup (\delta_{\ell'}^W - \chi).$$

where

$$\chi = \{(x/g_i) | (x/g_i) \in \delta_{\ell'}^W \text{ and } x \in UQ(\ell)\}.$$

□

In other words, $(x/g_i) \in \beta_\ell^W$ iff the i^{th} argument of literal ℓ in W is a variable x that does not appear in any argument of ℓ preceding the i^{th} argument. δ_ℓ^W is obtained from β_ℓ^W by deleting each element (x/g_i) in β_ℓ^W such that x is either an existentially quantified variable, or it is a universally quantified variable governed by an existentially quantified one. Notice that the definition of β_ℓ^W (resp. δ_ℓ^W) is very similar to the definition of substitution α_ℓ^u (resp. γ_ℓ^u) but here we consider generic constants instead of the components of the tuple u.

The next lemma states the relationship between the substitution β_ℓ^W (resp. δ_ℓ^W) and the substitution α_ℓ^u (resp. γ_ℓ^u). Also, it states some properties of these substitutions and the substitution $\delta_{\ell,\ell'}^W$ that will be used in the rest of this chapter.

Lemma 4.1 Let $\lambda_u = \{(g_1, /u), \ldots, (g_n/u_n)\}$. For every $\alpha_\ell^u \in A_{R,W}^{e,u}$ and $\gamma_\ell^u, \gamma_{\ell'}^u \in G_{R,W}^{e,u}$ such that $UQ(\ell) \subset UQ(\ell')$ we have the following:

$$\alpha_\ell^u = (\beta_\ell^W \circ \lambda_u) - \lambda_u \tag{4.1}$$
$$\gamma_\ell^u = (\delta_\ell^W \circ \lambda_u) - \lambda_u \tag{4.2}$$
$$W\gamma_\ell^u \text{ is identical to } W\delta_\ell^W \circ \lambda_u \tag{4.3}$$
$$Arg(\ell\alpha_\ell^u) = Arg(\ell\beta_\ell^W \circ \lambda_u) \tag{4.4}$$
$$\gamma_\ell^u \subset \gamma_{\ell'}^u \text{ iff } (\delta_{\ell,\ell'}^W \circ \lambda_u) - \lambda_u = \gamma_{\ell'}^u \tag{4.5}$$
$$W\gamma_{\ell'}^u \text{ is identical to } W\delta_{\ell,\ell'}^W \circ \lambda_u \text{ iff } \gamma_\ell^u \subset \gamma_{\ell'}^u \tag{4.6}$$

□

Proof:.

1. Let $(x/c) \in \alpha_\ell^u$. From Def. 3.1, $Arg(\ell\alpha_\ell^u) = u$. Therefore, there exists i $(1 \leq i \leq n)$ such that $Arg(\ell)_i = x$ and $u_i = c$. From Def. 4.1, $Arg(\ell)_i = x$ implies that $(x/g_j) \in \beta_\ell^W$, for some j $(1 \leq j \leq i)$. We have two cases:

   $i = j$: In this case, $(x/g_i) \in \beta_\ell^W$ and $u_i = c$ implies $(g_i/c) \in \lambda_u$. Thus (x/c) in $(\beta_\ell^W \circ \lambda_u)$.

   $j < i$: In this case, $(x/g_j) \in \beta_\ell^W$ implies that $Arg(\ell)_j = x$. Therefore $Arg(\ell)_i = Arg(\ell)_j = x$. Since $Arg(\ell\alpha_\ell^u) = u$, then $u_i = u_j = c$, which implies $(g_j/c) \in \lambda_u$. Thus $(x/c) \in (\beta_\ell^W \circ \lambda_u)$.

   By Def. 4.1 x is a variable and so it is not a generic constant. Therefore, $(x/c) \notin \lambda_u$. Hence in both cases $(x/c) \in (\beta_\ell^W \circ \lambda_u) - \lambda_u$. This shows that $\alpha_\ell^u \subseteq (\beta_\ell^W \circ \lambda_u) - \lambda_u$.

   Let $(x/c) \in ((\beta_\ell^W \circ \lambda_u) - \lambda_u)$, then there exists i $(1 \leq i \leq n)$ such that $(x/g_i) \in \beta_\ell^W$ and $(g_i/c) \in \lambda_u$. Thus, by Def. 4.1 and definition of λ_u respectively, we have $Arg(\ell)_i = x$ and $u_i = c$. Since $Arg(\ell\alpha_\ell^u) = u$, then $(x/c) \in \alpha_\ell^u$. This shows that $(\beta_\ell^W \circ \lambda_u) - \lambda_u \subseteq \alpha_\ell^u$

2. The proof is similar to (1).

3. None of the generic constants appears in W. Therefore, $W\gamma_\ell = W\delta_\ell^W \circ \eta_u$.

4. The proof is similar to (3).

5. We shall denote the substitution $(\delta_{\ell,\ell'}^W \circ \lambda_u) - \lambda_u$ by A. By the definition of $\delta_{\ell,\ell'}^W$, we have $\delta_\ell^W \subseteq \delta_{\ell'}^W$. Since $UQ(\ell) \subset UQ(\ell')$, then $\delta_\ell^W \subset \delta_{\ell'}^W$. From (2), we have $\gamma_\ell^u = (\delta_\ell^W \circ \lambda_u) - \lambda_u$. Thus, $\gamma_\ell^u \subset A$.

Only if part: Assume that γ^u_ℓ is a subset of $\gamma^u_{\ell'}$. Since $\gamma^u_\ell \subset A$, then $A = \gamma^u_{\ell'}$ if and only if $A - \gamma^u_\ell = \gamma^u_{\ell'} - \gamma^u_\ell$. Now

$$
\begin{array}{rcll}
(x/u_i) \in A - \gamma^u_\ell & \iff & (x/u_i) \in A \text{ and } (x/u_i) \notin \gamma^u_\ell & \\
& \iff & (x/u_i) \in A \text{ and } (x/g_i) \notin \delta^W_\ell & \text{from (2)} \\
& \iff & (x/g_i) \in \delta^W_{\ell,\ell'} \text{ and } (x/g_i) \notin \delta^W_\ell & \\
& \iff & (x/g_i) \in \delta^W_{\ell'} \text{ and } (x/g_i) \notin \delta^W_\ell & \text{from Def. 4.1} \\
& \iff & (x/u_i) \in \delta^W_{\ell'} \circ \lambda_u, \text{ and} & \\
& & (x/u_i) \notin \delta^W_\ell \circ \lambda_u & \text{since } (g_i/u_i) \in \lambda_u \\
& \iff & (x/u_i) \in (\delta^W_{\ell'} \circ \lambda_u) - \lambda_u \text{ and} & \\
& & (x/u_i) \notin (\delta^W_\ell \circ \lambda_u) - \lambda_u & \text{since } (x/u_i) \notin \lambda_u \\
& \iff & (x/u_i) \in \gamma^u_{\ell'} \text{ and } (x/u_i) \notin \gamma^u_\ell & \text{from (2)} \\
& \iff & (x/u_i) \in \gamma^u_{\ell'} - \gamma^u_\ell. &
\end{array}
$$

This shows that if $\gamma^u_\ell \subset \gamma^u_{\ell'}$ then $A = \gamma^u_{\ell'}$.

If part: Assume that $A = \gamma^u_{\ell'}$. As we have just shown, $\gamma^u_\ell \subset A$. Hence, $\gamma^u_\ell \subset \gamma^u_{\ell'}$.

6. We will prove first that the number of elements in $\delta^W_{\ell'} \circ \lambda_u$ is equal to the number of elements in $\delta^W_{\ell,\ell'} \circ \lambda_u$.

$$
\begin{array}{rcll}
|\delta^W_{\ell'} \circ \lambda_u| & = & |\delta^W_{\ell'}| + |\lambda_u| & \\
& = & |UQ(\ell')| + |\lambda_u| & \text{from Def. 4.1} \\
& = & |\delta^W_{\ell,\ell'}| + |\lambda_u| & \text{from Def. 4.1} \\
& = & |\delta^W_{\ell,\ell'} \circ \lambda_u| &
\end{array}
$$

Also

$$
\begin{array}{rll}
W\gamma^u_{\ell'} \text{ is } W(\delta^W_{\ell,\ell'} \circ \lambda_u) & \text{iff} & \ell'\gamma^u_{\ell'} \text{ is } \ell'(\delta^W_{\ell,\ell'} \circ \lambda_u) \\
& \text{iff} & \ell'\delta^W_{\ell'} \circ \lambda_u \text{ is } \ell'(\delta^W_{\ell,\ell'} \circ \lambda_u)
\end{array}
$$

Since $|\delta^W_{\ell'} \circ \lambda_u| = |\delta^W_{\ell,\ell'} \circ \lambda_u|$ then $\delta^W_{\ell'} \circ \lambda_u = \delta^W_{\ell,\ell'} \circ \lambda_u$. For otherwise, there was $x \in UQ(\ell')$ such that $(x/d) \in \delta^W_{\ell,\ell'} \circ \lambda_u$, $(x/c) \in \delta^W_{\ell'} \circ \lambda_u$, and $c \neq d$, which contradicts that $\ell'\delta^W_\ell \circ \lambda_u$ is identical to $\ell'(\delta^W_{\ell,\ell'} \circ \lambda_u)$. Thus

$$
\begin{array}{rlll}
W\gamma^u_{\ell'} \text{ is } W(\delta^W_{\ell,\ell'} \circ \lambda_u) & \text{iff} & \delta^W_{\ell'} \circ \lambda_u = \delta^W_{\ell,\ell'} \circ \lambda_u & \\
& \text{iff} & \delta^W_{\ell'} \circ \lambda_u - \lambda_u = \delta^W_{\ell,\ell'} \circ \lambda_u - \lambda_u & \\
& \text{iff} & \gamma^u_{\ell'} = \delta^W_{\ell,\ell'} \circ \lambda_u - \lambda_u & \text{from (2)} \\
& \text{iff} & \gamma^u_\ell \subset \gamma^u_{\ell'} & \text{from (5)}
\end{array}
$$

◇

4.1 First Pair of Meta Relations: $\mathcal{F}^W_R$ and $\widetilde{\mathcal{F}}^W_R$

We will now define the meta relations $\mathcal{F}^W_R$ and $\widetilde{\mathcal{F}}^W_R$. The tuples in these meta relation are mainly defined by using the substitutions β^W_ℓ, δ^W_ℓ and $\delta^W_{\ell,\ell'}$.

$\mathcal{F}_R^W$

g_1	g_2	c	$W\delta_{\ell_1}^W$	$-$
g_1	g_2	g_3	$W\delta_{\ell_2}^W$	$-$

$\widetilde{\mathcal{F}}_R^W$

g_1	g_2	c	$W\delta_{\ell_1,\ell_2}^W$	$-$

Figure 4.1: The Meta Relations $\mathcal{F}_R^W$ and $\widetilde{\mathcal{F}}_R^W$ of Example 4.1

Definition 4.2 (Meta Relations $\mathcal{F}_R^W$ and $\widetilde{\mathcal{F}}_R^W$)

$$\mathcal{F}_R^W = \bigcup_{e\in\{+,-\}} \{< Arg(\ell\beta_\ell^W), W\delta_\ell^W, e > | \ell \in L_{R,W}^e\}.$$

$$\widetilde{\mathcal{F}}_R^W = \bigcup_{e\in\{+,-\}} \{< Arg(\ell\beta_\ell^W), W\delta_{\ell,\ell'}^W, e > | \ell, \ell' \in L_{R,W}^e \text{ and } UQ(\ell) \subset UQ(\ell')\}.$$

□

In other words, for each (positive or negative) occurrence ℓ of the relation R in the constraint W, there is a tuple t in $\mathcal{F}_R^W$ that has $n+2$ components, where n is the arity of R. These components are defined as follows. The first n components are the arguments of the ground instance $\ell\beta_\ell^W$ of ℓ. These components are, therefore, generic constants in $\{g_1, \ldots, g_n\}$ and/or constants of ℓ. The $(n+1)^{th}$ component is a formula obtained by applying the substitution of δ_ℓ^W to W. The $(n+2)^{th}$ component is either the symbol '+' or '$-$'. It is '+' (resp. '$-$') if ℓ is positive (resp. negative) occurrence of R in W. The tuples of $\widetilde{\mathcal{F}}_R^W$ are defined in the same way as those of $\mathcal{F}_R^W$, but the substitution $\delta_{\ell,\ell'}^W$ is considered instead of δ_ℓ^W.

Remark 4.1 Several remarks can be made to the above definitions:

1. $\delta_\ell^W = \varepsilon$ iff $UQ(\ell) = \emptyset$, i.e, iff every variable x in ℓ is either an existentially quantified variable in W, or it is a universally quantified variable governed by existentially quantified variable.

2. $\mathcal{F}_R^W = \emptyset$ iff $L_{R,W} = \emptyset$, i.e., iff W has no occurrences of R.

3. $\widetilde{\mathcal{F}}_R^W = \emptyset$ iff either $\mathcal{F}_R^W = \emptyset$ or for every ℓ and ℓ' in $L_{R,W}$ neither $UQ(\ell) \subset UQ(\ell')$ nor $UQ(\ell') \subset UQ(\ell)$

Example 4.1 Let W be $\forall x \forall y \forall z(\neg R(x, y, c) \vee \neg R(y, x, z) \vee Q(x, y, z))$. The meta-relations $\mathcal{F}_R^W$ and $\widetilde{\mathcal{F}}_R^W$ for this example are given in Fig. 4.1 and the components of tuples in both of them are given in Table 4.1. □

$W: \quad \forall x \forall y \forall z(\neg R(x,y,c) \vee \neg R(y,x,z) \vee Q(x,y,z))$.

$EQ(W) = \emptyset$
$L^{+}_{R,W} = \emptyset$
$L^{-}_{R,W} = \{\neg R(x,y,c), \neg R(y,x,z)\}$

$\ell_1: \quad \neg R(x,y,c)$

$UQ(\ell_1) = \{x,y\}$.
$\beta^{W}_{\ell_1} = \{x/g_1, y/g_2\}$.
$Arg(\ell_1 \beta^{W}_{\ell_1}) =< g_1, g_2, c >; e = -$
$\delta^{W}_{\ell_1} = \{x/g_1, y/g_2\}$.
$W\delta^{W}_{\ell_1} : \forall z(\neg R(g_1,g_2,c) \vee \neg R(g_2,g_1,z) \vee Q(g_1,g_2,z))$.
$V^{W}_{\ell_1} : \forall z(\neg R(g_2,g_1,z) \vee Q(g_1,g_2,z))$.

$\ell_2: \quad \neg R(y,x,z)$

$UQ(\ell_2) = \{x,y,z\}$.
$\beta^{W}_{\ell_2} = \{y/g_1, x/g_2, z/g_3\}$.
$Arg(\ell_2 \beta^{W}_{\ell_2}) =< g_1, g_2, g_3 >; e = -$
$\delta^{W}_{\ell_2} = \{y/g_1, x/g_2, z/g_3\}$.
$W\delta^{W}_{\ell_2} : (\neg R(g_2,g_1,c) \vee \neg R(g_1,g_2,g_3) \vee Q(g_2,g_1,g_3))$.
$V^{W}_{\ell_2} : (\neg R(g_2,g_1,c) \vee Q(g_2,g_1,g_3))$.

$UQ(\ell_1) \subset UQ(\ell_2)$.

$\delta^{W}_{\ell_1,\ell_2} = \{x/g_1, y/g_2, z/g_3\}$.
$W\delta^{W}_{\ell_1,\ell_2} : (\neg R(g_1,g_2,c) \vee \neg R(g_2,g_1,g_3) \vee Q(g_1,g_2,g_3))$.
$V^{W}_{\ell_1,\ell_2} : (\neg R(g_1,g_2,c) \vee Q(g_1,g_2,g_3))$.

Table 4.1: Components of $\mathcal{F}^{W}_{R}$ and $\widetilde{\mathcal{F}}^{W}_{R}$ of Example 4.1

4.2 The Main Theorem

Given the operation $O(R,u)$ and the constraint W, the central point of deriving the simplified form $C^{e,u}_{R,W}$ is to obtain the set of wffs $\Gamma^{e,u}_{R,W}$. This is made by the first two steps of the algorithm given in Subsection3.2. First we shall illustrate by means of an example that $\Gamma^{e,u}_{R,W}$ can be obtained from $(\mathcal{F}^W_R\lambda_u)$ and $(\tilde{\mathcal{F}}^W_R\lambda_u)$ by using a relational algebra expression. Then we will prove this claim in Theorem 4.1.

$\mathcal{F}^W_R\lambda_u$

a	a	c	$W\delta^W_{\ell_1}\circ\lambda_u$	$-$
a	a	c	$W\delta^W_{\ell_2}\circ\lambda_u$	$-$

$\tilde{\mathcal{F}}^W_R\lambda_u$

a	a	c	$W\delta^W_{\ell_1,\ell_2}\circ\lambda_u$	$-$

Figure 4.2: The Meta Relations $\mathcal{F}^W_R\lambda_u$ and $\tilde{\mathcal{F}}^W_R\lambda_u$ of Example 4.2

Example 4.2 Let W be $\forall x\forall y\forall z(\neg R(x,y,c)\vee\neg R(y,x,z)\vee Q(x,y,z))$. For inserting the tuple $u=<a,a,c>$ into the extension of R, $\Gamma^{-,u}_{R,W}=\{W_1\}$ where

$$W_1:\forall z(\neg R(a,a,c)\vee\neg R(a,a,z)\vee Q(a,a,z)).$$

$\mathcal{F}^W_R$ and $\tilde{\mathcal{F}}^W_R$ for the given constraint are shown in Fig. 4.1. Now for the given tuple u, $\lambda_u=\{g_1/a,g_2/a,g_3/c\}$. Applying λ_u to $\mathcal{F}^W_R$ and $\tilde{\mathcal{F}}^W_R$ yields respectively the two relations $\mathcal{F}^W_R\lambda_u$ and $\tilde{\mathcal{F}}^W_R\lambda_u$ shown in Fig. 4.2 Where:

$$\begin{array}{ll} W\delta^W_{\ell_1}\circ\lambda_u & :\forall z(\neg R(a,a,c)\vee\neg R(a,a,z)\vee Q(a,a,z)).\\ W\delta^W_{\ell_2}\circ\lambda_u & :(\neg R(a,a,c)\vee\neg R(a,a,c)\vee Q(a,a,c)).\\ W\delta^W_{\ell_1,\ell_2}\circ\lambda_u & :(\neg R(a,a,c)\vee\neg R(a,a,c)\vee Q(a,a,c)).\end{array}$$

Let $F(u,-)$ be $(\$1=a)\wedge(\$2=a)\wedge(\$3=c)\wedge(\$5=-)$. Therefore,

$$\begin{array}{ll}\pi_4\sigma_{F(u,e)}(\mathcal{F}^W_R\lambda_u) & =\{W\delta^W_{\ell_1}\circ\lambda_u, W\delta^W_{\ell_2}\circ\lambda_u\}\\ \pi_4\sigma_{F(u,e)}(\tilde{\mathcal{F}}^W_R\lambda_u) & =\{W\delta^W_{\ell_1,\ell_2}\circ\lambda_u\}\end{array}$$

Since W_1 is identical to $W\delta_{\ell_1}\circ\lambda_u$, and $W\delta^W_{\ell_2}\circ\lambda_u$ is identical to $W\delta^W_{\ell_1,\ell_2}\circ\lambda_u$, then

$$\Gamma^{-,u}_{R,W}=\pi_4\sigma_{F(u,e)}(\mathcal{F}^W_R\lambda_u)-\pi_4\sigma_{F(u,e)}(\tilde{\mathcal{F}}^W_R\lambda_u)$$

Note that $Art(R)=3$. □

Theorem 4.1 The main Theorem Given the operation $O(R,u)$ and the constraint W. Let $F(u,e)$ be $\bigwedge^n_{i=1}(\$i=u_i)\wedge(\$(n+1)=e)$ and $\lambda_u=\{(g_1,/u),\ldots,(g_n/u_n)\}$. Then

$$\Gamma^{e,u}_{R,W}=\pi_{n+1}\sigma_{F(u,e)}(\mathcal{F}^W_R\lambda_u)-\pi_{n+1}\sigma_{F(u,e)}(\tilde{\mathcal{F}}^W_R\lambda_u).$$

□

Proof:. Assume that $\gamma^u_\ell\in G^{e,u}_{R,W}$ and for every $\gamma^u_{\ell'}\in G^{e,u}_{R,W}$, $\gamma^u_{\ell'}\not\subset\gamma^u_\ell$. From Def. 3.3, this is equivalent to assuming that $W\gamma^u_\ell\in\Gamma^{e,u}_{R,W}$. First we show that:

$$\gamma^u_\ell\in G^{e,u}_{R,W}\text{ iff }W\gamma^u_\ell\in\pi_{n+1}\sigma_{F(u,e)}(\mathcal{F}^W_R\lambda_u)\tag{4.7}$$

Then, we show that:

$$\text{for every } \gamma^u_{\ell'} \in G^{e,u}_{R,W}, \gamma^u_{\ell'} \not\subset \gamma^u_\ell \text{iff} W\gamma^u_\ell \notin \pi_{n+1}\sigma_{F(u,e)}(\tilde{\mathcal{F}}^W_R \lambda_u) \tag{4.8}$$

Proof of 4.7

$\gamma^u_\ell \in G^{e,u}_{R,W}$

$\iff \alpha^u_\ell \in A^{e,u}_{R,W}$ from Def. 3.2

$\iff Arg(\ell(\beta^W_\ell \circ \lambda_u) - \lambda_u) = u$ from Def. 3.2 and (1) of Lemma 4.1

$\iff Arg(\ell(\beta^W_\ell \circ \lambda_u)) = u, \ell \in L^e_{R,W}$ from (4) of Lemma 4.1

$\iff < Arg(\ell\beta^W_\ell \circ \lambda_u), W\delta^W_\ell \circ \lambda_u, e > \in \sigma_{F(u,e)}(\mathcal{F}^W_R \lambda_u)$ from Def. 4.2

$\iff W\delta^W_\ell \circ \lambda_u \in \pi_{n+1}\sigma_{F(u,e)}(\mathcal{F}^W_R \lambda_u)$

$\iff W\gamma^u_\ell \in \pi_{n+1}\sigma_{F(u,e)}(\mathcal{F}^W_R \lambda_u)$ from (3) of Lemma 4.1

Proof of 4.8

$W\gamma^u_\ell \in \pi_{n+1}\sigma_{F(u,e)}(\tilde{\mathcal{F}}^W_R \lambda_u)$

$\iff < u, W\gamma^u_\ell, e > \in \tilde{\mathcal{F}}^W_R$

$\iff < u, W\gamma^u_\ell, e > = < Arg(\ell'\beta^W_{\ell'} \circ \lambda_u), W\delta^W_{\ell',\ell} \circ \lambda_u, e >$

for some $\ell' \in L^e_{R,W}$ such that $UQ(\ell') \subset UQ(\ell)$ from Def. of $\tilde{\mathcal{F}}^W_R$

$\iff W\gamma^u_\ell$ is identical to $W\delta^W_{\ell',\ell} \circ \lambda_u$, and

$\gamma^u_{\ell'} \in G^{e,u}_{R,W}$ such that $UQ(\ell') \subset UQ(\ell)$

$\iff \gamma^u_{\ell'} \subset \gamma^u_\ell$ for some $\gamma^u_{\ell'} \in G^{e,u}_{R,W}$ from (6) Lemma 4.1

◇

In obtaining the set $\Gamma^{e,u}_{R,W}$ as in Theorem 4.1, we do not need to define the set of substitutions $A^{e,u}_{R,W}$ and $G^{e,u}_{R,W}$, but we need only to define λ_u and $F(u,e)$. The advantage of obtaining $\Gamma^{e,u}_{R,W}$ as in Theorem 4.1 is that the definitions of λ_u and $F(u,e)$ are easier than those of $A^{e,u}_{R,W}$ and $G^{e,u}_{R,W}$. Also, these definitions do not require analyzing the quantifier structure of W which were among the disadvantages of the simplification method.

Chapter 5

Simplifying the Formulas of Meta Relations

Now we present the second stage towards defining the meta relations $\mathcal{T}_R^W$ and $\widetilde{\mathcal{T}}_R^W$. In this stage we do the following. First, for each $W\delta_\ell^W$ appearing in $\mathcal{F}_R^W$, we define a simplified form, denoted by V_ℓ^W, such that if $W\delta_\ell^W \circ \lambda_u \in \Gamma_{R,W}^{e,u}$ then $V_\ell^W \lambda_u$ is equivalent to $W \circ \lambda_u$, and either it does not contain pre-valued literals or it contains a small number of them compared to the number of pre-valued literals that appear in $W\delta_\ell^W \circ \lambda_u$. Second, we define two meta relations $\mathcal{S}_R^W$ and $\widetilde{\mathcal{S}}_R^W$ in the same way as the meta relations $\mathcal{F}_R^W$ and $\widetilde{\mathcal{F}}_R^W$ were defined but this time we consider the simplified forms V_ℓ^W instead of the instances $W\delta_\ell^W$.

5.1 Simplifying Formulas of Meta Relation $\mathcal{F}_R^W$ and $\widetilde{\mathcal{F}}_R^W$

We start by the following definition in which we formally state how simplified forms for formulas appearing in $\mathcal{F}_R^W$ and in $\widetilde{\mathcal{F}}_R^W$ can be obtained.

Definition 5.1 (wff V_ℓ^W)
Let $\ell \in L_{R,W}^e$. V_ℓ^W is a wff defined as follows:

- if $\delta_\ell^W \neq \beta_\ell^W$ then V_ℓ^W is $W\delta_\ell^W$; and
- if $\delta_\ell^W = \beta_\ell^W$ then V_ℓ^W is the wff derived from $W\delta_\ell^W$ by replacing in $W\delta_\ell^W$ each occurrence of $\ell\delta_\ell^W$ by $\boldsymbol{F}$ and applying the absorption rules as much as possible.

□

Definition 5.2 (wff $V_{\ell,\ell'}^W$)
Let $\ell, \ell' \in L_{R,W}^e$. $V_{\ell,\ell'}^W$ is a wff defined as follows:

- if $\delta_{\ell'}^W \neq \beta_{\ell'}^W$ then $V_{\ell,\ell'}^W = W\delta_{\ell,\ell'}^W$; and
- if $\delta_{\ell'}^W = \beta_{\ell'}^W$ then $V_{\ell,\ell'}^W$ is the wff derived from $W\delta_{\ell,\ell'}^W$ by replacing in $W\delta_{\ell,\ell'}^W$ each occurrence of $\ell'\delta_{\ell,\ell'}^W$ by $\boldsymbol{F}$ and applying the absorption rules as much as possible.

$\square$

Informally, if δ_ℓ^W is not equal to β_ℓ^W, which means that at least one of the arguments of $\ell\delta_\ell^W$ is a variable, then V_ℓ^W is $W\delta_\ell^W$; and if δ_ℓ^W is equal to β_ℓ^W, which means that none of the arguments of $\ell\delta_\ell^W$ is a variable, then V_ℓ^W is obtained from $W\delta_\ell^W$ by deleting all occurrences of $\ell\delta_\ell^W$ in $W\delta_\ell^W$, such that the obtained expression is a wff. The definition of $V_{\ell,\ell'}^W$ is very similar to that of V_ℓ^W.
The reader can notice that the simplified form V_ℓ^W is derived from $W\delta_\ell^W$ in a process analogous to simplifying instances of W in $\Gamma_{R,W}^{e,u}$ in the third step of the simplification method. But here we use the generic constants instead of specific update values. The next lemma validates the simplified form V_ℓ^W.

Lemma 5.1 Let $W\delta_\ell^W \circ \lambda_u \in \Gamma_{R,W}^{e,u}$. Then in the new state S'

$$V_\ell^W \lambda_u \longleftrightarrow W\delta_\ell^W \circ \lambda_u.$$

$\square$

Proof:. We will prove the lemma for the case in which the operation $O(R,u)$ is an inserting operation, the proof for a deleting operation is very similar. Therefore $e = -$, and in the new state S', $\neg R(u)$ is equivalent to $\boldsymbol{F}$. Let $W\delta_\ell^W \circ \lambda_u \in \Gamma_{R,W}^{-,u}$. Then from Theorem 4.1, $Arg(\ell\beta_\ell^W \circ \lambda_u) = u$ and $\ell \in L_{R,W}^-$. This means that $\ell\beta_\ell^W \circ \lambda_u$ is $\neg R(u)$. Therefore, in the new state S', $\ell\beta_\ell^W \circ \lambda_u$ is equivalent to $\boldsymbol{F}$.
If $\delta_\ell^W \neq \beta_\ell^W$ then V_ℓ^W is $W\delta_\ell^W$ and thus there is nothing to prove. Assume that $\delta_\ell^W = \beta_\ell^W$ then $\ell\delta_\ell^W \circ \lambda_u$ is identical to $\ell\beta_\ell^W \circ \lambda_u$. Therefore, $\ell\delta_\ell^W \circ \lambda_u$ is identical to $\neg R(u)$ and hence it is equivalent to $\boldsymbol{F}$.
Now, let V be a wff deriverd from $W\delta_\ell^W \circ \lambda_u$ by replacing in $W\delta_\ell^W \circ \lambda_u$ each occurrence of $\ell\delta_\ell^W \circ \lambda_u$ by $\boldsymbol{F}$. and applying the absorption rules as much as possible. In the new state S', V is equivalent to $W\delta_\ell^W \circ \lambda_u$. From the definition of V_ℓ^W, V is identical to $V_\ell^W \lambda_u$. Thus, in the new state S', $V_\ell^W \lambda_u$ is equivalent to $W\delta_\ell^W \circ \lambda_u$

$\diamond$

As a consequence of Lemma 5.1 we have the following corollary.

Corollary 5.1 Let $\Gamma_{R,W}^{e,u} \neq \emptyset$. and $\Delta_{R,W}^{e,u} = \{V_\ell^W \lambda_u | W\delta_\ell^W \circ \lambda_u \in \Gamma_{R,W}^{e,u}\}$. Then

$$S' \in mod(\Gamma_{R,W}^{e,u}) \text{ iff } S' \in mod(\Delta_{R,W}^{e,u}).$$

$\square$

$\mathcal{S}_R^W$				
g_1	g_2	c	$V_{\ell_1}^W$	$-$
g_1	g_2	g_3	$V_{\ell_2}^W$	$-$

$\widetilde{\mathcal{S}}_R^W$				
g_1	g_2	c	V_{ℓ_1,ℓ_2}^W	$-$

Figure 5.1: Meta Relations $\mathcal{S}_R^W$ and $\widetilde{\mathcal{S}}_R^W$ of Example 5.1

5.2 Meta Relations $\mathcal{S}_R^W$ and $\widetilde{\mathcal{S}}_R^W$: Motivation

According to the steps of the algorithm given in Subsection 3.2, $C_{R,W}^{e,u}$ is derived from applying the steps 3 – 4 to the conjunction of formulas in $\Gamma_{R,W}^{e,u}$. From the definition of $\Gamma_{R,W}^{e,u}$ and (2) of Lemma 5.1, every $W\gamma_\ell^u \in \Gamma_{R,W}^{e,u}$ can be written as $W\delta_\ell^W \circ \lambda_u$. According to the way the formula V_ℓ^W is defined, $V_\ell^W \lambda_u$ can be seen as if it were derived by eliminating the pre-valued literals $R(u)$ and $\neg R(u)$ from $W\delta_\ell^W \circ \lambda_u$ in the third step of the algorithm. Thus the simplified form $C_{R,W}^{e,u}$ can be derived by applying the steps 3 – 4 of the algorithm to the conjunction the formulas in $\Delta_{R,W}^{e,u}$. As stated by Nicolas in his discussion of the simplification method (see page 249 of [Nic82]), the elimination of the pre-valued literals from the instances in $\Gamma_{R,W}^{e,u}$ disadvantages the simplified form $C_{R,W}^{e,u}$. This is done by the third step of the algorithm. According to the way the formula $\Delta_{R,W}^{e,u}$ is defined, the number of pre-valued literals that appear in the conjunction of formulas of $\Delta_{R,W}^{e,u}$ is less than or equal to the number of pre-valued literals that appear in the conjunction of formulas of $\Gamma_{R,W}^{e,u}$. This means that, obtaining $C_{R,W}^{e,u}$ from $\Delta_{R,W}^{e,u}$ is more desirable than of obtaining it from $\Gamma_{R,W}^{e,u}$.

5.3 Meta Relations $\mathcal{S}_R^W$ and $\widetilde{\mathcal{S}}_R^W$: Definition

The above motivates the definition of $\mathcal{S}_R^W$ and $\widetilde{\mathcal{S}}_R^W$. In fact $\mathcal{S}_R^W$ and $\widetilde{\mathcal{S}}_R^W$ are defined such that

$$\Delta_{R,W}^{e,u} = \pi_{n+1}\sigma_{F(u,e)}(\mathcal{S}_R^W \lambda_u) - \pi_{n+1}\sigma_{F(u,e)}(\widetilde{\mathcal{S}}_R^W \lambda_u).$$

Definition 5.3 (Meta Relations $\mathcal{S}_R^W$ and $\widetilde{\mathcal{S}}_R^W$)
$\mathcal{S}_R^W$ and $\widetilde{\mathcal{S}}_R^W$ are two meta relations obtained from $\mathcal{F}_R^W$ and $\widetilde{\mathcal{F}}_R^W$ respectively by using definitions 5.1 and 5.2 as follows.

$$\mathcal{S}_R^W = \{< Arg(\ell\beta_\ell^W), V_\ell^W, e > \mid < Arg(\ell\beta_\ell^W), W\delta_\ell^W, e > \in \mathcal{F}_R^W\}.$$

$$\widetilde{\mathcal{S}}_R^W = \{< Arg(\ell\beta_\ell^W), V_{\ell,\ell'}^W, e > \mid\ < Arg(\ell\beta_\ell^W), W\delta_{\ell,\ell'}^W, e > \in \widetilde{\mathcal{F}}_R^W\}.$$

□

Example 5.1 Let W be $\forall x\forall y\forall z(\neg R(x,y,c) \vee \neg R(y,x,z) \vee Q(x,y,z))$, as in Example 4.1. The meta relations $\mathcal{F}_R^W$ and $\widetilde{\mathcal{F}}_R^W$ for this constraint are shown in Fig. 4.1. From

Table 4.1 we have

$$\begin{array}{ll} V_{\ell_1}^W & : \forall z(\neg R(g_2, g_1, z) \vee Q(g_1, g_2, z)). \\ V_{\ell_2}^W & : \neg R(g_2, g_1, c) \vee Q(g_2, g_1, g_3)). \\ V_{\ell_1,\ell_2}^W & : (\neg R(g_1, g_2, c) \vee Q(g_1, g_2, g_3)). \end{array}$$

Thus replacing $W\delta_{\ell_1}^W$, $W\delta_{\ell_2}^W$, and $W\delta_{\ell_1,\ell_2}^W$ by $V_{\ell_1}^W$, $V_{\ell_2}^W$ and V_{ℓ_1,ℓ_2}^W, respectively, we get the meta-relations $\mathcal{S}_R^W$ and $\tilde{\mathcal{S}}_R^W$ shown in Figure 3.3.

$\mathcal{S}_R^W \lambda_u$

a	a	c	$V_{\ell_1}^W \circ \lambda_u$	$-$
a	a	c	$V_{\ell_2}^W \circ \lambda_u$	$-$

$\tilde{\mathcal{S}}_R^W \lambda_u$

a	a	c	$V_{\ell_1,\ell_2}^W \circ \lambda_u$	$-$

Figure 5.2: The meta-relations $\mathcal{S}_R^W \lambda_u$ and $\tilde{\mathcal{S}}_R^W \lambda_u$ of Example 4.1

Suppose that the tuple $u =< a, a, c >$ is inserted into the extension of R. Then $\lambda_u = \{g_1/a, g_2/a, g_3/c\}$ and $F(u, -)$ is $(\$1 = a) \wedge (\$2 = a) \wedge (\$3 = c) \wedge (\$5 = -)$. Applying λ_u to $\mathcal{S}_R^W$ and $\tilde{\mathcal{S}}_R^W$ we get the two meta-relations shown in Figure 3.4, where

$$\begin{array}{ll} V_{\ell_1}^W \circ \lambda_u : & \forall z(\neg R(a, a, z) \vee Q(a, a, z)). \\ V_{\ell_2}^W \circ \lambda_u : & \neg R(a, a, c) \vee Q(a, a, c)). \\ V_{\ell_1,\ell_2}^W \circ \lambda_u : & (\neg R(a, a, c) \vee Q(a, a, c)). \end{array}$$

By evaluating the expression $\pi_4 \sigma_{F(u,-)}(\mathcal{S}_R^W \lambda_u) - \pi_4 \sigma_{F(u,-)}(\tilde{\mathcal{S}}_R^W \lambda_u)$, we get the formula $V_{\ell_1}^W \lambda_u$ which is identical to the simplified form $C_{R,W}^{-,u}$ □

We conclude this section by the next theorem which states that for the operation $O(R, u)$, the set of wffs that result from performing the expression $\pi_{n+1}\sigma_{F(u,e)}(\mathcal{S}_R^W \lambda_u) - \pi_{n+1}\sigma_{F(u,e)}(\tilde{\mathcal{S}}_R^W \lambda_u)$ is $\Delta_{R,W}^{e,u}$; and that the evaluation of the formulas in $\Delta_{R,W}^{e,u}$ is sufficient for determining whether W is satisfied in the new state S' or not.

Theorem 5.1

$$\Delta_{R,W}^{e,u} = \pi_{n+1}\sigma_{F(u,e)}(\mathcal{S}_R^W \lambda_u) - \pi_{n+1}\sigma_{F(u,e)}(\tilde{\mathcal{S}}_R^W \lambda_u).$$

$$\text{If } \Delta_{R,W}^{e,u} \neq \emptyset \text{ then } S' \in mod(W) \text{ iff } S' \in mod(\Delta_{R,W}^{e,u}).$$

□

Proof:.

1.

$$\begin{array}{llll} V_\ell^W \lambda_u \in \Delta_{R,W}^{e,u} & \Longleftrightarrow & W\delta_\ell^W \circ \lambda_u \in \Gamma_{R,W}^{e,u} & \text{from Def. of } \Delta_{R,W}^{e,u} \\ & \Longleftrightarrow & W\delta_\ell^W \circ \lambda_u \in \pi_{n+1}\sigma_{F(u,e)}(\mathcal{F}_R^W) \text{ and} & \\ & & W\delta_\ell^W \circ \lambda_u \notin \pi_{n+1}\sigma_{F(u,e)}(\tilde{\mathcal{F}}_R^W) & \text{from Theorem 4.1} \\ & \Longleftrightarrow & < u, W\delta_\ell^W \circ \lambda_u, e > \in (\mathcal{F}_R^W) \text{ and} & \\ & & < u, W\delta_\ell^W \circ \lambda_u, e > \notin (\tilde{\mathcal{F}}_R^W) & \\ & \Longleftrightarrow & < u, V_\ell^W \lambda_u, e > \in (\mathcal{S}_R^W) \text{ and} & \\ & & < u, V_\ell^W \lambda_u, e > \notin (\tilde{\mathcal{S}}_R^W) & \text{from Def. 5.3} \\ & \Longleftrightarrow & V_\ell^W \lambda_u \in \pi_{n+1}\sigma_{F(u,e)}(\mathcal{S}_R^W) \text{ and} & \\ & & V_\ell^W \lambda_u \notin \pi_{n+1}\sigma_{F(u,e)}(\tilde{\mathcal{S}}_R^W) & \end{array}$$

2.
Let $\Delta^{e,u}_{R,W} \neq \emptyset$ then $\Gamma^{e,u}_{R,W} \neq \emptyset$. From the validation of the simplification method, we have, $S' \in mod(W)$ iff $S' \in mod(\Gamma^{e,u}_{R,W})$. Also, from Corollary 5.1 we have $S' \in mod(\Delta^{e,u}_{R,W})$ iff $S' \in mod(\Gamma^{e,u}_{R,W})$. Hence $S' \in mod(W)$ iff $S' \in mod(\Delta^{e,u}_{R,W})$

$\diamond$

In conclusion, we have shown that the simplified form $C^{e,u}_{R,W}$ can be derived by applying Steps 3-4 of the simplification method to the conjunction of the formulas of $\Delta^{e,u}_{R,W}$.

Chapter 6

Removing Redundancy

In Chapter 5, we have shown that the simplified form $C^{e,u}_{R,W}$ can be derived by applying Steps 3-4 of the simplification method to the conjunction of the formulas of $\Delta^{e,u}_{R,W}$. In this chapter we will show that by deleting some redunadent tuples from meta relations $\mathcal{S}^W_R$ and $\widetilde{\mathcal{S}}^W_R$ we can obatin an equivalent pairs of meta relations $\mathcal{T}^W_R$ and $\widetilde{\mathcal{T}}^W_R$. Using $\mathcal{T}^W_R$ and $\widetilde{\mathcal{T}}^W_R$ we can obtain $C^{e,u}_{R,W}$ by just only applying the third step of the simplification method.

6.1 Meta Relations $\mathcal{T}^W_R$ and $\widetilde{\mathcal{T}}^W_R$: Motivation

Let $\Delta^{e,u}_{R,W} \neq \emptyset$, then $\Delta^{e,u}_{R,W}$ can divide into subsets $\Delta_1, \ldots, \Delta_k$ such that

- for every i and j, $(1 \leq i, j \leq k)$, $\Delta_i \cap \Delta_j = \emptyset$; and
- all formulas in Δ_i, $(1 \leq i \leq k)$, are identical up to the permutation of the disjunctions, a permutation of literals and a renaming of variables.

Let Ω be any subset of $\Delta^{e,u}_{R,W}$ such that $|\Omega| = k$. Then

1. for every $W' \in \Omega$, there exists one and only one Δ_i, $(1 \leq i \leq n)$, such that $W' \in \Delta_i$. This means that no two formulas in Ω are identical up to the permutation of the disjunctions, a permutation of literals and a renaming of variables; and
2. for every $V \in \Delta^{e,u}_{R,W}$ either $V \in \Omega$ or there exists $V' \in \Omega$ such that V and V' are identical up to the permutation of the disjunctions, a permutation of literals and a renaming of variables, thus $S' \in mod(\Delta^{e,u}_{R,W})$ iff $S' \in mod(\Omega)$.

Let W_Δ and W_Ω be the conjunction of wffs of $\Delta^{e,u}_{R,W}$ and Ω respectively. From 1 and 2, W_Ω can be seen as if it were derived by applying Step 4 of the simplification method to W_Δ. The order of Step 3 and Step 4 of simplification method is immaterial, i.e., $C^{e,u}_{R,W}$ can also be derived by applying Step 4 first and then Step 3 to W_Δ. Thus $C^{e,u}_{R,W}$ can be derived by applying only Step 3 of the simplification method to W_Ω.

This motivates the definition of $\mathcal{T}_R^W$ and $\widetilde{\mathcal{T}}_R^W$. In fact $\mathcal{T}_R^W$ and $\widetilde{\mathcal{T}}_R^W$ will be defined such that

$$\Omega = \pi_{n+1}\sigma_{F(u,e)}(\mathcal{T}_R^W \lambda_u) - \pi_{n+1}\sigma_{F(u,e)}(\widetilde{\mathcal{T}}_R^W \lambda_u).$$

6.2 Meta Relations $\mathcal{T}_R^W$ and $\widetilde{\mathcal{T}}_R^W$: Definition

$\mathcal{T}_R^W$ and $\widetilde{\mathcal{T}}_R^W$ are defined by deleting some tuples from $\mathcal{S}_R^W$ and $\widetilde{\mathcal{S}}_R^W$ respectively. The following definition characterizes these tuples, in which we define a binary relation, denoted $\sim$, on the tuples of $\mathcal{S}_R^W$.

Definition 6.1 (Binary Relation $\sim$) Let $t, t' \in \mathcal{S}_R^W$. t and t' are said to be similar, denoted $t \sim t'$, iff $< t_1, \ldots, t_n, t_{n+2} >$ is equal to $< t'_1, \ldots, t'_n, t'_{n+2} >$ and t_{n+1}, t'_{n+1} are identical up to a permutation of the disjunctions, a permutation of the atomic formulas and a renaming of variables. □

It is clear that $\sim$ is an equivalence relation on $\mathcal{S}_R^W$. The next definition defines the meta relations $\mathcal{T}_R^W$ and $\widetilde{\mathcal{T}}_R^W$.

Definition 6.2 (Meta Relations $\mathcal{T}_R^W$ and $\widetilde{\mathcal{T}}_R^W$)
Let m be the number of equivalence classes generated by $\sim$ on $\mathcal{S}_R^W$.

- $\mathcal{T}_R^W$ is a subset of $\mathcal{S}_R^W$ such that $|\mathcal{T}_R^W| = m$, and for every two distinct tuples t and t' in $\mathcal{T}_R^W$, $t \not\sim t'$.
- $\widetilde{\mathcal{T}}_R^W$ is a subset of $\widetilde{\mathcal{S}}_R^W$ such that $< Arg(\ell\beta_\ell^W), V_{\ell,\ell'}^W, e > \in \widetilde{\mathcal{T}}_R^W$ iff
  - $< Arg(\ell\beta_\ell^W), V_\ell^W, e > \in \mathcal{S}_R^W$,
  - $< Arg(\ell\beta_\ell^W), {V'}_\ell^W, e > \in \mathcal{S}_R^W$, and
  - $< Arg(\ell\beta_\ell^W), V_{\ell,\ell'}^W, e > \in \widetilde{\mathcal{S}}_R^W$.

□

In other words, if there are m equivalence classes for the relation $\sim$ on $\mathcal{S}_R^W$, then $\mathcal{T}_R^W$ is any subset of $\mathcal{S}_R^W$ that has one and only one tuple from each equivalence class. Notice that if $m > 1$, then there are at least two subsets of $\mathcal{S}_R^W$ that satisfy the two conditions stated for the definition of $\mathcal{T}_R^W$; in this case one of these subset may be taken as meta relation $\mathcal{T}_R^W$.

Example 6.1 Let W be $\forall x \forall y \forall z(\neg R(x,y) \vee \neg R(x,z) \vee y = z)$. The meta-relations $\mathcal{F}_R^W$, $\mathcal{S}_R^W$ and $\mathcal{T}_R^W$ for the given constraint W and relation R are shown in Fig. 6.1. The components of tuples in each of these meta-relations are given in Table 6.1. Note that $\widetilde{\mathcal{F}}_R^W = \emptyset$ because neither $UQ(\ell_1) \subset UQ(\ell_2)$ nor $UQ(\ell_2) \subset UQ(\ell_1)$. Thus $\widetilde{\mathcal{S}}_R^W$ and $\widetilde{\mathcal{T}}_R^W$ are both empty.

Suppose that the tuple $u =< a, b >$ is inserted into the extension of the relation R. Then $\lambda_u = \{g_1/a, g_2/b\}$, and $F(u, -)$ is $(\$1 = a) \wedge (\$2 = b) \wedge (\$4 = -)$. Applying λ_u to $\mathcal{T}_R^W$ yields the relation $\mathcal{T}_R^W \lambda_u$ shown in (d) of Figure 3.4 where:

$$V_{\ell_1}^W \lambda_u : \forall z(\neg R(a, z) \vee b = z).$$

This wff results from evaluating $\pi_3 \sigma_{F(u,-)}(\mathcal{T}_R^W \lambda_u)$. According to the algorithm of the simplification method the simplified form, $C_{R,W}^{-,u}$, for this example is one and only one of the following wffs:

$$W_1' : \forall z(\neg R(a, z) \vee b = z).$$
$$W_2' : \forall y(\neg R(a, y) \vee y = b).$$

Since $V_{\ell_1}^W \lambda_u$ is identical to W_1', then

$$\pi_3 \sigma_{F(u,-)}(\mathcal{T}_R^W \lambda_u) = \{C_{R,W}^{-,u}\}.$$

$\mathcal{F}_R^W \lambda_u$

g_1	g_2	$W\delta_{\ell_1}^W$	$-$
g_1	g_2	$W\delta_{\ell_2}^W$	$-$

$\mathcal{S}_R^W$

g_1	g_2	$V_{\ell_1}^W$	$-$
g_1	g_2	$V_{\ell_2}^W$	$-$

$\widetilde{\mathcal{T}}_R^W$

g_1	g_2	V_{ℓ_1,ℓ_2}^W	$-$

$\widetilde{\mathcal{T}}_R^W \lambda_u$

a	b	$V_{\ell_1,\ell_2}^W \circ \lambda_u$	$-$

Figure 6.1: Meta Relations $\mathcal{F}_R^W$, $\mathcal{S}_R^W$, $\mathcal{T}_R^W \lambda_u$ and $\widetilde{\mathcal{T}}_R^W \lambda_u$ of Example 6.1

□

6.3 Validating Derivation of $C_{R,W}^{e,u}$ from Meta Relations

Now we will prove that $C_{R,W}^{e,u}$ is the formula obtained by only applying the third step of the simplification method to the conjunction of the formulas result from performing the expression:

$$\pi_{n+1} \sigma_{F(u,e)}(\mathcal{T}_R^W \lambda_u) - \pi_{n+1} \sigma_{F(u,e)}(\widetilde{\mathcal{T}}_R^W \lambda_u).$$

Theorem 6.1 Let $\Delta_{R,W}^{e,u} \neq \emptyset$. Let $\Delta_1, \ldots, \Delta_k$ be subsets of $\Delta_{R,W}^{e,u}$ such that

- $\Delta_{R,W}^{e,u} = \bigcup_{i=1}^k \Delta_i$;
- for every i and j $(1 \leq i, j \leq k)$ $\Delta_i \cap \Delta_j = \emptyset$; and
- all formulas in Δ_i, $(1 \leq i \leq k)$, are identical up to the permutation of the disjunctions, a permutation of literals and a renaming of variables.

Let $\Omega_{R,W}^{e,u} = \pi_{n+1}\sigma_{F(u,e)}(\mathcal{T}_R^W\lambda_u) - \pi_{n+1}\sigma_{F(u,e)}(\widetilde{\mathcal{T}}_R^W\lambda_u)$. Then

$$\Omega_{R,W}^{e,u} \subseteq \Delta_{R,W}^{e,u}.$$

$$|\Omega_{R,W}^{e,u}| = k.$$

$$S' \in mod(\Delta_{R,W}^{e,u}) \text{ iff } S' \in mod(\Omega_{R,W}^{e,u}).$$

□

Proof:.

1. From Def. 6.2 $\mathcal{T}_R^W \subseteq \mathcal{S}_R^W$ and $\widetilde{\mathcal{T}}_R^W \subseteq \widetilde{\mathcal{S}}_R^W$. Therefore $\mathcal{T}_R^W\lambda_u \subseteq \mathcal{S}_R^W\lambda_u$ and $\widetilde{\mathcal{T}}_R^W\lambda_u \subseteq \widetilde{\mathcal{S}}_R^W\lambda_u$. Thus

$$\begin{aligned} \pi_{n+1}\sigma_{F(u,e)}(\mathcal{T}_R^W\lambda_u) &\subseteq \pi_{n+1}\sigma_{F(u,e)}(\mathcal{S}_R^W\lambda_u) \\ \pi_{n+1}\sigma_{F(u,e)}(\widetilde{\mathcal{T}}_R^W\lambda_u) &\subseteq \pi_{n+1}\sigma_{F(u,e)}(\widetilde{\mathcal{S}}_R^W\lambda_u) \end{aligned}$$

   Hence

$$\begin{aligned} \Omega_{R,W}^{e,u} &\subseteq \pi_{n+1}\sigma_{F(u,e)}(\mathcal{S}_R^W\lambda_u) - \pi_{n+1}\sigma_{F(u,e)}(\widetilde{\mathcal{S}}_R^W\lambda_u) \\ &= \Delta_{R,W}^{e,u}. \end{aligned}$$

2. From the definition of $\mathcal{T}_R^W$, for every tuple $t \in \mathcal{S}_R^W$, there is $t' \in \mathcal{T}_R^W$ such that $t \sim t'$. Therefore for every $< u, V_\ell^W\lambda_u, e > \in \sigma_{F(u,e)}(\mathcal{S}_R^W\lambda_u)$ there is $< u, V'^W_\ell\lambda_u, e >$ in $\sigma_{F(u,e)}(\mathcal{S}_R^W\lambda_u)$ such that $V_\ell^W\lambda_u$ and $V'^W_\ell\lambda_u$ are identical up to the permutation of the disjunctions, a permutation of literals and a renaming of variables. Thus for every $V_1 \in \Delta_{R,W}^{e,u}$, there is $V_2 \in \Omega_{R,W}^{e,u}$ such that V_1 and V_2 are identical up to the permutation of the disjunctions, a permutation of literals and a renaming of variables.
   Suppose that for some i, $(1 \leq i \leq k)$, $\Omega_{R,W}^{e,u} \cap \Delta_i = \emptyset$. Let $V \in \Delta_i$, then $V \in \Delta_{R,W}^{e,u}$. Then there is $V' \in \Omega_{R,W}^{e,u}$ such that V and V' are identical up to the permutation of the disjunctions, a permutation of literals and a renaming of variables. From definition of Δ_i, this implies that $V' \in \Delta_i$, which contradicts that $\Omega_{R,W}^{e,u} \cap \Delta_i = \emptyset$. Thus for every i, $(1 \leq i \leq k)$, $\Omega_{R,W}^{e,u} \cap \Delta_i \neq \emptyset$.
   From the definition of $\mathcal{T}_R^W$, for every two tuples $t, t' \in \mathcal{T}_R^W$, $t \not\sim t'$. Therefore for every two wffs $V, V' \in \Omega_{R,W}^{e,u}$, V and V' are not identical up to the permutation of the disjunctions, a permutation of literals and a renaming of variable. Thus for every i, $(1 \leq i \leq k)$ $|\Delta_i \cap \Omega_{R,W}^{e,u}| = 1$. Hence the number of wffs of $\Omega_{R,W}^{e,u}$ is k.

3. Let $\Delta_{R,W}^{e,u} \neq \emptyset$ then $\Omega_{R,W}^{e,u} \neq \emptyset$.

   Only if part: Assume that $S' \in mod(\Delta_{R,W}^{e,u})$. From (1), $\Omega_{R,W}^{e,u} \subseteq \Delta_{R,W}^{e,u}$. Thus $S' \in mod(\Omega_{R,W}^{e,u})$.

If part: As we have just shown in the proof of (2), for every $V \in \Delta^{e,u}_{R,W} - \Omega^{e,u}_{R,W}$, there is $V' \in \Omega^{e,u}_{R,W}$ such that V and V' are identical up to the permutation of the disjunctions, a permutation of literals and a renaming of variables. This implies that for every $V \in \Delta^{e,u}_{R,W} - \Omega^{e,u}_{R,W}$, there is $V' \in \Omega^{e,u}_{R,W}$ such that V and V' are equivalent. Thus if $S' \in mod(\Omega^{e,u}_{R,W})$ then $S' \in mod(\Delta^{e,u}_{R,W} - \Omega^{e,u}_{R,W})$ and hence $S' \in mod(\Delta^{e,u}_{R,W})$

◇

From Theorem 6.1 and the first paragraph of this subsection, it follows that $C^{e,u}_{R,W}$ is the formula obtained by only applying the third step of the simplification method to the conjunction of the formulas result from performing the expression:

$$\pi_{n+1}\sigma_{F(u,e)}(\mathcal{T}^{W}_{R}\lambda_u) - \pi_{n+1}\sigma_{F(u,e)}(\tilde{\mathcal{T}}^{W}_{R}\lambda_u).$$

W : $\forall x \forall y \forall z(\neg R(x,y) \vee \neg R(x,z) \vee y = z)$.

$EQ(W) = \emptyset$
$L^{+}_{R,W} = \emptyset$
$L^{-}_{R,W} = \{\neg R(x,y), \neg R(x,z)\}$

ℓ_1 : $\neg R(x,y)$

$UQ(\ell_1) = \{x, y\}$.
$\beta^W_{\ell_1} = \{x/g_1, y/g_2\}$.
$Arg(\ell_1 \beta^W_{\ell_1}) =< g_1, g_2 >; e = -$
$\delta^W_{\ell_1} = \{x/g_1, y/g_2\}$.
$W\delta^W_{\ell_1} : \forall z(\neg R(g_1, g_2) \vee \neg R(g_1, z) \vee g_2 = z))$.
$V^W_{\ell_1} : \forall z(\neg R(g_1, z) \vee g_2 = z)$.

ℓ_2 : $\neg R(x,z)$

$UQ(\ell_1) = \{x, z\}$.
$\beta^W_{\ell_2} = \{x/g_1, y/g_2\}$.
$Arg(\ell_2 \beta^W_{\ell_1}) =< g_1, g_2 >; e = `-'$.
$\delta^W_{\ell_2} = \{x/g_1, y/g_2\}$.
$W\delta^W_{\ell_2} : \forall y(\neg R(g_1, y) \vee \neg R(g_1, g_2) \vee y = g_2)$.
$V^W_{\ell_2} : \forall y(\neg R(g_1, y) \vee y = g_2)$.

Table 6.1: Components of tuples of $\mathcal{F}^W_R$ and $\widetilde{\mathcal{S}}^W_R$ of Example 6.1

Chapter 7

Transactions and Meta Relations

In this chapter we will generlaize results obtained in the Chapter 6 by considering transactions rather than atomic operations.

7.1 The Main Theorem: Transactions

In the following theorem we will use the results obtained on the meta-relations $\mathcal{T}_R^W$ and $\tilde{\mathcal{T}}_R^W$ in Chapter 6, to prove the following:

- if for every operation $O(R,u)$ in the transaction Tr, $\sigma_{F(u,e)}\mathcal{T}_R^W\lambda_u = \emptyset$, then W is unaffected by operations in Tr, i.e., it will remain satisfied in the new state S'; and
- if for some operation $O(R,u)$ in the transaction Tr, $\sigma_{F(u,e)}\mathcal{T}_R^W \neq \emptyset$, then W may be affected by the operations of Tr, and in this case we can obtain a subset of $\cup_{O(R,u)\in Tr}\Omega_{R,W}^{e,u}$ that is sufficient to evaluate W in the new state S'.

Theorem 7.1 If $\bigcup_{O(R,u)\in Tr}\sigma_{F(u,e)}\mathcal{T}_R^W\lambda_u = \emptyset$ then $S' \in mod(W)$, otherwise

$$S' \in mod(W) \text{ iff } S' \in mod(\Omega_W^*).$$

Where

$$\Omega_W^* = \begin{cases} \{W\} & \text{if } W \in \Omega_W^\star \\ \Omega_W^\star & \text{otherwise} \end{cases} \quad \text{and } \Omega_W^\star = \bigcup_{O(R,u)\in Tr} \Omega_{R,W}^{e,u}.$$

□

Proof:. Let $\bigcup_{O(R,u)\in Tr}\sigma_{F(u,e)}\mathcal{T}_R^W\lambda_u = \emptyset$. Then $\bigcup_{O(R,u)\in Tr}\sigma_{F(u,e)}\mathcal{F}_R^W\lambda_u$ is $\emptyset$. By Theorem 4.1, this implies that $\Gamma_{R,W}^{e,u} = \emptyset$ for every $O(R,u) \in Tr$. Thus by Theorem 3.1, $\Gamma_W^\star = \emptyset$, and hence, $S' \in mod(W)$.

Let $\bigcup_{O(R,u)\in Tr} \sigma_{F(u,e)} \mathcal{T}^W_R \lambda_u \neq \emptyset$. Then $\Omega^\star_W \neq \emptyset$. If $W \in \Omega^\star_W$, then $\Omega^*_W = \{W\}$ and thus there is nothing to prove. So let $W \notin \Omega^\star_W$. Therefore $\Omega^*_W = \Omega^\star_W$. Thus

$$
\begin{array}{lll}
S' \in mod(\Omega^*_W) & \Longleftrightarrow & S' \in mod(\Omega^\star_W) \\
 & \Longleftrightarrow & S' \in mod(\Omega^{e,u}_{R,W}) \text{ for every } O(R,u) \in Tr, \\
 & & \text{such that } \Omega^{e,u}_{R,W} \neq \emptyset \\
 & \Longleftrightarrow & S' \in mod(\Delta^{e,u}_{R,W}) \text{ for every } O(R,u) \in Tr, \\
 & & \text{such that } \Delta^{e,u}_{R,W} \neq \emptyset \qquad \text{from Theorem 6.1} \\
 & \Longleftrightarrow & S' \in mod(\Gamma^{e,u}_{R,W}) \text{ for every } O(R,u) \in Tr, \\
 & & \text{such that } \Gamma^{e,u}_{R,W} \neq \emptyset \qquad \text{from Corollary 5.1} \\
 & \Longleftrightarrow & S' \in mod(\Gamma^\star_W) \qquad \text{from Corollary 3.1} \\
 & \Longleftrightarrow & S' \in mod(W). \qquad \text{from Corollary 3.1}
\end{array}
$$

Hence $S' \in mod(W)$ iff $S' \in mod(\Omega^*_W)$

◇

7.2 Deriving C^*_W from Meta Relations $\mathcal{T}^W_R$ and $\widetilde{\mathcal{T}}^W_R$

In Chapter 6, we have shown that the simplified form $C^{-,u}_{W,R}$ (resp. $C^{+,u}_{W,R}$), defined by the simplification method in case of changing state S by operation $I(R,u)$ (resp. $D(R,u)$), is the wff obtained by applying the third step of the simplification method to the conjunction of wffs of $\Omega^{e,u}_{R,W}$.

To complete this work, we will discuss in the remainder of this section whether the simplified form C^*_W, defined by the simplification method in case of changing state S by the transaction Tr, can be obtained by applying the third step of the simplification method to conjunctions of wffs in $\Omega^\star_W$. Two remarks have to be made about the set of wffs $\Omega^\star_W$.

1. For every $O(R,u) \in Tr$, $\Omega^{e,u}_{R,W} \subseteq \Delta^{e,u}_{R,W}$. Thus for every W_1 and W_2 in $\Omega^{e,u}_{R,W}$ such that $W_1 \neq W_2$, W_1 and W_2 are not identical up to the permutation of the disjunctions, a permutation of literals and a renaming of variables. So is the case for every two wffs of $\Omega^\star_W$.

2. For every $W_1 \in \Omega^{e,u}_{R,W}$, either $W_1 \in \Gamma^{e,u}_{R,W}$ or there is a $W_2 \in \Gamma^{e,u}_{R,W}$ such that W_1 is obtained by deleting some pre-valued literals from W_2. Thus, for every $W_1 \in \Omega^\star_W$, either $W_1 \in \Gamma^\star_W$ or there is a $W_2 \in \Gamma^\star_W$ such that W_1 is obtained by deleting some pre-valued literals from W_2.

Let K_W be the wff obtained by applying the third step of the simplification method to the conjunction of wffs in Ω^*_W. By the two remarks given above the wff K_W can be seen as obtained by applying the steps of the simplification method to the set $\Gamma^\star_W$. Since the simplified form C^*_W is obtained by applying the steps of the simplification method to the set of wffs Γ^*_W and $\Gamma^*_W \subseteq \Gamma^\star_W$. Then to determine whether K_W is identical to C^*_W we have to consider the following cases:

$\Gamma_W^* = \Gamma_W^\star$: In this case it is clear that K_W and C_W^* are identical.

$\Gamma_W^* \subset \Gamma_W^\star$: In this case there are two subcases:

- $W \in \Gamma_W^\star$: In this subcase, on the one hand, $W \in \Omega_W^\star$. Thus $\Omega_W^* = \{W\}$ and hence K_W is W. On the other hand, $W \in \Gamma_W^\star$ iff $\varepsilon \in G_W^*$ (cf. Definition 3.4) which by the definition of Γ_W^* implies that $\Gamma_W^* = \{W\}$. Thus C_W^* is W. Hence K_W and C_W^* are identical.
- $W \notin \Gamma_W^\star$: In this subcase, neither C_W^* nor K_W contains the constraint W. But since $\Gamma_W^* \subset \Gamma_W^\star$, C_W^* may be a subformula of K_W.

In conclusion, the simplified form C_W^* is identical to K_W, except for the subcase $W \notin \Gamma_W^\star$ of the case $\Gamma_W^* \subset \Gamma_W^\star$. In this subcase, it may happen that C_W^* is subformula of K_W. But we emphasize that in this subcase, K_W does not contain the constraint W. Thus the redundancy that may exist in K_W with respect to C_W^* is not serious.

Chapter 8

Conclusion and Future Work

In this report we have shown that all the steps of the method proposed by Nicolas in [Nic82] for simplifying constraints can done at compile time.

We have done that by developing a representation that stores simplified instances of W together with other information about occurrences of R in W into *meta relations*. The simplified instances stored in the meta relations are obtained form W by applying the same simplification steps of the method, but here we use *generic constants* instead of specific update values. When an update is performed to the relation R, the generic constants in the meta relations are replaced with the update values and a relational algebra expression is performed on the obtained relation, resulting in a set of formulas. We have proved that it is sufficient to evaluate these formulas in the new state to determine that the constraint W is satisfied in the new state. Also, we have proved that in the case of inserting (resp. deleting) a tuple u into (resp. from) the relation R, applying the third step of the method to the conjunctions of these formulas is identical the simplified form $C^{-,u}_{R,W}$ (resp. $C^{+,u}_{R,W}$) obtained by steps of the simplification method.

Many researchers have address problems of compiling constraints before the database becoming in interactive use. In this point, works presented in [HMN84, Dec87, Llo87] are related our work. However our approach different in that we use model-theoretic view for databases rather than proof-theoretic which is taken in these approaches.

The meta relations is presented in the context of passive database. However the work can be extended to active databases as follows. We can add a new component to meta relations. These component can be used to store user-defined actions or actions derived from simplified forms of meta relation using techniques of [FPT92, CFPT94]. By this way the components of meta relation can be viewed as representations for ECA rules.

Bibliography

[BM88] E. Bertino and D. Musto. Correctness of Semantic Integrity Checking in Database Management Systems. *Acta Informatica*, 26:25–57, 1988.

[CFPT94] S. Ceri, P. Fraternali, S. Paraboschi, and L. Tanca. Automatic Generation of Production Rules for Integrity Maintenance. *ACM Transactions on Database Systems*, 19(3):367–422, September 1994.

[Dec87] H. Decker. Integrity Enforcements on Deductive Databases. In L. Kerschberg, editor, *Proc. of the 1st Int. Conf. on Expert Database Systems, Charleston, South Carolina, April 1986*, pages 381–395, Menlo Park, CA, 1987. Benjamin/Cummings.

[FPT92] P. Fraternali, S. Paraboschi, and L. Tanca. Automatic Rule Generation for Constraint Enforcement. In U. W. Lipeck and B. Thalheim, editors, *Modelling database dynamics — Selected papers from the 4th Int. Workshop on Foundations of Models and Languages for Data and Objects*. British Computer Society Press, 1992.

[GMN84] H. Gallaire, J. Minker, and J.-M. Nicolas. Logic and Databases: A Deductive Approach. *ACM Computing Surveys*, 16(2):153–185, June 1984.

[HMN84] L. J. Henschen, W. W. McCune, and S. A. Naqvi. Compiling Constraint-Checking Programs from First Order Formulas. In *Advances in Database Theory, Volume II*, pages 145–169. Plenum Press, 1984.

[KSS87] R. A. Kowalski, F. Sadri, and P. Soper. Integrity Checking in Deductive Databases. In P. M. Stocker and W. Kent, editors, *Proc. of the 13th Int. Conf. on Very Large Data Bases (VLDB'87), Brighton, England*, pages 61–70, Los Altos, CA, September 1987. Morgan Kaufmann Publishers.

[LL93] T. W. Ling and S. Y. Lee. Improving Integrity Checking for ∃-Constraints. In S. C. Moon and H. Ikeda, editors, *Proc. of the 3rd Int. Conf. on Database Systems for Advanced Applications (DASFAA'93), Taejon, Korea, April 1993*, pages 343–350, Singapore, 1993. World Scientific Press.

[Llo87] J. W. Lloyd. *Foundations of Logic Programming*. Springer-Verlag, New York, 2 edition, 1987.

[MH89] W. W. McCune and L. J. Henschen. Maintaining State Constraints in Relational Databases: A Proof Theoretic Basis. *Journal of the ACM*, 36(1):69–91, 1989.

[Nic82] J.-M. Nicolas. Logic for Improving Integrity Checking in Relational Data Bases. *Acta Informatica*, 18(3):227–253, 1982.

[Qia88] X. Qian. An Effective Method for Integrity Constraint Simplification. In J. Carlis, editor, *Proc. of the 4th IEEE Int. Conf. on Data Engineering, ICDE'88, Los Angeles, CA, USA*, pages 338–345. IEEE Computer Society Press, February 1988.

[QS87] X. Qian and D. R. Smith. Integrity Constraint Reformulation for Efficient Validation. In P. M. Stocker and W. Kent, editors, *Proc. of the 13th Int. Conf. on Very Large Data Bases (VLDB'87), Brighton, England*, pages 417–425, Los Altos, CA, September 1987. Morgan Kaufmann Publishers.