**UNIVERSITAT POLITÈCNICA DE CATALUNYA**

*Departament de llenguatges i sistemes informàtics*

# ACTUALITZACIÓ CONSISTENT DE BASES DE DADES DEDUCTIVES

Autor: Enric Mayol Sarroca
Director: Ernest Teniente i López

Barcelona, 2000

# 5. Comparacions amb altres treballs

En aquesta capítol presentem la comparació de diferents mètodes d'actualització de vistes i del manteniment de restriccions d'integritat respecte al mètode presentat en aquesta tesi. Aquests treballs són els que hem considerat en l'anàlisi de l'estat de l'art resumit en la taula 1.1 de la secció 1.2 d'aquest document.

En aquesta comparació només es consideren els aspectes relacionats amb la definició dels mètodes, és a dir, el model de dades utilitzat, tipus de peticions d'actualització permeses i les característiques de les solucions obtingudes. Els aspectes de l'eficiència del mecanisme de generació de solucions d'algun d'aquests mètodes s'analitza al capítol 7 d'aquest mateix document.

Aquest capítol de comparacions s'inicia amb l'anàlisi dels mètodes [Wüt93, CHM95, CST95, Dec97, LT97] que tracten els problemes d'actualització de vistes i de manteniment de restriccions d'integritat, de forma integrada. A la secció 5.2 s'analitza un segon grup de mètodes [CFPT94, Ger94, Maa98, Sch98] que no permeten l'actualització de vistes i, que per tant, només consideren el problema del manteniment de restriccions d'integritat. El darrer grup, considerat en la secció 5.3, contempla un mètode [LPS93] que tracta el problema de l'actualització de vistes sense considerar les restriccions d'integritat. La darrera comparació es realitza a la secció 5.4. En aquest darrer cas s'analitzen les principals diferències entre el mètode presentat en aquesta tesi i el seu precursor: el Mètode dels Esdeveniments. Una anàlisi detallada dels mètodes [GL90, KM90] pot trobar-se a [Ten92, TO95].

Dins de cada secció, es presenta cada un dels mètodes analitzats amb detall i, per a cada un d'ells, es descriuen els trets principals del mètode i es comenten els principals problemes o limitacions que presenten respecte al nostre mètode.

## 5.1 Comparació amb mètodes d'actualització de vistes i manteniment de restriccions d'integritat.

Com ja hem comentat anteriorment, l'actualització de vistes i el manteniment de restriccions d'integritat són dos problemes íntimament relacionats, que s'han de afrontar de forma integrada. Així doncs, els treballs que tracten aquests problemes ho fan de forma integrada proposant un únic mètode per a resoldre els dos problemes. Els mètodes analitzats en aquesta secció són exemples de treballs que segueixen, de la mateixa manera que el nostre mètode, aquest enfocament integrador.

La forma com es consideren la petició d'actualització i la informació relativa a les restriccions d'integritat en cadascun dels mètodes és diferent. De totes maneres, hem detectat dues estratègies d'integració clarament diferenciades. En primer lloc, tenim un grup de treballs

[Wüt93, CST95, LT97], que donada una petició d'actualització sobre fets bàsics i derivats, estenen aquesta petició amb la incorporació de la informació referent a les restriccions d'integritat definides a la base de dades. Així doncs, traduint aquesta petició d'actualització estesa s'assegura el manteniment de la consistència de la base de dades. Un altre grup de mètodes, com els presentats a [CHM95, Dec97], no incorporen la informació de les restriccions a la petició d'actualització, sinó que és el propi mecanisme de traducció el que ha de considerar en algun moment del procés, les restriccions d'integritat que es poden veure afectades per les actualitzacions proposades per a satisfer la petició d'actualització.

Aquesta distinció ens serveix per a fer una descripció més estructurada dels mètodes analitzats i comentar els problemes o limitacions que presenten en comú. A la secció 5.1.1 es descriuen el primer grup de mètodes i a la secció 5.1.2 el segon grup.

### 5.1.1 Mètodes que estenen la petició d'actualització

Els mètodes analitzats en aquesta secció [Wüt93, CST95, LT97] coincideixen en estructurar el mecanisme de traducció de la petició d'actualització dividit en dues etapes. La primera etapa consisteix en aplicar un mecanisme de *desplegament* (unfolding, en anglès) per a obtenir, a partir de la petició d'actualització, una única fórmula F definida tant sols per a predicats bàsics, que caracteritza totes les solucions de la petició d'actualització. Bàsicament, aquesta etapa consisteix en incorporar a la petició d'actualització la informació relativa a les restriccions d'integritat i anar substituint els predicats derivats que hi apareixen per la seva corresponent definició. Al final d'aquesta etapa s'obté una fórmula F expressada únicament amb predicats bàsics i predicats avaluables. En la segona etapa, a partir de la fórmula F es dedueixen els fets bàsics que cal inserir i els fets bàsics que cal esborrar de la base de dades per tal de satisfer la petició d'actualització inicial i de no violar cap restricció d'integritat.

La principal diferència entre aquests mètodes rau en la forma com s'integra la informació de les restriccions d'integritat a la petició d'actualització inicial, en la primera etapa del procés.

A continuació es pot trobar una anàlisi detallada de cada un d'aquests mètodes i la seva comparació amb el mètode proposat en aquesta tesi.

#### 5.1.1.1 Mètode de B. Wüthrich [Wüt93]

En aquest mètode, una petició d'actualització U consisteix en una conjunció d'insercions i esborrats de fets bàsics i derivats, estesa amb la conjunció de totes les restriccions d'integritat per tal d'assegurar el manteniment de la consistència de la base de dades.

Un cop obtinguda una fórmula U* equivalent a petició inicial U, expressada únicament amb predicats bàsics, es dedueixen els fets bàsics que cal inserir i els fets bàsics que cal esborrar de la base de dades. En aquest procés, aquest mètode requereix la utilització de dos conjunts auxiliars $T_1$ i $T_2$ on s'acumulen subfòrmules de la fórmula U*. Al conjunt $T_2$ s'hi acumulen

aquelles subfòrmules que van esdevenint certes, mentre que al conjunt $T_1$ les que esdevenen falses. La finalitat d'aquests conjunts és poder comprovar que tota fórmula que s'havia fet certa (o falsa) en algun moment, no esdevingui falsa (o certa) per la proposta de noves insercions o esborrats de fets bàsics.

Tota solució es representa per dos conjunts d'actualitzacions, el conjunt I d'insercions i el conjunt D d'esborrats, els quals han de ser sempre disjunts. La solució finalment obtinguda depèn dels conjuntands i disjuntands de la fórmula U* que s'han considerat durant la segona etapa del procés de traducció.

Una primera diferència entre aquest mètode i el nostre fa referència al tractament dels conjunts $T_1$ i $T_2$. D'alguna manera, aquests conjunts tenen una finalitat similar a la del conjunt de condicions C del nostre mètode. Però, la diferència està en que per a assegurar que les fórmules del conjunt $T_2$ ($T_1$) es mantenen certes (o falses), aquest mètode aplica una política de comprovació d'aquestes fórmules, en canvi, en el nostre mètode s'aplica una política de manteniment.

Aquest mètode presenta bàsicament dos problemes: en certs casos, aquest mètode no és capaç d'obtenir totes les solucions que existeixen per a una petició d'actualització U. El segon problema rau en que al no considerar l'extensió de la base de dades, pot generar solucions que no són mínimes.

*No obté totes les solucions*

En certs casos, aquest mètode no es capaç de generar totes les solucions que són correctes. El problema és degut a que aquest mètode suposa que sempre existeix un ordre concret per a considerar les regles de derivació i les restriccions d'integritat. En realitat, hi ha casos en què aquest ordre no existeix.

**Exemple 5.1**: Suposem la petició d'actualització consistent en la inserció del fet Aresta(A, C) a la base de dades següent:

Node(A) Aresta(A, B)
Node(B) Aresta(B, A)

$Ic1 \leftarrow Node(x) \wedge \neg \exists y\ Aresta(x, y)$

$Ic2 \leftarrow Node(x) \wedge \neg \exists z\ Aresta(z, x)$

$Ic3 \leftarrow Aresta(x, y) \wedge \neg Node(x)$

$Ic4 \leftarrow Aresta(x, y) \wedge \neg Node(y)$

El mètode de Wüthrich no és capaç d'obtenir la solució S = {Aresta(A, C), Node(C), Aresta(C, D), Node(D), Aresta(D, B) }. En canvi, el nostre mètode no està condicionat a aquest ordre de tractament i sí que obté aquesta solució. □

*No considera la base de dades extensional*

El segon problema que presenta aquest mètode és que pot generar solucions que no són mínimes. Aquest fet és degut a que a l'incloure una nova actualització en els conjunts I o D, no comprova prèviament quin és el contingut de la base de dades extensional. Així doncs, les solucions que genera aquest mètode poden contenir actualitzacions redundants i, per tant, no ser mínimes.

**Exemple 5.2**: Considerem la petició per a inserir el fet P(A) a la base de dades següent:

$S(A, B)$

$P(x) \leftarrow Q(x) \wedge R(x)$

$R(x) \leftarrow S(x, y)$

Aquest mètode obté, entre altres, la solució caracteritzada pels conjunts $I = \{Q(A), S(A,C)\}$ i $D = \emptyset$, on C és un valor qualsevol del domini assignat per l'usuari. El nostre mètode aprofita que el fet S(A, B) és cert a la base de dades i obté, a més a més, la solució $S = \{\iota Q(A)\}$. Aquesta solució solament seria obtinguda per aquest mètode en el cas que l'usuari assignés a la variable 'y' el valor B en lloc del C anterior.

□

### 5.1.1.2 Mètode de L. Console, M. L. Sapino i D. Theseider [CST95]

L. Console, M. L. Sapino i D. Theseider proposen a [CST95] un mètode basat en abducció per a l'actualització de vistes i el manteniment restriccions d'integritat en bases de dades deductives. Donada una petició d'actualització $\phi$, el mètode obté totes les traduccions possibles que satisfan la petició d'actualització i que no violen cap restricció d'integritat.

El procés de traducció està dividit en dues etapes. A la primera, i de manera similar al mètode de Wüthrich, la petició d'actualització $\phi$ es transforma en una fórmula F* expressada únicament en termes de predicats bàsics i predicats avaluables. A la segona etapa del procés de traducció, es té en compte la base de dades extensional per a simplificar i instanciar les variables existencials que apareguin a F*. Al final d'aquest procés s'obté una fórmula totalment instanciada F** que defineix les diferents solucions (transaccions) que satisfan la petició d'actualització $\phi$ sense violar cap restricció d'integritat.

Aquests mètode presenta una limitació i un problema. En primer lloc, les restriccions d'integritat que poden ser expressades amb el seu llenguatge estan considerablement limitades. En segon lloc, i en certs casos, aquest mètode obté solucions que no sempre tenen perquè ser mínimes.

*Restriccions d'integritat limitades a dos tipus específics*

Una limitació important que presenta aquest mètode [CST95] està relacionada amb el poder expressiu del llenguatge de definició de les restriccions d'integritat, ja que aquestes sols poden

estar definides per predicats bàsics i, com ja ha estat comentat a la introducció (secció 1.2), aquest fet limita considerablement el conjunt de restriccions d'integritat que aquest mètode pot expressar i mantenir.

A més a més, les restriccions d'integritat que aquest mètode pot mantenir han de ser d'un dels tipus següents:

1) restriccions expressades com a denegacions, on en el cos de la regla tan sols hi poden aparèixer dos literals. És a dir, de la forma següent: $\leftarrow P(x) \wedge Q(y)$

2) restriccions d'integritat referencial acícliques del tipus $P(x) \rightarrow \exists y\, Q(x, y)$

En la nostra opinió, aquests tipus de restriccions d'integritat són una limitació considerable d'aquest mètode ja que hi ha moltes restriccions d'integritat que no es poden expressar amb aquests formats. Per exemple, aquest mètode no pot gestionar les restriccions d'integritat de clau, expressades de la forma següent:

$$\leftarrow P(k, x) \wedge P(k, y) \wedge x \neq y$$

ja que té més de dos literals al cos de la regla.

A més, no totes les restriccions d'integritat referencial que involucrin algun predicat derivat poden expressar-se tant sols en funció de predicats bàsics i amb el format 2). Per exemple, siguin Q, R i S predicats bàsics i P un predicat derivat definit per:

$$P \leftarrow R$$

$$P \leftarrow S$$

la restricció d'integritat referencial $Q \rightarrow P$ no pot ser expressada per aquest mètode, ja que per a definir-la, utilitzant solament predicats bàsics s'hauria de escriure de la forma següent:

$$Q \rightarrow (R \vee S)$$

expressió que no coincideix amb cap dels dos tipus de restriccions permesos.

En canvi, aquestes dues restriccions d'integritat poden ser expressades en forma de denegació amb predicats bàsics i derivats. Per tant, totes elles poden ser expressades i tractades pel nostre mètode. No cal dir, que en aquest sentit, el nostre mètode pot expressar i mantenir moltes restriccions que el mètode de [CST95] no pot tractar.

*Solucions no mínimes*

Les solucions caracteritzades per la fórmula F** no sempre són mínimes. En certs casos, hi ha disjuntands de F** que són redundants respecte a d'altres disjuntands i, en altres casos, n'hi ha que poden contenir actualitzacions que no són estrictament necessàries.

**Exemple 5.3**: Suposem la petició d'actualització $\phi = \{\neg P\}$ que correspon a l'esborrat del fet derivat P de la base de dades següent:

$$R(1) \quad R(2) \quad S(2) \qquad P \leftarrow R(x) \wedge \neg S(x)$$

La fórmula obtinguda és $F^{**} = [\neg R(1) \wedge \neg R(2)] \vee [\neg R(1)] \vee [S(1) \wedge \neg R(2)] \vee [S(1)]$. Cada disjuntand de F** es correspon a una solució a la petició $\phi$. Observi's que en realitat, dels quatre disjuntands el primer i tercer no es corresponen a solucions mínimes. El literal $\neg R(2)$ dels dos disjuntands (indicant l'esborrat del fet R(2)) és totalment redundant i innecessari per a satisfer la petició inicial.

En aquest mateix exemple, el nostre mètode només genera les dues solucions mínimes existents: $T_1=\{\delta R(1)\}$ i $T_2=\{\iota S(1)\}$.

□

### 5.1.1.3 Mètode de J. Lobo i G. Trajcevsky [LT97]

J. Lobo i G. Trajcevsky proposen a [LT97] un algorisme que obté una única solució (mínima) que satisfà la petició d'actualització sense violar cap restricció d'integritat.

Aquest algorisme consisteix bàsicament en aplicar un procés de *desplegament* a tots els predicats derivats que apareixen a la petició d'actualització, obtenint una fórmula F en forma normal disjuntiva equivalent a la petició d'actualització. Aquesta fórmula F és ampliada amb els residus de les restriccions d'integritat que podrien ser violades per les actualitzacions contingudes en la fórmula F. Un cop ampliada aquesta fórmula, les variables no instanciades que resten a la fórmula F són instanciades tenint en compte els fets existents a la base de dades extensional. Al final d'aquest procés s'obté una fórmula normal disjuntiva totalment instanciada que caracteritza les diferents solucions a la petició inicial. Cada disjuntand defineix una possible solució S que satisfà la petició inicial d'actualització i no viola cap restricció d'integritat.

En aquest mètode en particular, no es permet la seva definició de restriccions d'integritat en termes de predicats derivats, i a més a més, imposen que el conjunt de restriccions d'integritat sigui *resolution complete*. Respecte a les solucions que aquest mètode pot obtenir, cal indicar que en certs casos, aquest mètode pot obtenir solucions que no són correctes.

*Limitacions en la definició de les restriccions d'integritat*

Una diferència important entre el nostre enfocament i aquest mètode, està en la definició de les restriccions d'integritat. Mentre nosaltres permetem la definició de restriccions d'integritat utilitzant tant predicats bàsics com predicats derivats, a [LT97] les restriccions d'integritat tan sols poden estar definides per predicats bàsics. Com ja hem comentat amb anterioritat, aquest fet limita considerablement el poder expressiu de les restriccions d'integritat.

A més a més, els autors imposen que les restriccions d'integritat han de ser *resolution complete*. És a dir, a partir de les restriccions d'integritat definides a la base de dades, no s'ha de poder deduir cap altre restricció d'integritat implícita.

**Exemple 5.4**: Suposem definides les restriccions d'integritat següents:

$\leftarrow Q(x) \wedge \neg R(x)$

$\leftarrow R(x) \wedge S(x)$

aquestes restriccions no són *resolution complete*, perquè es pot deduir una tercera restricció a partir de les anteriors. Aquesta restricció és la següent:

$\leftarrow Q(x) \wedge S(x)$

□

El problema que presenta la imposició de la condició de *resolution complete* sobre les restriccions d'integritat és que, pel que nosaltres coneixem, no s'ha definit un mecanisme que a partir d'un conjunt de restriccions que no és *resolution complete* generi el conjunt equivalent que sí que ho sigui.

*Solucions que no satisfan la petició inicial*

Un primer problema que presenta el mètode de [LT97] és que, en certs casos, la fórmula obtinguda al final procés de traducció F no sempre caracteritza solucions correctes. Hi ha casos en què a causa de la negació, alguna de les solucions caracteritzada per un disjuntand de F pot no satisfer la petició d'actualització inicial, encara que sí es satisfacin les restriccions d'integritat.

**Exemple 5.5**: Suposem la petició consistent en la inserció del fet derivat Q(B,2) a la base de dades següent:

$S(A,1)$

$Q(x, y) \leftarrow \neg P \wedge S(x, y)$

$P \leftarrow S(x, y) \wedge \neg T(y)$

La fórmula obtinguda per l'aplicació d'aquest mètode és $F= [\neg S(A,1) \wedge S(B,2)] \vee [T(1) \wedge S(B,2)]$. En aquest cas cap de les dues solucions associades satisfà la petició inicial d'actualització. El segon literal de cada solució és necessari per a satisfer la petició d'actualització, però a la vegada indueix que el fet derivat P esdevingui cert, de forma que la petició d'actualització $U = \{Q(B,2)\}$ no s'aconsegueix.

En aquest mateix exemple, el nostre mètode obté la solució $T_1=\{\iota T(1), \iota S(B, 2), \iota T(2)\}$ i la solució $T_2=\{\delta S(A, 1), \iota S(B, 2), \iota T(2)\}$. Ambdues satisfan la petició inicial d'actualització. L'actualització $\iota T(2)$ compensa l'efecte indesitjable de $\iota S(B, 2)$ i permet així satisfer la petició inicial.

□

### 5.1.2 Mètodes que consideren les restriccions durant el procés de traducció

Els dos treballs analitzats en aquesta secció [CHM95] i [Dec96, Dec97] tenen mecanismes de traducció diferents. El primer mètode [CHM95] utilitza regles actives per a reparar les

possibles violacions de les restriccions d'integritat. Durant el procés de traducció, la informació de les restriccions d'integritat es considera únicament quan es proposa una actualització que pot violar alguna restricció d'integritat. El segon mètode [Dec96, Dec97] es basa en una extensió abductiva del procediment de resolució SLD. En aquest mètode, cada cop que es considera una nova actualització, es comprova que aquesta no violi cap restricció d'integritat. De forma, que en els dos mètodes, les restriccions d'integritat es consideren durant el procés de traducció i només en el moment en que es proposa alguna actualització que les pugui afectar.

#### 5.1.2.1 Mètode de I. A. Chen, R. Hull i D. McLeod [CHM95]

Chen, Hull i McLeod proposen a [CHM95] un model d'execució basat en regles actives per a l'actualització i comprovació de restriccions d'integritat en un model semàntic de base de dades orientat a objectes.

El model de base de dades utilitzat permet definir classes i atributs derivats, així com, diferents tipus de restriccions d'integritat: restriccions implícites del model d'objectes (de tipus d'atributs, ISA, ...) i restriccions definides explícitament (disjunció entre classes, cardinalitat d'atributs, inclusió, pertinença ..).

El model presentat en aquest treball està basat en el que anomenen *Limited Ambiguity Rules (LAR)* que obtenen de forma automàtica a partir de les restriccions d'integritat, classes i atributs derivats de l'esquema de la base de dades. Una regla *LAR* és una regla activa del tipus CA (condició-acció). Les regles *LAR* es divideixen en diferents grups: regles *upward* que permeten una propagació ascendent dels canvis sobre classes o atributs bàsics cap a classes o atributs derivats. Les regles *downward* permeten una propagació descendent dels canvis. La resta de regles serveixen per a detectar errors, violacions d'integritat o l'existència d'infinites solucions.

Donada una petició d'actualització $\Delta_{user}$, l'algorisme d'execució de les regles *LAR* permet obtenir totes les solucions que satisfan la petició i que no violen cap restricció d'integritat. L'ordre d'execució d'aquestes regles està definit pel *Principle of Down-Up Propagation*, que bàsicament consisteix en executar en primer lloc totes les regles *downward* i posteriorment totes les *upward*. La resta de regles ens permeten assegurar que l'algorisme d'execució acaba amb un conjunt de solucions o una notificació d'error a causa de possibles violacions de restriccions d'integritat o l'existència d'infinites solucions.

Aquest mètode presenta una diferència primordial respecte la resta de mètodes analitzats en aquesta secció 5, ja que aplica gairebé sempre una política de comprovació de restriccions d'integritat encara que per algun tipus de restricció molt simple i específica proposa formes de reparar possibles violacions.

Però, el principal problema que presenta aquest mètode és que en certs casos no és capaç de generar totes les solucions a una petició d'actualització.

*Comprovació de restriccions d'integritat*

La diferència més important d'aquest mètode respecte al nostre està en el tractament que es fa de les restriccions d'integritat. Mentre que nosaltres apliquem una política de manteniment de les restriccions d'integritat, en aquest treball, les restriccions d'integritat només es comproven. Així doncs, al violar-se una restricció d'integritat es rebutja la transacció que s'està considerant. Evidentment, el considerar una política de comprovació de restriccions d'integritat impedeix que s'obtinguin solucions vàlides que altres mètodes que apliquen una política de manteniment de restriccions d'integritat sí poden obtenir. En aquest sentit, tal com es considera a [Ten92], el manteniment de restriccions d'integritat és més eficaç que la comprovació de les mateixes.

De totes formes, cal comentar que per a uns tipus molt concrets de restriccions d'integritat com, per exemple, restriccions sobre el domini o rang dels atributs, restriccions que imposen que un atribut sigui univaluat o que els valors de dos atributs siguin disjunts, aquest mètode aplica una política de manteniment de restriccions d'integritat. Aquestes restriccions d'integritat són tingudes en compte en la generació automàtica de les regles *LAR* i es proposen actualitzacions alternatives que evitin violar aquestes restriccions. Així doncs, les transaccions que podrien deixar la base de dades inconsistent, són rebutjades durant la seva generació.

*No troba algunes solucions vàlides*

Hi ha casos, en que aquest mètode pot no trobar solucions a una petició d'actualització que el nostre mètode si pot obtenir. Fins i tot, si considerem el cas de no violar-se cap restricció d'integritat es pot seguir donant aquest problema. Suposem una transacció que permeti induir a la vegada la inserció i l'esborrat d'un mateix fet derivat. Davant d'aquesta situació, el mètode de [CHM95] rebutja aquesta transacció com a incorrecte. Aquest mètode no té cap possibilitat d'evitar aquesta solució. Això és degut a què, en primer lloc, aquest fet es detecta al final del procés de traducció i, en segon lloc, perquè el principi de *Down-Up Propagation* no permet l'execució de cap regla *downward* després de l'execució de les regles d'*upward*, fet que permetria considerar actualitzacions alternatives que desfessin aquesta situació. Amb el nostre mètode i davant d'aquesta situació, es proposen actualitzacions addicionals que permeten obtenir una transacció consistent.

**Exemple 5.6**: Suposem la base de dades següent a la que es vol aplicar la petició d'actualització $\Delta_{user}$ = {+(P, has-instance, Id1), -(S, has-instance, Id1)} que consisteix en la inserció i esborrat d'un objecte identificat per Id1 de les classes P i S respectivament.

CLASS P

derivation : Q and R

CLASS R

derivation: S or T

DB = {(S, has-instance, Id1), (R, has-instance, Id1)}

on Q, T i S són classes bàsiques i les classes derivades són P i R. La classe P es defineix com la conjunció de Q i R i la classe R es defineix com la disjunció entre S i T.

En aquest exemple, el mètode de [CHM95] no obté cap solució, ja que al proposar esborrar el fet (S, has-instance, Id1), les regles *upward* indueixen un esborrat de (P, has-instance, Id1) el qual es incompatible amb la seva inserció proposada en la petició d'actualització $\Delta_{user}$.

En aquest exemple, el nostre mètode obtindria la solució $\Delta = \{\delta S(Id1), \iota Q(Id1), \iota T(Id1)\}$. On la inserció $\iota T(Id1)$ evita induir l'esborrat $\delta P(Id1)$ i permet satisfer la petició d'actualització demanada.

□

### 5.1.2.2 Mètode H. Decker [Dec96, Dec97]

El mètode proposat per H. Decker a [Dec97] està orientat a l'actualització de vistes i manteniment de restriccions d'integritat en bases de dades deductives. Aquest mètode està basat en un procediment de demostració anomenat SLDAI, que consisteix en una extensió del procediment de resolució SLD amb un mecanisme abductiu.

Donada una petició d'actualització, aquest mètode obté una solució que conté insercions i esborrats de fets bàsics que satisfan la petició inicial i no violen cap restricció d'integritat. Per a obtenir totes les solucions possibles, aquest mètode ha de reconsiderar cadascun dels literals seleccionats en l'arbre de derivació.

D'una forma similar al nostre mètode, el procediment SLDAI està definit per l'alternança entre una fase constructiva (*Refutation*) i una fase de consistència (*Consistency*). En la primera, a partir d'un objectiu es pretén assolir la clàusula buida utilitzant el contingut de la base de dades i un conjunt d'hipòtesis H. Cada cop que es vol incloure una nova hipòtesi en el conjunt H, cal comprovar que no es contradiu amb altres hipòtesis ni amb cap restricció d'integritat mitjançant una fase de consistència. Sempre que en l'etapa de consistència es vol considerar una nova hipòtesi, cal iniciar una nova fase constructiva.

La principal limitació que té aquest mètode és que no és adequat pels casos en que la traducció de la petició d'actualització involucra alguna regla que contingui variables existencials perquè el procediment de traducció no tracta correctament aquests casos.

Aquest tractament incorrecte ve motivat per causes diferents segons la regla existencial sigui considerada en una fase constructiva o de consistència. En el cas de la fase constructiva, el problema és que el mètode no sap com tractar els fets bàsics no instanciats. En el cas de la fase de consistència, el problema és que no es té en compte el contingut de la base de dades extensional i això pot provocar que en molts casos no s'obtinguin solucions que realment existeixen.

**Exemple 5.7**: Suposem una base de dades amb una sola regla de derivació $P \leftarrow S(x)$. Amb la petició d'inserir el fet derivat P, aquest mètode no genera cap solució perquè s'entrebanca, és a dir, entra en un procés d'*entrebanc* (en anglès, floundering) al no saber com tractar un literal corresponent a un fet bàsic no instanciat.

$\leftarrow P$
|
$\leftarrow S(x)$
|
s'entrebanca

En realitat, suposant dominis finits per la variable 'x', existeixen tantes solucions com possibles valors correctes hi hagi en aquest domini. Amb el nostre mètode es consideren tantes branques com elements hi hagi en el domini.

□

En l'exemple següent, es pot comprovar que el mètode de Decker no obté cap solució, quan en realitat n'existeixen dues. Això es degut a que en l'etapa de consistència no es considera la base de dades extensional.

**Exemple 5.8**: Sigui la petició d'actualització d'inserir el fet P a la base de dades següent:

$R(A, B) \leftarrow$

$P \leftarrow Q(A)$

$\leftarrow Q(x) \wedge R(x, y) \wedge \neg S(y)$

Els arbres de derivació de la fase constructiva i de la de consistència associats són els següents:

REF.
$\leftarrow P$
|
$\leftarrow Q(A)$
|
s'entrebanca

CONS.
$Q(A) \leftarrow$ $H = \{Q(A)\}$
|
$\leftarrow R(A, y) \wedge \neg S(y)$ (*)
|
s'entrebanca

Al intentar resoldre qualsevol literal de l'objectiu (*), el mètode s'entrebanca perquè no sap tractar cap dels dos literals al no estar completament instanciats i, per tant, el mètode no pot tenir en compte el fet R(A, y) de la base de dades extensional. Això provoca que el mètode no obtingui cap solució a la petició inicial.

En aquest exemple, tenint en compte la base de dades extensional, el nostre mètode obté les solucions $T_1 = \{\iota Q(A), \delta R(A, B)\}$ i $T_2 = \{\iota Q(A), \iota S(B)\}$, que són les dues úniques solucions vàlides que satisfan la petició inicial.

□

## 5.2 Comparació amb mètodes de manteniment de restriccions d'integritat

En aquesta secció s'analitzen aquells treballs orientats únicament al manteniment de restriccions d'integritat i que no permeten tractar el problema de l'actualització de vistes. Així doncs, les peticions d'actualització solament poden incloure actualitzacions de fets bàsics.

Els treballs analitzats en aquesta secció tenen en comú el fet de generar i utilitzar un conjunt de regles actives (o similars) per a reparar les violacions de les restriccions d'integritat provocades per l'aplicació d'una transacció sobre la base de dades. A més a més, aquests treballs sols poden tractar restriccions d'integritat definides amb predicats bàsics i predicats avaluables, ja que aquests mètodes no saben com tractar l'actualització de vistes, i per tant, tenen limitada la seva aplicabilitat a bases de dades molt concretes.

Les regles actives considerades pels mètodes analitzats poden ser de dos tipus: regles *ECA* (Esdeveniment-Condició-Acció) o regles *CA* (Condició-Acció). La diferència entre unes i altres es troba en el mecanisme de detecció del moment en què s'han d'executar. En el primer cas, al detectar que un esdeveniment E ha ocorregut, la regla activa s'executa comprovant en primer lloc si les condicions de C es compleixen i, en cas afirmatiu, s'executen les accions especificades en A. En el cas de les regles CA, l'execució de la regla s'inicia quan es comprova que les condicions de C són certes i, llavors, s'executen les accions d'A.

Aquestes regles actives es defineixen en temps de compilació tenint en compte la definició de les restriccions d'integritat. En temps d'execució, un cop s'ha aplicat una transacció T sobre la base de dades, les actualitzacions de T poden provocar que alguna restricció d'integritat esdevingui violada i aleshores, les regles actives s'executen i realitzen noves actualitzacions sobre la base de dades per tal de reparar aquestes violacions.

Hi ha dues diferències primordials entre els mètodes analitzats que segueixen aquest enfocament actiu i el nostre enfocament. En primer lloc, i com ja hem comentat anteriorment, no es permet l'actualització de vistes i, per tant, les restriccions d'integritat no es poden definir utilitzant predicats derivats. La segona diferència està el moment en què es determina quina és l'actualització que cal aplicar per tal de reparar la violació d'una restricció d'integritat. En un enfocament actiu, aquesta acció reparadora ha estat definida en temps de compilació al generar les regles actives. En canvi, en el nostre enfocament, aquesta actualització reparadora es determina en temps d'execució, en el mateix moment que es detecta la violació de la restricció d'integritat.

Un problema identificat i estudiat a [Sch98, ST99], que presenten molts dels mètodes que segueixen un enfocament actiu pel manteniment de restriccions d'integritat, és el de no garantir la preservació de l'efecte de la transacció inicial T. Hi ha casos, en que les actualitzacions realitzades per les regles actives al reparar una restricció d'integritat desfan les actualitzacions realitzades per la transacció inicial T. En aquests casos, aquest enfocament actiu garanteix que

l'estat final de la base de dades és consistent, però no pot assegurar que se satisfaci la petició d'actualització inicial T. Els mètodes [CFPT94, Ger94, Maa98] analitzats en aquesta secció tenen aquest problema. En canvi, el mètode [Sch98] està bàsicament orientat a resoldre aquesta anomalia.

**Exemple 5.9**: Considerem la petició d'actualització T = {ιS(5,6), ιT(6,6)} que consisteix en la inserció del fet S(5, 6) i del fet T(6,6) a la base de dades següent:

R(5, 6)

Ic1(x, y, z) ← S(x, y) ∧ T(y, z) ∧ ¬ R(x, z)

Ic2(x, y, z) ← S(x, y) ∧ R(z, y)

Els mètodes [CFPT94, Ger94, Maa98] permeten obtenir, entre d'altres, una solució composada per les actualitzacions següents: {ιS(5,6), ιT(6,6), δR(5,6), δS(5,6)}. Les dues primeres actualitzacions són necessàries per a satisfer la petició d'actualització inicial. La tercera actualització repara la restricció Ic2 i la darrera actualització repara la restricció Ic1. L'estat final resultant és consistent, però el fet S(5,6) no és cert a la base de dades, ja que al reparar la restricció Ic1 s'ha desfet l'efecte desitjat de l'aplicació de T.

□

Aquesta situació es dóna perquè, en l'enfocament actiu, no es pot distingir entre els fets que eren certs en l'estat antic de la base de dades, les actualitzacions realitzades per T i les realitzades per les regles actives que ja s'han executat.

Amb el nostre mètode sí que es pot fer aquesta distinció ja que no s'actualitza la base de dades fins al final del procés de manteniment de les restriccions d'integritat. A més a més, el nostre mètode preveu aquestes situacions amb la gestió de les condicions del conjunt C. En aquest mateix exemple el nostre mètode no obtindria cap solució, ja que no es pot assegurar a la vegada la consistència de la base de dades i la satisfacció de la petició d'actualització T.

A les seccions següents hi descrivim algunes particularitats específiques de cada mètode.

### 5.2.1 Mètodes de S.Ceri, P. Fraternali, S. Paraboschi i L. Tanca [CFPT94]

Un treball especialment rellevant en l'àmbit del manteniment de les restriccions d'integritat a les bases de dades actives és el treball presentat a [CFPT94]. Aquest treball proposa l'arquitectura d'un sistema basat en regles actives pel manteniment de restriccions d'integritat en bases de dades relacionals.

Aquest sistema genera de forma automàtica (amb participació puntual del dissenyador) i en temps de compilació les regles actives o de produc ´ (CA rules) necessàries per reparar les restriccions d'integritat definides a una base de dades. Per a obtenir aquestes regles, es diferencien tres etapes. En la primera s'obtenen, a partir de la definició de les restriccions d'integritat, el conjunt complet de regles actives que poden reparar les restriccions d'integritat. En una segona etapa, es construeix un graf d'activació on s'analitzen quines regles poden

activar i quines regles poden ser activades per altres regles. Aquest graf serveix per a poder donar pesos a aquestes regles i per a eliminar els cicles entre regles actives. Així doncs, el nou conjunt de regles obtingut té assegurat que la seva execució finalitza. En una tercera etapa, es defineix un ordre total entre aquestes regles actives, s'eliminen les regles redundants i es comprova que tota restricció tingui almenys una regla activa que la repara.

Un punt en comú entre aquest mètode i el nostre, a diferència dels altres mètodes analitzats fins ara, és que ambdós admetem la modificació com un nou tipus d'actualització. Però en aquest cas, sols es permeten actualitzacions de fets bàsics.

Una diferència entre aquest treball i el nostre mètode està en que, encara que aquest permet definir vistes, no gestiona directament la seva actualització. En particular, aquest mètode permet definir vistes per a millorar el poder expressiu de les restriccions d'integritat, però en cap moment defineix cap mecanisme per a poder-les actualitzar, ni tant sols es defineix com adaptar un mecanisme d'actualització de vistes existent al seu mètode. Això fa que, en la pràctica, amb aquest mètode, les restriccions d'integritat definides en termes de vistes no puguin ser mantingudes adequadament.

Una altra diferència d'aquesta proposta respecte al nostre mètode està en el paper del dissenyador i/o l'usuari final durant tot el procés del manteniment de les restriccions d'integritat. En el nostre mètode, l'usuari final pot participar per a escollir (si cal) quina de les solucions aplicar sobre la base de dades. En canvi, en el treball de [CFPT94], la participació del dissenyador pot ser decisiva durant la generació de les regles actives. En particular, el dissenyador en determinats casos ha de participar en la generació de les regles actives modificant els pesos assignats a les regles, simplificant els arcs del graf d'activació, o bé modificant el conjunt de regles actives generades automàticament.

Altres limitacions d'aquest treball recauen bàsicament en certes restriccions imposades en el llenguatge de definició de les restriccions d'integritat i en el fet que, en certs casos, no poden generar totes les possibles formes de reparar una restricció d'integritat.

*Restriccions d'integritat limitades*

A més de les limitacions que ja han estat comentades a l'inici d'aquesta secció, aquest mètode imposa la limitació addicional de què en el cos de la restricció d'integritat no hi poden aparèixer dos literals del mateix predicat amb signe oposat (*twin predicates*). Així doncs, amb aquest mètode no es poden definir restriccions com la següent:

$$\leftarrow P(x, y) \wedge \neg P(y, x)$$

Davant d'aquest tipus de restriccions d'integritat, l'algorisme de generació de regles actives no és complet i el dissenyador és el que ha de definir a mà les regles actives necessàries per a reparar aquestes restriccions.

*No generen totes les solucions*

Abans de finalitzar el procés de generació de les regles actives, de totes les regles actives que es porten generades, s'eliminen aquelles que són redundants o que poden provocar execucions infinites. Aquest fet provoca que les solucions obtingudes per aquest mètode depengui directament del conjunt de regles finalment considerat. Així doncs, a l'eliminar alguna regla activa (o modificar els seus pesos), no es podran generar tots les possibles formes de reparar cada restricció d'integritat, i conseqüentment, no es podran generar totes les possibles solucions a una petició d'actualització.

Amb el nostre mètode es tenen en compte totes les formes de reparar una restricció d'integritat i per tant obtenim totes les solucions que satisfan la petició d'actualització i no violen cap restricció d'integritat.

### 5.2.2 Mètode de M. Gertz i U. W. Lipeck [GL93, Ger94]

Gertz i Lipeck [GL93, Ger94] proposen un mètode pel manteniment de restriccions d'integritat en bases de dades actives. Aquest mètode, a l'igual que el mètode de [CHM95], permet aplicar diferents polítiques d'imposició de restriccions d'integritat en una mateixa base de dades i, fins i tot, com en aquest mètode, a la mateixa restricció. Segons quina sigui l'actualització que viola una restricció, el dissenyador pot decidir aplicar diferents polítiques per la imposició de la restricció d'integritat violada: es pot aplicar manteniment, proposant altres actualitzacions reparadores; es pot aplicar comprovació, desfent l'actualització realitzada; o, fins i tot, es pot considerar aquesta violació com una excepció a la restricció d'integritat, si no es prenen mesures correctores.

En una primera etapa, aquest mètode construeix en temps de compilació un graf de dependències entre els violadors potencials i els reparadors de cada restricció d'integritat. Aquest graf es construeix a partir de la definició de les restriccions d'integritat i de les diferents reaccions que ha definit el dissenyador. A partir d'aquest graf de dependències es genera el conjunt ordenat de les regles actives necessàries per a mantenir les restriccions d'integritat. L'execució d'aquestes regles actives ens asseguraran el manteniment de la consistència de la base de dades.

De la mateixa manera que el mètode anterior ([CFPT94]) aquest també permet definir la modificació com un tipus d'actualització, i també restringit a modificacions de fets bàsics.

Per altra banda, existeixen dues diferències primordials entre aquest enfocament i el nostre. En primer lloc, les restriccions d'integritat han de estar de. .des en *Forma Normal Implicativa* en termes de predicats bàsics o avaluables, i on no està permès l'ús de la negació. En segon lloc, aquest mètode no es capaç d'obtenir totes les solucions existents ja que per cada violació d'una restricció no es consideren tots els possibles reparadors.

*No generen totes les solucions*

En el cas de tenir definides més d'una possible acció reparadora davant una violació concreta, aquest mètode obliga a què el dissenyador defineixi prioritats o pesos entre aquests possibles reparadors. D'aquesta forma, en temps d'execució, al produir-se la violació de la restricció sols s'executarà la reparació (que es pugui aplicar) amb major prioritat. Així doncs, quan una restricció sigui violada només es tindrà en compte una única forma de reparar-la.

Amb el nostre mètode, aquesta situació no es pot donar ja que quan una restricció és violada es consideren tots els possibles reparadors, donant lloc a diferents solucions. En aquest sentit, el nostre enfocament pot restaurar la consistència de la base de dades de formes que el mètode de Gertz no podrà, ja que la solució generada dependrà de com el dissenyador ha definit les prioritats de cada un d'aquests reparadors.

### 5.2.3 Mètode de K. D. Schewe i B. Thalheim [ST96, Sch98]

El treball realitzat per K. D. Schewe i B. Thalheim està orientat al manteniment de restriccions d'integritat d'una base de dades relacional. El principal esforç d'aquest treball rau a intentar assegurar que els canvis que es realitzen a l'aplicar una transacció T sobre la base de dades, no es vegin anul·lats per l'efecte correctiu de les actualitzacions necessàries per a mantenir les restriccions d'integritat. Problema que, com ja hem vist, comparteixen els dos mètodes analitzats anteriorment.

El mètode de Schewe i Thalheim utilitza regles actives ECA per a reparar les restriccions d'integritat. El propòsit del treball és generar les regles actives necessàries per a restaurar la consistència de la base de dades en cas de violar-se alguna restricció d'integritat i, a la vegada, assegurar que l'execució d'aquestes regles actives no desfaci l'efecte de la transacció inicial T.

Donat un conjunt de restriccions d'integritat, es genera un conjunt de regles actives, que almenys conté una regla activa per a reparar cada una de les restriccions d'integritat considerades. A partir d'aquest conjunt de regles es construeix un hipergraf on es representen els predicats de la base de dades i les regles actives que els actualitzen. Aquest graf s'utilitza per a identificar les seqüències d'execucions de regles (camins crítics) que poden anul·lar l'efecte de la transacció inicial T.

El principal resultat del treball és la identificació de les condicions necessàries i suficients que han de complir les restriccions d'integritat per tal que l'hipergraf generat no contingui cap d'aquests camins crítics i, per tant, assegurar que l'efecte de la transacció T no s'anul·la per l'execució de les regles actives. Aquesta situació es dona quan les restriccions d'integritat són estratificades i per a qualsevol combinació de les regles actives generades. A més a més, si les restriccions només són localment estratificades també es pot assegurar la preservació de l'efecte de la transacció inicial. Però en aquest darrer cas, només un subconjunt de les regles actives asseguren aquesta preservació de l'efecte de T.

Una de les principals diferències entre aquest treball [ST96, Sch98] i el nostre treball rau en el poder expressiu del model de dades utilitzat. Aquest enfocament està restringit al model relacional sense la definició de vistes. A més a més, les restriccions d'integritat han d'estar definides en *Forma Normal Implicativa* amb almenys dos literals i un predicat bàsic a l'esquerra de la implicació. Tampoc es permet la negació. Així doncs, les restriccions següents no poden ser expressades amb el seu llenguatge:

Ic1: $\text{false} \leftarrow Q(x, 4)$

Ic2: $\text{false} \leftarrow \neg S(4)$

Un problema que presenta aquest mètode és que, en certs casos, no pot generar totes les solucions que realment existeixen d'una petició inicial d'actualització.

*No generen totes les solucions*

En certs casos, el conjunt de regles actives que aquest mètode té en compte per a assegurar que no es desfà l'efecte de la transacció inicial és un subconjunt de totes les regles actives associades a les restriccions. Així doncs, en aquests casos, no es podran generar sempre totes les solucions que permeten mantenir la consistència de la base de dades, ja que no es podran considerar totes les alternatives de reparació de cada una de les restriccions d'integritat.

**Exemple 5.10**: Considerem la petició d'inserció del fet Wire(Id1, HB, A, 2, 0) a la base de dades següent (exemple adaptat de l'article [Sch98]):

Tube(Id1, HB, 4),

Wire(Id5, HB, A, 2, 0),

$\text{Wire}(\text{wire\_id}, \text{conn}, \text{w\_type}, \text{volt}, \text{pwr}) \rightarrow \text{Tube}(\text{tube\_id}, \text{conn}, \text{t\_type})$

$\text{Wire}(\text{wire\_id}, \text{conn}, \text{w\_type}, \text{volt}, \text{pwr}) \wedge \text{Tube}(\text{tube\_id}, \text{conn}, \text{t\_type}) \rightarrow \text{wire\_id} \neq \text{tube\_id}$

En aquest exemple, els mètodes [CFPT94, Ger94] poden obtenir la solució següent: $S=\{\iota\text{Wire}(Id1,HB,A,2,0), \delta\text{Tube}(Id1,HB,4), \delta\text{Wire}(Id1,HB,A,2,0), \delta\text{Wire}(Id5,HB,A,2,0)\}$. Com es pot comprovar, aquesta solució és incorrecte ja que no satisfà la petició inicial d'actualització.

En aquest cas, el mètode de Schewe no genera cap solució, ja que el mètode evita obtenir tota solució que no satisfaci la petició inicial. Però, en aquest cas, aquest mètode no pot obtenir la solució $S=\{\iota\text{Wire}(Id1,HB,A,2,0), \delta\text{Tube}(Id1,HB,4), \iota\text{Tube}(Id9,HB,9)\}$ on els valors Id9 i 9 són dos valors dels corresponents dominis. Aquesta solució sí que és obtinguda pel nostre mètode.

□

### 5.2.4 Mètode de N. Bidoit i S. Maabout [BM97, Maa98]

En aquests dos treballs, Bidoit i Maabout presenten la definició de la semàntica que ha de tenir un programa basat en regles actives per a l'actualització de bases de dades. Defineixen el

procés d'actualització com un procés en dues etapes: en la primera, donada una base de dades $\Delta$, una actualització inicial U i un programa d'actualització Pc composat per regles, es generen, a partir de U i Pc, totes les actualitzacions necessàries a aplicar sobre $\Delta$, per a satisfer U. En la segona etapa, aquestes actualitzacions s'apliquen físicament sobre $\Delta$.

Aquest treball defineix un algorisme que genera una única regla activa (tipus CA) per a reparar cada una de les restriccions d'integritat definides a la base de dades. En els casos en que les restriccions tenen literals en comú o comparteixen algun literal del mateix predicat, però de signe oposat, l'algorisme identifica quines són les actualitzacions que poden reparar més d'una restricció d'integritat a la vegada, i són les regles que generen aquestes actualitzacions les que es consideren preferentment per a formar part del programa Pc. Per una altra banda, en els casos en que la inserció i l'esborrat del mateix fet poden ser útils per a reparar diferents restriccions, s'imposen les condicions necessàries per a que no es generin a la vegada les dues actualitzacions, i que només s'activi la regla que genera la inserció o la que genera l'esborrat.

Una particularitat d'aquest treball, que el diferencia de la resta de mètodes analitzats, és que no s'imposa la condició de que la base de dades inicial sigui consistent. En aquest sentit, si la petició d'actualització no es pot satisfer consistentment, l'estat final es correspon al resultant de restaurar la consistència de l'estat inicial.

Aquest mètode té, al nostre entendre, bàsicament una limitació i un problema. En primer lloc, l'algorisme de generació de regles actives restringeix considerablement el tipus de restriccions que pot tractar. Però el principal problema que presenta és que no és capaç de trobar totes les solucions a una petició d'actualització T, ja que només considera una forma de reparar cada restricció d'integritat. A més a més, en certs casos, al mantenir la consistència de la base de dades, es pot desfer l'efecte de la petició inicial d'actualització.

*Limitacions a la definició de les restriccions d'integritat*

Les restriccions d'integritat han d'estar definides en *Forma Normal Implicativa* i únicament per predicats bàsics i predicats avaluables. Però per altra banda, l'algorisme presentat en aquest treball imposa altres limitacions addicionals a la definició de les restriccions d'integritat: les restriccions d'integritat han de tenir almenys un literal positiu en la seva definició i no poden aparèixer dos o més literals del mateix predicat amb mateix signe. Així doncs, restriccions com la següent no poden ser tractades per aquest mètode:

$\leftarrow P(x, y) \wedge P(x, y') \wedge y \neq y'$

$\leftarrow \neg P(0)$

*No generen totes les solucions*

L'algorisme que genera el programa Pc a partir de la base dades només considera un subconjunt de totes les possibles regles que poden reparar les restriccions d'integritat de la base de dades, i per cada restricció d'integritat, només es considera una de les possibles regles

actives que la reparen. Això comporta que no es podran generar totes les solucions existents a una petició d'actualització, i a la vegada, que la solució generada depèn directament de la regla activa o programa considerat.

**Exemple 5.11**: Consideri's la petició d'inserir el fet P (T={+P}) a la següent base de dades:

$\Delta$={Q}

inc $\leftarrow$ P $\wedge$ Q

Un programa Pc generat per aquest mètode és el següent:

Q $\rightarrow$ -P

Un cop aplicada la transacció T sobre la base de dades $\Delta$, obtenim una nova base de dades $\Delta_{user}$ ={P, Q}. Al aplicar-hi el programa Pc, el nou estat de la base de dades $\Delta_{final} = \Delta$ = {Q} que no viola cap restricció d'integritat. Però, com es pot comprovar, aquest programa no és correcte ja que desfà l'efecte de la petició inicial +P.

□

## 5.3 Comparació amb mètodes d'actualització de vistes

En aquesta secció s'analitza un mètode solament orientat a l'actualització de vistes, és a dir, que no té en compte les restriccions d'integritat. Aquest mètode és una adaptació, per a bases de dades deductives, d'un treball [LP92] relacionat amb d'actualització de bases de dades relacionals seguint un enfocament basat en la relació universal.

### 5.3.1 Mètode de D. Laurent, V. Phang Luong i N. Spyratos [LPS93]

El treball presentat per D. Laurent, V. Phang Luong i N. Spyratos [LPS93] proposa un mètode per a l'actualització de vistes en bases de dades deductives sense restriccions d'integritat.

La principal característica d'aquesta proposta, que els diferencia d'altres mètodes, és el fet de traduir la petició d'actualització de forma totalment determinista. Els autors proposen un mecanisme per a actualitzar bases de dades deductives en el que no es requereix la participació de l'usuari durant el procés de traducció, ni per escollir entre les diferents solucions obtingudes, ni per a assignar valors concrets a variables existencials no instanciades.

Per a assolir aquest determinisme, els autors proposen emmagatzemar en la base de dades de forma explícita tots els fets inserits i tots els esborrats, tant si són fets bàsics com derivats. Així doncs, una base de dades deductiva $\Delta$ està definida a partir de tres conjunts: *I_facts* contindrà tots els fets inserits, *D_facts* els esborrats i *Rules* correspon al conjunt de regles de derivació. Les úniques condicions que imposen a la base de dades $\Delta$, són que els dos conjunts de fets siguin disjunts (I_facts $\cap$ D_facts = $\emptyset$) i que les regles de Rules siguin segures.

Aquesta proposta utilitza el mateix mecanisme per actualitzar els fets bàsics i els fets derivats. Si la petició d'actualització és una inserció, cal afegir el fet al conjunt I_facts i eliminar-lo de D_facts (si hi és); si en canvi, es tracta d'un esborrat, cal afegir-lo al conjunt D_facts i eliminar-lo, si cal, del conjunt I_facts. D'aquesta forma, per cada petició d'actualització sols existeix una única solució i, per tant, no cal considerar cap criteri de minimalitat entre solucions.

Un primer inconvenient d'aquesta proposta és que al tenir les actualitzacions de fets derivats explícitament a la base de dades, els mecanismes tradicionals de Datalog per a processar consultes no són directament aplicables. Per això, cal transforma la definició de la base de dades deductiva $\Delta$ en termes de la EDB i la IDB de la manera següent:

- $\Delta$ = (EDB, IDB)

- Definir un predicat auxiliar P*(x) associat a cada predicat derivat P(x)

- EDB = I_facts $\cup$ { P*(x) / P(x) $\in$ D_facts}.

- IDB conté les regles de Rules definides com: $P(x) \leftarrow L_1 \wedge \ldots \wedge L_n \wedge \neg P^*(x)$

El problema principal d'aquest mètode és que en el procés d'actualització de vistes no es tenen en compte, en cap moment, les regles de derivació que defineixen la vista. Aquest fet pot tenir dues conseqüències indesitjables: en primer lloc, pot provocar incoherències entre el resultat d'una consulta i el contingut real de la base de dades, i per altra banda, pot permetre obtenir solucions no vàlides.

*Resultat de consultes incorrectes*

Per a resoldre una consulta referent a un fet derivat amb aquest mètode, cal avaluar la regla de derivació de la IDB associada al predicat derivat. Degut a que el mecanisme d'actualització no ha considerat aquestes regles, en certs casos, el resultat de les consultes pot no correspondre's amb el contingut real de la base de dades. En el següent exemple es pot detectar aquesta situació.

**Exemple 5.12**: Suposem la següent base de dades i la seva reformulació:

I_facts = {P(1)}; D_facts = $\varnothing$; => *EDB = {P(1)};*

Rules = { T(x) $\leftarrow$ P(x); => *IDB = { T(x)* $\leftarrow$ *P(x)* $\wedge \neg$ *T*(x);*

P(x) $\leftarrow$ Q(x)} *P(x)* $\leftarrow$ *Q(x)* $\wedge \neg$ *P*(x)}*

Al preguntar a la base de dades sobre el fet derivat P(1), el resultat és 'cert' ja que el fet P(1) pertany a la EDB. De la mateixa manera, T(1) és cert perquè l'avaluació de regla de la IDB permet deduir aquest fet. En canvi, al preguntar pel fet Q(1), la consulta ens retorna un valor de fals, ja que el fet Q(1) no és present a l'EDB. Però en realitat, en una base de dades deductiva el fet Q(1) hauria de ser cert ja que és la única manera de deduir el fet P(1). Aquesta incoherència

és deguda a que al inserir el fet P(1) a la base de dades no es va tenir en compte la seva regla de derivació i, per tant, no es va inserir el corresponent fet bàsic Q(1).

□

Amb el nostre mètode, les peticions d'actualització de fets derivats es tradueixen sempre considerant les regles de derivació (i regles d'esdeveniment associades), evitant possibles contradiccions entre el contingut de la base de dades extensional i el resultat del processament de les consultes.

*Generació de solucions no vàlides*

Al no tenir en compte les regles de derivació a l'actualitzar fets derivats, aquest mètode pot provocar la obtenció de solucions que no són vàlides i, fins i tot, l'obtenció de solucions incoherents amb la pròpia definició de la vista a actualitzar.

**Exemple 5.13**: Suposem la petició d'esborrat del fet P(A) en la base de dades següent:

$$\text{Rules} = \{P(x) \leftarrow Q(x) \wedge \neg R(x);$$
$$T(x) \leftarrow R(x)\}$$
$$\text{I\_facts} = \{R(A)\} \qquad \text{D\_facts} = \{Q(A)\}$$

Un cop actualitzada la base de dades, els nous conjunts I_facts' i D_facts' contenen el següents fets: I_facts' = {R(A)}i D_facts' = {Q(A), P(A)}.

En aquest exemple, es pot comprovar que el mètode de [LPS93] ha permès esborrar el fet P(A) que no era cert a l'estat antic de la base de dades. De la mateixa manera, davant la petició d'inserir el fet T(A), el mètode de [LPS93] permetria fer aquesta inserció afegint el fet T(A) al conjunt I_facts'. El problema és que el mètode no ha detectat que el fet T(A) ja és cert a l'estat antic de la base de dades al poder-se deduir utilitzant la segona regla de Rules.

En aquest exemple, el nostre mètode no generaria cap solució perquè el nostre mètode no permet esborrar un fet que és fals a l'estat antic de la base de dades, ni permet inserir-ne un que ja és cert.

□

## 5.4 Comparació amb el Mètode dels Esdeveniments [Ten92, TO95].

El mètode presentat en aquesta tesi és una extensió del mètode dels Esdeveniments presentat a [Ten92, TO95]. Dues són les principals aportacions del nostre mètode respecte al Mètode dels Esdeveniments: Per una banda, i a nivell de definició del mètode, el nostre mètode incorpora la modificació com un nou tipus d'actualització bàsica i, a més a més, permet definir quins atributs composen la clau de cada un dels predicats definits a la base de dades. Per l'altra, l'aportació més significativa del nostre mètode respecte el seu precursor [Ten92, TO95] és la incorporació de tècniques per a la millora de l'eficiència del mètode per l'actualització de vistes i el manteniment de les restriccions d'integritat.

En aquesta secció, ens limitarem a comentar les principals diferències entre els dos mètodes respecte a la seva definició i les funcionalitats que aporten. La comparació detallada a nivell d'eficiència es presenta en el capítol 7 d'aquest mateix document.

Com es pot veure a la taula 1.1 de la introducció, el problema i el tipus de base de dades considerat en els dos mètodes és el mateix. És a la petició d'actualització on existeix la diferència més remarcable entre els dos mètodes: el Mètode dels Esdeveniments sols pot gestionar actualitzacions que consisteixen en la inserció o l'esborrat d'un fet de la base de dades, en canvi, en el nostre mètode incorporem la modificació d'un fet com un nou tipus d'actualització bàsic. El Mètode dels Esdeveniments únicament pot simular aquest tipus d'actualització mitjançant la combinació d'un esborrat seguit d'una inserció del mateix fet. Respecte al mecanisme o procés de traducció dels dos mètodes no hi diferències importants: els dos mecanismes estan basats en el procediment de resolució SLDNF amb una alternança entre la derivació constructiva i la derivació de consistència. Òbviament la gestió del nou tipus d'actualització comporta una redefinició del nostre mètode en relació al dels Esdeveniments. Respecte a les solucions obtingudes, aquestes segueixen el mateix criteri de minimalitat. I, com en el cas del Mètode dels Esdeveniments, s'ha demostrat la completesa i correctesa del nostre mètode.

De totes formes, la possibilitat de definir explícitament la clau dels predicats de la base de dades i la incorporació del nou tipus d'actualització, representat per l'esdeveniment de modificació $\mu P(\underline{\mathbf{k}}, \mathbf{x}, \mathbf{x}')$, ha introduït certes diferències entre els dos mètodes que cal comentar. La incorporació del nou esdeveniment i la definició explícita de la clau dels predicats fa que la base de dades augmentada A(D) en els dos mètodes sigui diferent. A més a més, la definició explícita de la clau dels predicats, també ha provocat dues diferències a tenir en compte: per una banda, la forma com es fa el manteniment de les restriccions d'integritat de clau i, per l'altre, el canvi en la semàntica d'una inserció o d'un esborrat entre els dos mètodes. Finalment, la incorporació de la modificació com a nou tipus d'actualització ha provocat que el nostre mètode obtingui certes solucions mínimes que en el Mètode dels Esdeveniments no es consideren mínimes i, per tant, no s'obtenen.

*La base de dades augmentada A(D) és diferent*

El Mètode dels Esdeveniments utilitza la definició de base de dades augmentada presentada a [Oli91]. En canvi, el nostre mètode es basa en la utilització de les regles de la base de dades augmentada definida a [UO92, Urp93].

La primera diferència entre aquestes dues bases de dades augmentades rau en que la segona [UO92, Urp93], a part de incloure les regles de transició i les regles d'esdeveniment d'inserció i d'esborrat (com a [Oli91]) també inclou les regles d'esdeveniment de modificació. Per tal de definir l'esdeveniment de modificació en la base de dades augmentada de [UO92, Urp93] s'ha introduït el concepte de clau, que no era considerat a [Oli91].

Però la principal aportació en la generació de la base de dades augmentada A(D), descrita a [UO92, Urp93], és el gran nombre de simplificacions que es poden aplicar a les regles d'esdeveniment generades tenint en compte únicament la informació semàntica de les claus (primàries i secundàries) dels predicats bàsics i derivats definits a la base de dades.

A part de les dues diferències anteriors, cal tenir en compte que el nostre mètode no utilitza directament les regles tal com estan definides a [UO92, Urp93], sinó que a aquestes regles s'hi apliquen les dues transformacions que ja han estat comentades al capítol 3 d'aquest document: per una banda, l'explicitació dels prerequisits dels esdeveniments que apareixen al cos de les regles d'esdeveniment, i per l'altra, la redefinició de les regles d'esdeveniment d'esborrat i modificació pel cas dels predicats derivats definits per més d'una regla deductiva.

La primera és una transformació menor i el seu objectiu és el tenir explícita, a les regles d'esdeveniment, tota la informació que caldrà considerar del contingut de la base de dades al utilitzar-les en les derivacions constructives i de consistència. Aquesta transformació serà especialment útil en la definició de les tècniques per a millorar l'eficiència del procés d'actualització de vistes.

L'objectiu de la segona transformació és més rellevant a l'hora de comparar el nostre mètode amb el Mètode dels Esdeveniments. En el cas de l'actualització de vistes definides per més d'una regla de derivació (m>1), el Mètode dels Esdeveniments pot generar solucions que no són mínimes, però que en un procés final de filtrat, seran eliminades del conjunt de solucions mínimes obtingudes pel mètode. En canvi, amb la redefinició d'aquestes regles d'esdeveniment (proposada al capítol 3), s'aconsegueix que davant de l'actualització d'aquest tipus de predicats derivats el nostre mètode solament obtingui solucions mínimes i, per tant, no sigui necessari en aquest cas aquest procés final de filtrat.

*Manteniment de les restriccions de clau*

En el Mètode dels Esdeveniments, es considera que la clau d'un predicat P(x, y) està composada per tots els seus atributs. En cas de voler definir explícitament una clau diferent pel predicat P (p. e. x és la clau de P) cal fer-ho mitjançant la següent restricció d'integritat:

$$Ic\_clau\_P(x, y) \leftarrow P(x, y) \wedge P(x, y') \wedge y \neq y'$$

Aquest fet, fa que el nombre de restriccions d'integritat definides a la base de dades augmenti considerablement. I conseqüentment, augmenti l'esforç necessari per a assegurar el manteniment de les restriccions d'integritat.

En el nostre mètode, les restriccions d'integritat de clau no cal definir-les explícitament com a restriccions d'integritat, i la pròpia definició del mètode ens permet assegurar el seu manteniment. Això és gràcies a que, en la generació (i simplificació) de la base de dades augmentada A(D), s'ha considerat la informació semàntica de la clau dels predicats i, per tant, la definició de les regles d'esdeveniment d'un predicat asseguren que no es viola la restricció de

clau d'aquest predicat. A més a més, cada cop que s'inclou un esdeveniment al conjunt T (regla A2 de la formalització del mètode) es comprova que no es violen aquestes restriccions de clau.

*Canvi en la semàntica dels esdeveniments*

Amb la incorporació del concepte de clau d'un predicat, la definició dels esdeveniments d'inserció i d'esborrat ha canviat, i per tant, també ha canviat la seva semàntica.

**Exemple 5.14**: Suposem la base de dades composada pel predicat derivat P, els predicats bàsics R i Q, on les claus d'aquests predicats estan composades per l'atribut 'x':

$R(1, 2)$

$P(\underline{x}, y) \leftarrow R(\underline{x}, y) \wedge \neg Q(\underline{x}, y)$

Observi's que en el cas del Mètode dels Esdeveniments es requereix incloure explícitament les restriccions d'integritat addicionals següents:

$Ic\_clau\_P(x, y) \leftarrow P(x, y) \wedge P(x, y') \wedge y \neq y'$

$Ic\_clau\_R(x, y) \leftarrow R(x, y) \wedge R(x, y') \wedge y \neq y'$

$Ic\_clau\_Q(x, y) \leftarrow Q(x, y) \wedge Q(x, y') \wedge y \neq y'$

La petició d'actualització per a inserir el fet Q(3,4) a la base de dades es representa en els dos mètodes amb l'esdeveniment $\iota Q(3,4)$. Per tant, en aquest cas, aquesta actualització es considera en els dos mètodes com una inserció. El mateix passa amb l'esborrat del fet R(1,2).

En canvi, en el cas de voler afegir a la base de dades el fet R(1,3) els dos mètodes ho representen de formes diferents. El Mètode dels Esdeveniments considera aquesta actualització com una inserció representada per l'esdeveniment $\iota R(1,3)$. Però, es pot comprovar que per tal de no violar la restricció de clau del predicat R, el Mètode dels Esdeveniments determina la necessitat de l'esdeveniment addicional $\delta R(1,2)$. En canvi, amb el nostre mètode aquesta actualització es correspon a una modificació representada per l'esdeveniment $\mu R(1,2,3)$.

Aquest exemple posa de manifest, que els esdeveniments de modificació (per exemple $\mu R(1,2,3)$) han de simular-se en el Mètode dels Esdeveniments mitjançant la combinació d'un esdeveniment d'esborrat i d'un esdeveniment d'inserció ($\delta R(1,2) + \iota R(1,3)$). □

A més a més, en aquest exemple, també es pot veure que una inserció (o esborrat) d'un fet a la base de dades amb el Mètode dels Esdeveniments no sempre es correspon a una inserció (o esborrat) d'aquest fet en el nostre mètode. En determinats casos, aquesta inserció o esborrat es pot correspondre (amb algun esdeveniment addicional determinat internament pel Mètode dels Esdeveniments) a una modificació en el nostre mètode. En aquest sentit, podem dir que els esdeveniments d'inserció i d'esborrat en el nostre mètode són més precisos que els mateixos esdeveniments en el Mètode dels Esdeveniments.

*Obtenció de solucions diferents*

El fet d'incorporar la modificació com un nou tipus d'actualització permet que el nostre mètode obtingui solucions que el Mètode dels Esdeveniments no pot obtenir. Aquest fet és degut a que, encara que els dos mètodes utilitzen el mateix criteri de minimalitat entre solucions, hi ha solucions que en el nostre mètode es consideren mínimes i que en el Mètode dels Esdeveniments no ho són. Així doncs, el nombre total de solucions a una mateixa petició d'actualització pot ser diferent en els dos mètodes.

De fet, tenint en compte la definició de solució mínima i al simular la modificació mitjançant un esborrat seguit d'una inserció, el Mètode dels Esdeveniments, en certs casos, no es capaç d'obtenir solucions mínimes que el nostre mètode sí obté.

**Exemple 5.15**: Suposem la mateixa base de dades de l'exemple 5.14, on la petició d'actualització consisteix en l'esdeveniment $U=\delta P(1, 2)$ i existeix un fet bàsic addicional a la base de dades extensional: $R(1, 2)$.

El Mètode dels Esdeveniments obté, per a la petició U, les dues solucions següents:

$T_1 = \{\delta R(1, 2)\}$ $T_2 = \{\delta Q(1, 3), \iota Q(1, 2)\}$

En canvi, el nostre mètode, obté les quatre solucions següents:

$T_1 = \{\delta R(1, 2)\}$ $T_2 = \{\mu Q(1, 3, 2)\}$

$T_3 = \{\mu R(1, 2, 3)\}$ $T_4 = \{\mu R(1, 2, 4), \mu Q(1, 3, 4)\}$

És fàcilment comprovable que en els dos casos, totes les solucions són mínimes, satisfan la petició inicial d'actualització $\delta P(1, 2)$ i no violen la restricció de clau.

També es pot observar que el Mètode dels Esdeveniments no obté les solucions associades a les solucions $T_3$ i $T_4$ obtingudes pel nostre mètode. Aquestes solucions es correspondrien a les transaccions següents:

$T_3 = \{\delta R(1, 2), \iota R(1, 3)\}$

$T_4 = \{\delta R(1, 2), \iota R(1, 3), \delta Q(1, 3), \iota Q(1, 4)\}$

Observi's que aquestes no es poden considerar solucions mínimes ja que inclouen l'esdeveniment $\delta R(1, 2)$ que és, a la vegada, una solució per sí mateix ($T_1$) i, per tant, són eliminades del conjunt final de solucions pel procés de filtrat del Mètode dels Esdeveniments. □