

UNIVERSITAT POLITÈCNICA DE CATALUNYA
DEPARTAMENT DE LLENGUATGES I SISTEMES INFORMÀTICS

ERNEST TENIENTE i LÓPEZ

**EL MÈTODE DELS ESDEVENIMENTS
PER A L'ACTUALITZACIÓ DE VISTES EN
BASES DE DADES DEDUCTIVES**

TESI DOCTORAL

DIRIGIDA PEL DR. ANTONI OLIVÉ i RAMON

BARCELONA

1992

Memòria presentada per l'Ernest Teniente i López per tal
d'aconseguir el grau de Doctor en Informàtica per la Universitat
Politécnica de Catalunya.

A tu, Albert. Perquè, malgrat tot,
sempre et sentiré al meu costat.

A la Roser, als pares, a en David i a
tots aquells que, d'alguna manera o
altra, ens heu ajudat a superar
aquests moments tan difícils.

Índex

1	Introducció	1
1.1	Descripció del problema	2
1.2	Antecedents en bases de dades relacionals	5
1.2.1	Complement d'una vista	6
1.2.2	Traducció segons el tipus de vista	9
1.2.3	Hipòtesi de la relació universal	14
1.3	Antecedents en bases de dades deductives	16
1.3.1	Mètode de Tomasic	16
1.3.2	Mètode de Decker	17
1.3.3	Mètode de Guessoum i Lloyd	20
1.3.4	Mètode de Kakas i Mancarella	21
1.4	El Mètode dels Esdeveniments	24
2	Bases de dades deductives	27
2.1	Regles deductives	27
2.2	Restriccions d'integritat	28
3	Regles de transició i d'esdeveniment	31
3.1	Esdeveniments	31
3.2	Regles de transició	33
3.3	Regles d'esdeveniment d'inserció	35
3.4	Regles d'esdeveniment d'esborrat	37
3.5	Base de dades augmentada	38
4	Mètode dels Esdeveniments: cas simplificat	41
4.1	Descripció del mètode	41
4.2	Exemples	43
4.3	Formalització del Mètode dels Esdeveniments	49

5	Mètode dels Esdeveniments: cas general	55
5.1	Descripció del mètode	55
5.2	Exemples	57
5.3	Formalització del Mètode dels Esdeveniments	64
6	Correctesa i Completesa	69
6.1	Correctesa	69
6.2	Completesa	73
7	Satisfacció de les restriccions d'integritat	79
7.1	Imposició de les restriccions d'integritat	80
7.2	Comprovació de les restriccions d'integritat	83
8	Característiques addicionals del mètode	85
8.1	Modificacions de vista	85
8.2	Prevenició d'efectes secundaris en altres vistes	87
8.3	Anotació de regles	90
9	Implementació	93
9.1	Aspectes generals de la implementació	94
9.2	Implementació de la derivació constructiva	96
9.3	Implementació de la derivació de consistència	98
10	Comparació amb els mètodes de bases de dades deductives	99
10.1	Mètode de Decker	99
10.1.1	Solucions invalidades a causa de la negació	100
10.1.2	Necessitat d'iteració	102
10.1.3	Satisfacció de les restriccions d'integritat	105
10.1.4	Actualitzacions de vista múltiples i mixtes	105
10.2	Mètode de Guessoum i Lloyd	105
10.3	Mètode de Kakas i Mancarella	106
10.3.1	Treball innecessari	107
10.3.2	Satisfacció de les restriccions d'integritat	109
11	Lligam amb els antecedents en bases de dades relacionals	113
11.1	Complement d'una vista	113
11.2	Traducció segons el tipus de vista	117
11.3	Hipòtesi de la relació universal	119

12	Conclusions i recerca futura	121
	12.1 Conclusions	121
	12.2 Recerca futura	123
	Referències	125
	Annex: Codi de la implementació	131

1. Introducció

La tecnologia de les bases de dades ha evolucionat ràpidament des que els primers sistemes de gestió de bases de dades (SGBD) van aparèixer, a l'entorn de l'any 1960. Actualment, les bases de dades relacionals tenen un paper molt important en els sistemes d'informació de la indústria, la banca i l'administració pública, entre d'altres. Una característica comuna d'aquests sistemes d'informació és que gestionen grans quantitats de dades, però les operacions que realitzen sobre aquestes dades acostumen a ser bastant simples.

Sovint és desitjable integrar en la base de dades una definició del coneixement comú compartit per diferents usuaris per a facilitar el desenvolupament d'aplicacions, per a evitar el múltiple desenvolupament de programes amb funcionalitats similars o bé per a proporcionar capacitat de raonament. Aquest coneixement es pot expressar mitjançant regles deductives que defineixen fets derivats a partir d'altres fets (bàsics i/o derivats). Aquesta és la idea bàsica de les bases de dades deductives. Les bases de dades deductives generalitzen les relacionals perquè inclouen no només fets i restriccions d'integritat sinó també regles deductives. Utilitzant aquestes regles es poden deduir nous fets a partir dels fets que estan emmagatzemats explícitament.

Algunes bases de dades relacionals permeten definir vistes. En realitat, aquesta capacitat té algunes analogies amb la possibilitat de definir fets derivats en bases de dades deductives. No obstant, la facilitat de definició de vistes en els SGBD relacionals actuals no es pot comparar en potència amb les regles deductives com a mecanisme de definició.

Les bases de dades deductives poden gestionar qualsevol tipus d'aplicació, essent especialment aconsellables [GV89, UI188] per a aplicacions de CAD/CAM, gestió intel·ligent de documents, criptografia, planificació de robots, disseny de VLSI, etc. Aquestes aplicacions es caracteritzen per la necessitat de realitzar consultes i modificacions ràpides sobre les dades, com en el cas de les bases de dades relacionals, però també necessiten realitzar operacions més complexes sobre les dades que les que requerien les aplicacions prèvies.

El desenvolupament de les bases de dades deductives comença a finals de la dècada dels 50. A grans trets, la seva història es pot dividir en tres parts, abastant cadascuna d'elles uns 10 anys aproximadament. La primera etapa (1957-1968) comprèn les experiències inicials en deducció. S'implementen els primers precursors de bases de dades deductives (com ara el sistema RDF desenvolupat per Rand Corporation). L'any 1965, Robinson [Rob65] publica el principi de resolució i es comença a estudiar l'aplicació d'aquest principi per a implementar d'una manera uniforme la deducció en bases de dades.

La segona etapa (1969-1978) correspon als anys formatius. Es proposa el concepte de programació lògica i s'implementa mitjançant el llenguatge PROLOG. Van Emden i Kowalski [vEK76] desenvolupen els fonaments de la programació lògica i se celebra a Toulouse el 1977 el *Workshop on Logic and Databases* que reuneix a tots els experts capdavanters en aquesta àrea.

La tercera etapa (1979-present) es caracteritza pel desenvolupament dels fonaments teòrics de les bases de dades deductives i de la programació lògica. Reiter [Rei84] proposa teories formals per a les bases de dades i reinterpreta la perspectiva tradicional de la teoria de models en termes de la teoria de la demostració. També es clarifica la gestió de la negació en el cos de les regles amb la introducció de les bases de dades estratificades i s'estudia la gestió de la informació disjuntiva en bases de dades disjuntives. Vegeu [Min88, GM92] per a un estudi exhaustiu de les bases de dades deductives.

Les bases de dades deductives haurien de gestionar les actualitzacions de fets bàsics i de fets derivats (vista) uniformement. D'aquesta manera l'usuari no distingiria quin és el tipus d'informació que s'ha d'actualitzar. L'*actualització de vistes* consisteix a determinar com es tradueix una actualització d'una vista en modificacions dels fets bàsics corresponents. Aquesta tesi descriu la teoria, aplicació i implementació d'un nou mètode, anomenat Mètode dels Esdeveniments, per a l'actualització de vistes en bases de dades deductives.

Aquest capítol introductorí comprèn quatre seccions. En la primera es descriu el problema de l'actualització de vistes. En les seccions 1.2 i 1.3 es presenten les diferents solucions proposades fins ara en el camp de les bases de dades relacionals i deductives, respectivament. Finalment, en la darrera secció es presenten els trets generals del Mètode dels Esdeveniments, formalitzats més endavant en els capítols 4 i 5 d'aquesta tesi.

1.1 Descripció del problema

En bases de dades es fa servir el concepte de *vista* o *esquema extern* per a delimitar aquells continguts de la base de dades rellevants per a cada grup d'usuaris. Una vista és una estructura de

dades virtual, derivada mitjançant una funció de definició a partir dels fets bàsics o d'altres vistes. Per tant, l'extensió d'una vista no té una existència independent, sinó que està completament definida per l'aplicació de la funció de definició a l'extensió actual de la base de dades.

Exemple: considerem una base de dades relacional que conté una relació bàsica d'empleats (EMP) amb els atributs empleat, sou i departament, i una altra de departaments (DEPT) amb els atributs departament i director del departament. Suposem que el contingut de la base de dades és el següent:

<table style="width: 100%; border-collapse: collapse;"> <tr> <th style="text-align: left;">EMP</th> <th style="text-align: left;">(Emp,</th> <th style="text-align: left;">Sou,</th> <th style="text-align: left;">Dept)</th> </tr> <tr> <td>Joan</td> <td>12.000</td> <td>Vendes</td> <td></td> </tr> <tr> <td>Toni</td> <td>15.000</td> <td>Vendes</td> <td></td> </tr> <tr> <td>Enric</td> <td>10.000</td> <td>Finances</td> <td></td> </tr> </table>	EMP	(Emp,	Sou,	Dept)	Joan	12.000	Vendes		Toni	15.000	Vendes		Enric	10.000	Finances		<table style="width: 100%; border-collapse: collapse;"> <tr> <th style="text-align: left;">DEPT</th> <th style="text-align: left;">(Dept,</th> <th style="text-align: left;">Dir)</th> </tr> <tr> <td>Vendes</td> <td>Maria</td> <td></td> </tr> <tr> <td>Finances</td> <td>Dolors</td> <td></td> </tr> </table>	DEPT	(Dept,	Dir)	Vendes	Maria		Finances	Dolors	
EMP	(Emp,	Sou,	Dept)																							
Joan	12.000	Vendes																								
Toni	15.000	Vendes																								
Enric	10.000	Finances																								
DEPT	(Dept,	Dir)																								
Vendes	Maria																									
Finances	Dolors																									

Sobre aquesta base de dades definim, mitjançant l'àlgebra relacional, una vista V1 que ens diu qui és el director de cada empleat:

<table style="width: 100%; border-collapse: collapse;"> <tr> <th style="text-align: left;">V1 (Emp, Dir)</th> </tr> <tr> <td>$f_1 = (EMP * DEPT) [Emp, Dir]$</td> </tr> </table>	V1 (Emp, Dir)	$f_1 = (EMP * DEPT) [Emp, Dir]$	<table style="width: 100%; border-collapse: collapse;"> <tr> <th style="text-align: left;">V1 (Emp, Dir)</th> </tr> <tr> <td>Joan Maria</td> </tr> <tr> <td>Toni Maria</td> </tr> <tr> <td>Enric Dolors</td> </tr> </table>	V1 (Emp, Dir)	Joan Maria	Toni Maria	Enric Dolors
V1 (Emp, Dir)							
$f_1 = (EMP * DEPT) [Emp, Dir]$							
V1 (Emp, Dir)							
Joan Maria							
Toni Maria							
Enric Dolors							

Els avantatges principals de les vistes són els següents:

a) *Simplifiquen* la interfície amb l'usuari; permeten que aquest ignori les dades que no li interessin. En l'exemple anterior, l'usuari que interaccioni amb la vista V1 no s'ha de preocupar dels sous ni dels departaments dels diferents empleats.

b) Afavoreixen la *independència lògica de les dades*; fan possible que es pugui variar l'estructura lògica de la base de dades sense necessitat d'efectuar els canvis corresponents en els programes.

Per exemple, suposem que, per algun motiu determinat, la relació bàsica d'empleats ha de ser substituïda per dues de les seves projeccions, p.ex. R1(Emp, Dept) i R2(Emp, Sou), de tal manera que la relació bàsica EMP es pot reconstruir a partir d'aquestes. En aquest cas, definint la vista EMP com una join de les relacions bàsiques R1 i R2, no caldria modificar les aplicacions que accedeixen a l'antiga relació bàsica EMP. Aquest avantatge no es pot assolir si la base de dades no disposa de la capacitat per a definir i manipular vistes.

c) Proporcionen una mesura de *protecció*; impedeixen que l'usuari accedeixi a dades externes a la seva vista. En l'exemple anterior, l'usuari que treballi amb la vista V1 no té accés al sou dels diferents empleats de l'empresa.

Les operacions que s'efectuen sobre una vista no es poden realitzar directament, sinó que s'han de traduir a operacions equivalents sobre l'estructura de la base de dades. En un entorn ideal, l'usuari no hauria de distingir entre les relacions bàsiques i les vistes i, per tant, els llenguatges de consulta i els de manipulació s'haurien d'aplicar de la mateixa manera en ambdós casos.

Els usuaris interaccionen amb les vistes mitjançant consultes i actualitzacions. Les consultes de vistes no recursives no presenten cap problema conceptual especial. Una consulta sobre la vista es processa component-la amb la definició de vista i s'obté així una consulta equivalent expressada en termes de les relacions bàsiques [Sto75]. En canvi, el cas de les vistes recursives encara no està completament resolt. Vegeu [BR86] per a una síntesi dels diferents mètodes proposats en aquest sentit.

De manera similar a les consultes, qualsevol petició d'actualització de vista haurà de ser traduïda a modificacions dels fets bàsics corresponents. Un cop aplicades aquestes modificacions, el nou estat de la base de dades induirà un nou estat de la vista. Es pretén aconseguir que el nou estat es correspongui al màxim amb la realització de la petició directament sobre la vista original. Aquest procés es pot descriure mitjançant el diagrama següent [Kel85]:

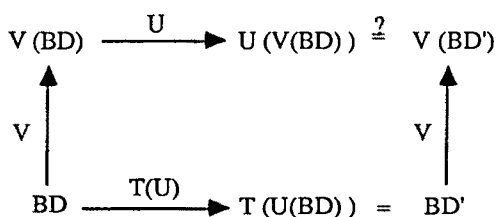


Figura 1.1

En aquest diagrama, BD és una base de dades qualsevol i V una funció de definició de vista. Quan l'usuari especifica una actualització U sobre la vista $V(BD)$, aquesta es tradueix en un conjunt de modificacions dels fets bàsics $T(U)$. Aquestes modificacions generen BD' quan s'apliquen a la base de dades actual. Aleshores, el nou estat de la vista és $V(BD')$. Observi's que, en alguns casos, la vista podria no canviar d'acord amb la petició de l'usuari. Aleshores, $V(BD')$ seria diferent d' $U(V(BD))$.

El processament de les actualitzacions de vista planteja alguns problemes importants:

a) En general, la modificació de la base de dades que satisfà una petició d'actualització de vista no és única. Consideri's una altra vegada l'exemple anterior. Per a aconseguir que la Dolors deixi de ser la directora de l'Enric, es pot esborrar el fet que l'Enric és un empleat del departament de

Finances (EMP (Enric,Finances)) o bé que la Dolors és la directora d'aquest departament (DEPT (Finances,Dolors)).

b) Una modificació de la base de dades pot produir *efectes secundaris* en la vista. No es produiran efectes secundaris si $U(V(BD)) = V(BD')$, és a dir, si la vista ha canviat exactament d'acord amb la petició de l'usuari. Els efectes secundaris són indesitjables i s'haurien d'evitar sempre que es pugui. En l'exemple anterior, per aconseguir que la Maria deixi de ser la directora d'en Joan (és a dir per a esborrar el fet vista V1 (Joan,Maria)) es pot esborrar el tuple DEPT (Vendes,Maria). Malauradament, aquest esborrat també provocaria que la Maria deixés de ser la directora d'en Toni, i aquesta podria no ser la nostra intenció original.

c) Quan es modifica una base de dades que ha de satisfer restriccions d'integritat es necessita algun mecanisme addicional per a comprovar que la base de dades actualitzada també satisfà aquestes restriccions.

El problema de l'actualització de vistes consisteix a determinar com s'ha de traduir apropiadament una petició d'actualització de vista en un conjunt de modificacions dels fets bàsics corresponents. Esperem que la necessitat de realitzar actualitzacions de vista hagi quedat prou justificada pels arguments donats en aquesta secció. Malgrat aquest fet, convé remarcar que a la proposició 3.2 del "Third Generation Database System Manifesto" [CADF90] es diu literalment: "Les vistes actualitzables són essencials". La qual cosa reforça encara més aquesta necessitat.

S'han proposat dos enfocaments principals per a resoldre aquest problema. El primer suggereix que les vistes es tractin com *tipus abstractes de dades* [TFC83, FC85, MW88, SM89], de tal manera que la definició de la vista inclogui una descripció del conjunt d'actualitzacions de vista permeses, conjuntament amb la seva traducció.

El segon enfocament consisteix a definir un procediment general de traducció (un *traductor*). Les entrades del traductor són la definició de la vista, l'actualització de vista demanada i l'estat actual de la base de dades. La sortida del traductor són les modificacions de la base de dades que satisfan la petició. El Mètode dels Esdeveniments presentat en aquesta tesi segueix aquest enfocament.

1.2 Antecedents en bases de dades relacionals

En aquesta secció presentem una síntesi d'alguns mètodes proposats per a l'actualització de vistes en bases de dades relacionals [Ban79, Spy80, BS81, KU84, CP84, Kel87, FSS79, CA79, Mas84, Kel85, Kel86a, Kel86b, Dat86, SLW88, LS91, BV88, Lan90]. Hem cregut convenient presentar

només aquells mètodes que segueixen l'enfocament dels traductors automàtics perquè aquest és l'enfocament adoptat pel Mètode dels Esdeveniments. Vegeu [Ten92] per a un estudi complet de les solucions proposades en bases de dades relacionals.

Com ja hem esmentat anteriorment, l'enfocament dels Traductors Automàtics consisteix a definir un *traductor* que a partir de la definició de la vista, l'actualització de vista demanada i l'estat actual de la base de dades obté automàticament el conjunt de modificacions de les relacions bàsiques que satisfan la petició. Cal tenir en compte que, en general, la traducció que satisfà una actualització de vista no és única i, per tant, es necessita algun mecanisme per a escollir la traducció més adient.

Els mètodes que segueixen l'enfocament dels Traductors Automàtics en bases de dades relacionals es poden classificar de la manera següent:

- Complement d'una vista: [Ban79, Spy80, BS81, KU84, CP84, Kel87]
- Traducció segons el tipus de vista: [FSS79, CA79, Mas84, Kel85, Kel86a, Kel86b, Dat86, SLW88, LS91]
- Hipòtesi de la relació universal: [BV88, Lan90]

Els mètodes corresponents a cadascuna d'aquestes classes es descriuen a les seccions 1.2.1, 1.2.2 i 1.2.3, respectivament.

1.2.1 Complement d'una vista

Bancilhon i Spyratos [Ban79, Spy80, BS81] proposen un mètode molt elegant per a traduir actualitzacions de vista. A cada vista se li associa un *complement* que descriu la informació no visible en la vista. Intuïtivament, un complement d'una vista és la informació addicional que cal donar conjuntament amb la vista per a poder calcular tota la base de dades. Aleshores, una actualització de vista es tradueix de tal manera que només es canvia la vista, mentre que el seu complement es manté invariant.

Què s'entén per *informació no visible en una vista* ? Consideri's per exemple una base de dades que conté una relació bàsica d'empleats EMP amb els atributs nom de l'empleat i departament al qual està assignat i una altra de departaments DEPT amb els atributs nom de departament i director del departament. Suposem que el contingut d'aquestes relacions és el següent:

EMP (Emp, Dept)	DEPT (Dept, Dir)
Joan Vendes	Vendes Maria
Toni Vendes	Finances Dolors
Enric Finances	

Sobre aquesta base de dades es defineixen dues vistes: V1 i V2. Un fet vista V1(e,d) diu que l'empleat e està assignat al departament d, mentre que la vista V2 ens dóna informació de qui és el director de cada departament.

V1 (Emp, Dept)
 $f_1 = \text{EMP} [\text{Emp}, \text{Dept}]$

V1	(Emp,	Dept)
	Joan	Vendes
	Toni	Vendes
	Enric	Finances

V2 (Emp, Dir)
 $f_2 = (\text{EMP} * \text{DEPT}) [\text{Emp}, \text{Dir}]$

V2	(Emp,	Dir)
	Joan	Maria
	Toni	Maria
	Enric	Dolors

La vista V1 s'ha definit restringint la nostra atenció a la relació bàsica d'empleats. Aleshores, un tuple d'aquesta vista es pot actualitzar fàcilment mantenint el contingut de la relació DEPT invariant. En aquest cas, la noció d'informació no visible en V1 és bastant simple ja que la base de dades es pot descompondre en dues parts independents. Malauradament, en el cas general serà molt més difícil definir el complement d'una vista. Per exemple, considerem la vista V2. És difícil definir exactament quina és la informació no visible en aquesta vista. La relació bàsica EMP podria ser-ne el complement, però també ho podria ser DEPT.

A més, interessa definir complements *mínims* d'una vista. Diem que un complement és mínim si no existeix cap altra vista que proporciona menys informació i que també és un complement. A [Ban79] es demostra que una vista pot tenir diversos complements mínims. En l'exemple anterior, tant EMP com DEPT són complements mínims de V2. Bancilhon i Spyrtatos argumenten que l'existència de múltiples complements es correspon a l'existència de diferents polítiques d'actualització de vistes. Lògicament, la tria d'una política dependrà de cada aplicació concreta. L'exemple següent serveix per a constatar aquesta afirmació.

Considerem una altra vegada la base de dades anterior i suposem que l'usuari vol esborrar de la vista V2 que la Dolors és la directora de l'Enric. Si definim EMP com a complement de V2, aquesta petició es traduirà en l'esborrat del tuple DEPT (Finances,Dolors). En canvi, si DEPT fos el complement de V2 escollit, la traducció es reduiria a suprimir el tuple EMP (Enric, Finances). És a dir, en el primer cas la política d'actualització de vistes rau a destituir directors de departaments, mentre que en el segon consisteix a acomiadar els empleats.

Es poden destacar tres resultats importants d'aquesta teoria:

- donada una vista, sempre n'existeix un complement.
- donada una vista i un complement, la traducció que satisfà la petició de modificació i deixa invariant el complement (si existeix) és única.

- a [BS81] es presenta una fórmula que donada una petició d'actualització traduïble permet calcular la seva traducció.

Com els mateixos autors reconeixen, el problema principal d'aquesta teoria rau en la seva implementació pràctica. En aquest sentit, a [CP84] s'estudien els problemes que sorgeixen quan s'intenta aplicar aquest enfocament en el context del model relacional. En concret, s'estudia un àmbit reduït d'aplicació on l'esquema de la base de dades conté una única relació R (hipòtesi de la relació universal) i un conjunt de dependències funcionals i on totes les vistes es defineixen mitjançant projeccions d'aquesta relació.

En aquest entorn es dedueixen les condicions per a conèixer quan dues vistes són complementàries i es presenten les condicions necessàries i suficients per a implementar la inserció, l'esborrat i la modificació d'un tuple en una vista mantenint el complement invariant (considerant que quan l'usuari especifica una vista, també n'especifica el complement). Potser el resultat més important d'aquest treball és la demostració que trobar el complement més petit d'una vista (p.ex. el complement amb el menor nombre d'atributs) és un problema NP-complet.

Una altra crítica que s'ha fet a aquesta teoria [KU84, Kel87] és que es poden refusar peticions d'actualització de vista que tenen traducció pel fet que aquesta no manté el complement invariant. Per tant, en alguns casos pot ser massa restrictiu imposar aquesta condició. En l'exemple següent [Kel87] es presenta un traductor que és força raonable però que no preserva cap complement.

Suposem que tenim la relació AB , amb dos atributs A i B , i la dependència funcional $A \rightarrow B$. Suposem que el domini d' A conté com a mínim un element a_1 , i el domini de B en contingui com a mínim dos, b_1 i b_2 . Definim la vista V com la selecció de tots els tuples d' AB on $B=b_1$. El traductor següent no preserva cap dels possibles complements d'aquesta vista:

Insert tuple $\langle a,b \rangle$: Si existeix un tuple $\langle a,y \rangle$, aleshores reemplaçar $\langle a,y \rangle$ per $\langle a,b \rangle$, altrament, inserir $\langle a,b \rangle$.

Per a resoldre aquest problema, a [KU84] es proposa restringir el tipus dels complements seleccionats quan es desitgi acceptar totes les actualitzacions traduïbles. Per a fer-ho es defineix el concepte de vistes *independents*. Si dues vistes són independents i complementàries es pot mantenir l'estat d'una vista invariant mentre es genera qualsevol estat possible de l'altra vista. Per tant, la tria d'un complement independent permet que totes les actualitzacions que s'expressen contra la vista es puguin traduir a modificacions de la base de dades.

1.2.2 Traducció segons el tipus de vista

Una segona alternativa per a resoldre el problema de l'actualització de vistes en bases de dades relacionals consisteix a definir per cada operador de definició d'una vista quina és la seva traducció. Per cada operador (selecció, projecció, join, etc.) i tipus d'actualització (inserció, esborrat, modificació) es donarà la seva traducció expressada en modificacions de les relacions bàsiques corresponents. Així, per exemple, es pot definir una regla que ens diu quines són les modificacions de les relacions bàsiques que cal efectuar per a satisfer una petició d'esborrat de qualsevol vista definida mitjançant una projecció.

Malgrat diferir en el tipus d'operadors tractats, les diferents propostes realitzades en aquest sentit [FSS79, Mas84, Kel85, Kel86a, Dat86, LS91] es poden resumir en la taula següent. Aquesta taula és aplicable per a actualitzar un únic tuple d'una vista però, exceptuant el cas de la join, també serveix per a actualitzacions múltiples, tenint en compte que en realitza una cada vegada. Per simplificar la presentació, no hem considerat el cas de peticions de modificació [FSS79, Kel85, Dat86] ni els operadors producte cartesià [FSS79, CA79] i intersecció [LS91]. La traducció en aquests casos es pot trobar a les referències esmentades.

	INSERCIONS	ESBORRATS
PROJECCIO: $V = R [X]$ [FSS79], [Mas84], [Kel85], [Dat86], [LS91]	Inserir en R <i>SAP-PROJ</i>	Esborrar de R
SELECCIÓ: $V = R(X \Theta Y)$ [FSS79], [Mas84], [Kel85], [Dat86], [LS91]	Inserir en R o Modificar atribut selecció	Esborrar de R o Modificar atribut selecció <i>SAP-SEL</i>
UNIÓ: $V = R \cup S$ [FSS79], [Mas84], [Kel86a], [LS91]	Inserir en R o Inserir en S o ambdues <i>SAP-UNIO</i>	Esborrar de R i Esborrar de S
DIFERENCIA: $V = R - S$ [Mas84], [LS91]	Inserir en R i Esborrar de S	Esborrar de R o Inserir en S o ambdues <i>SAP-DIF</i>
JOIN: $V = R[X=Y]S$ [Kel85], [Dat86], [LS91]	Inserir en R i Inserir en S	Esborrar de R o Esborrar de S o ambdues <i>SAP-JOIN</i>

Figura 1.2

No tots els autors proposen totes les traduccions expressades en la taula precedent, sinó que en algunes referències consideren casos particulars d'aquestes. Aquest fet serà estudiat detalladament

quan es comenti la traducció de cada operador concret. L'operació Inserir en R (resp. Esborrar de R) es defineix de tal manera que no produeixi cap efecte sobre R en cas que el tuple que s'ha d'inserir (resp. esborrar) pertanyi (resp. no pertanyi) a la base de dades.

Com es pot comprovar, en alguns casos existeixen diverses traduccions que satisfan una actualització de vista. Aquest fet denota una ambigüitat semàntica en el sentit que manca coneixement per a escollir quina és la traducció més adient. Més endavant explicarem les solucions proposades per a resoldre aquest problema. De la mateixa manera que [Mas84, LS91] anomenarem aquests casos *Problemes d'Ambigüitat Semàntica (SAP)*.

Projecció: Per a simplificar, només hem considerat el cas en què els atributs de la vista V inclouen tots els atributs clau de la relació bàsica R. Aleshores, l'esborrat d'un tuple de la vista es tradueix en un esborrat d'un tuple de la relació bàsica subjacent. Una inserció d'un tuple en la vista es tradueix també en la inserció d'un tuple bàsic. En aquest darrer cas, caldrà determinar els valors dels atributs de R que no apareixen en V. La majoria de les propostes realitzades [FSS79, Mas84, Kel85, Dat86] recomanen l'ús del valor NUL per aquests atributs, però també es podrien usar valors per defecte o bé podria ser que el valor dels atributs que no apareixen a la vista depengués dels atributs que són visibles en ella [LS91].

Per tant, en peticions d'inserció ens trobem amb un problema d'ambigüitat semàntica (*SAP-PROJ*) ja que necessitem coneixement addicional per a determinar exactament el valor dels atributs invisibles en la vista.

Selecció: Alguns autors [FSS79, Mas84, Dat86] proposen que les insercions i esborrats d'una vista es tradueixin sempre en insercions i esborrats de les relacions bàsiques que apareixen en la definició de la vista. En canvi, [Kel85] i [LS91] consideren que en alguns casos és millor traduir una actualització d'una vista en una inserció o supressió de la relació bàsica corresponent mentre que en altres és millor traduir-la en una modificació dels valors d'alguns atributs d'aquesta. L'exemple següent [Kel85] ens serveix per a il·lustrar aquest segon raonament.

Suposem que tenim una relació bàsica d'empleats EMP, amb els atributs codi d'empleat, nom, localització i un atribut que indica si l'empleat pertany a l'equip de futbol de l'empresa o no. L'empresa té dues localitzacions: Barcelona i Mataró. Sobre aquesta relació es defineix una vista que ens diu tots els empleats que treballen a Mataró: $V1 = EMP$ (Localització = "Mataró"). Si volem esborrar l'empleat amb codi=17 d'aquesta vista, una traducció raonable consisteix a esborrar l'empleat de la base de dades. Així, haurem traduït un esborrat d'una vista en un esborrat del tuple corresponent.

Suposem que tenim una altra vista que ens dóna els empleats que juguen a l'equip de futbol de l'empresa, $V2 = EMP$ (Futbol = "S"). Sembla poc raonable traduir l'esborrat de l'empleat amb codi=14 d'aquesta vista en un esborrat del tuple de l'empleat (si no és que considerem que el futbol és molt important). Una possibilitat més raonable consisteix a modificar el valor de l'atribut Futbol amb un "No". Per tant, en aquest segon cas un esborrat d'una vista s'ha traduït en una modificació del valor de l'atribut de selecció.

Per tant, en peticions d'esborrat caldrà usar algun tipus de coneixement per a decidir quina opció es més raonable en cada cas (*SAP-SEL*). Observi's que en peticions d'inserció la tria d'una de les dues alternatives depèn de l'estat de la base de dades i, per tant, no es correspon a cap problema d'ambigüitat semàntica.

Unió i Diferència: El raonament que s'ha d'aplicar en les vistes definides mitjançant unió o diferència és similar als anteriors. En aquests casos ens apareixen dos problemes nous d'ambigüitat semàntica, *SAP-UNIÓ* i *SAP-DIF*.

Join: Hi ha quatre condicions possibles en la restricció d'una join del tipus $V = R[X=Y]S$:

- a) $X = \text{Clau}(R)$ i $Y = \text{Clau}(S)$
- b) $X = \text{Clau}(R)$ i $Y \neq \text{Clau}(S)$
- c) $X \neq \text{Clau}(R)$ i $Y = \text{Clau}(S)$
- d) $X \neq \text{Clau}(R)$ i $Y \neq \text{Clau}(S)$

El cas a) es correspon a una join sobre la clau primària de les dues relacions. Els casos b) i c) es corresponen a una join entre la clau primària d'una relació i la forana de l'altra. [Kel85] i [Dat86] tracten els casos a), b) i c), i a [LS91] es tracten també tres subcasos de la quarta condició.

En la taula precedent s'ha expressat només la traducció corresponent als tres primers casos. Una petició d'inserció es tradueix en la inserció d'un tuple de cada relació bàsica que apareix en la definició de la vista. En esborrats d'una vista cal distingir dos casos diferents: en b) i c) l'esborrat de la vista es tradueix en un esborrat d'un tuple de la relació bàsica que participa en la join amb clau forana. En el cas a) es pot escollir entre l'esborrat d'un tuple de qualsevol de les relacions bàsiques que apareix en la definició de la vista o bé l'esborrat d'ambdós. Aquest cas es correspon al problema d'ambigüitat semàntica *SAP-JOIN*.

Exemple 1.1: Aquest exemple serveix per a il·lustrar l'enfocament descrit en aquesta secció. Considerem una base de dades que conté una relació bàsica d'empleats i una de departaments: *EMP* (*Emp*, *Dept*) i *DEPT* (*Dept*, *Dir*), on els atributs subratllats es corresponen a les claus primàries de cada relació. Sobre aquesta base de dades es defineixen dues vistes: $V1 = EMP$ [*Emp*, *Dept*] i $V2 = EMP$ [*Dept = Dept*] *DEPT*. El contingut d'aquesta base de dades és el següent:

EMP	(Emp	Dept)
	Joan	D1
	Pere	D1
	Marc	D2

DEPT	(Dept	Dir)
	D1	Maria
	D2	Anna

V1	(Emp	Dept)
	Joan	D1
	Pere	D1
	Marc	D2

V2	(Emp	Dept	Dir)
	Joan	D1	Maria
	Pere	D1	Maria
	Marc	D2	Anna

Veiem quina és la traducció per cadascuna de les actualitzacions de vista següents:

- Esborrar (V1(Marc, D2)) => Esborrar (EMP(Marc, D2))
- Inserir (V2(Quim,D2,Anna)) => Inserir(EMP(Quim, D2)), Inserir(DEPT(D2,Anna))

Observi's que aquest segon inserir no modifica el contingut de la base de dades ja que el tuple DEPT (D2, Anna) ja hi pertany.

- Esborrar (V2(Joan, D1, Maria)) => Esborrar (EMP(Joan, D1, 1000)).

O sigui, en aquest cas, un esborrat d'una vista definida mitjançant una join es tradueix en un esborrat de la relació bàsica que participa en la join amb clau forana.

Una proposta diferent la podem trobar a [CA79]. En aquest article es descriuen procediments de traducció per a cada tipus d'actualització de vista, independentment de quin hagi estat l'operador usat en la seva definició. Aquests traductors es poden aplicar a vistes definides mitjançant seleccions, projeccions i joins. Malgrat aquesta diferència a l'hora de presentar els traductors, les solucions obtingudes es corresponen a les expressades anteriorment.

Vistes definides mitjançant diversos operadors: Les traduccions expressades en la taula anterior es corresponen a vistes simples. És a dir, a vistes definides mitjançant un únic operador. No obstant, una vista es pot definir usant diversos operadors. En aquest cas, considerant la seqüència d'avaluació de l'expressió que defineix la vista podem identificar una seqüència de vistes V_1, V_2, \dots, V_{k-1} corresponent de manera que, per a cada $i = 2 \dots k$, la vista V_i es deriva a partir de la vista V_{i-1} mitjançant una única operació de definició.

En la figura següent [FSS79] es pot veure com l'aplicació iterativa de les regles de traducció és suficient per a actualitzar vistes definides mitjançant qualsevol nombre d'operacions.

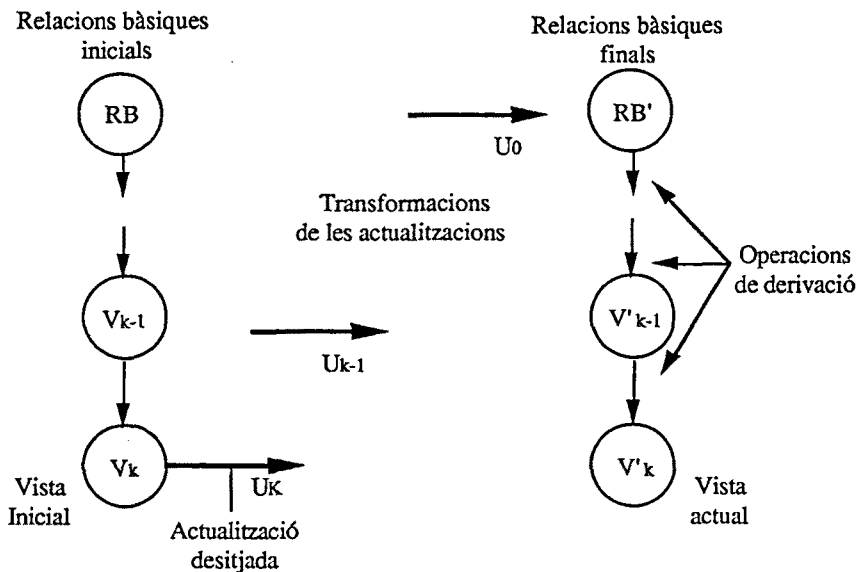


Figura 1.3

La modificació demanada U_k sobre la vista V_k es transforma en una modificació U_{k-1} sobre la vista V_{k-1} d'acord amb l'operació mitjançant la qual V_k es va derivar de V_{k-1} . La modificació sobre V_{k-1} també es transforma en una modificació sobre V_{k-2} , i així successivament, fins que s'obté una modificació U_0 de les relacions bàsiques RB. L'aplicació d' U_0 sobre RB condueix a les noves relacions bàsiques RB'. A partir d'aquestes s'apliquen les diferents operacions de derivació fins a obtenir la vista modificada V'_k .

Problemes d'ambigüitat semàntica: Com hem esmentat anteriorment, un dels problemes més importants de l'actualització de vistes és el de l'existència de múltiples traduccions que satisfan una petició d'actualització. Aquest problema es coneix amb el nom de Problemes d'Ambigüitat Semàntica (SAP). La tria arbitrària d'una solució és inacceptable ja que podria conduir a un estat de la base de dades semànticament inconsistent. Per tant, cal algun mecanisme per a obtenir el coneixement necessari per a escollir la traducció més adient.

S'han realitzat diferents propostes per a resoldre aquest problema. Keller [Kel86a, Kel86b] suggereix que tota la semàntica necessària s'obtingui en el temps de definició de la vista. Això es pot aconseguir mitjançant un diàleg estructurat amb l'administrador de la base de dades. Els interrogants que es plantegen durant el diàleg es basen en la definició de la vista, en la informació estructural de l'esquema de la base de dades i en les preguntes fetes prèviament durant el diàleg. Segons les respostes que es donin s'obindrà un traductor específic o l'altre. Aleshores, usant aquest traductor les actualitzacions de vista es podran traduir unívocament en modificacions de la base de dades.

A diferència de Keller, Masunaga [Mas84] esmenta que, segons quina sigui la naturalesa de les ambigüitats semàntiques, caldrà la interacció amb l'usuari en temps d'execució. És a dir, en alguns casos els SAP es poden resoldre usant les restriccions d'integritat o dependències funcionals de la vista, però en altres es necessita la interacció humana per a escollir la millor traducció.

La darrera proposta realitzada [SLW88,LS91] integra d'alguna manera les dues precedents. En aquesta se suggereix que la informació semàntica necessària per a resoldre les ambigüitats s'obtingui tant en temps de definició com en temps d'execució. Segons aquesta proposta, el coneixement semàntic es classifica en dues classes diferents: *semàntica de la base de dades* i *semàntica de l'aplicació*. La semàntica de la base de dades està formada per les restriccions estructurals, com ara "Codi_Emp és clau d'EMPLEATS", la informació extensional de la base de dades, com ara "hi ha un empleat amb Codi_Emp = 323", i restriccions d'integritat, com ara les dependències funcionals. El coneixement semàntic relatiu a l'aplicació s'anomena semàntica de l'aplicació.

Aquest coneixement el podrà especificar l'administrador de la base de dades en temps de definició o bé l'usuari en temps d'execució. La solució proposada consisteix a intentar resoldre les ambigüitats usant únicament el coneixement obtingut en temps de definició. Si aquest coneixement no és suficient caldrà demanar a l'usuari el coneixement addicional necessari per a escollir la millor solució.

1.2.3 Hipòtesi de la relació universal

En aquest darrer apartat presentem els traductors automàtics relacionals basats en la *hipòtesi de la relació universal* [Sag83], segons la qual l'usuari veu la base de dades com si fos una única relació definida sobre l'univers d'atributs U. Els sistemes de bases de dades basats en aquesta hipòtesi intenten alliberar els usuaris d'haver d'especificar els camins d'accés tant en el nivell físic com en el lògic. Tota la informació sobre l'estructura lògica de la base de dades (p. ex. el seu esquema conceptual) és oculta per l'usuari. Quan aquest consulta o actualitza aquesta relació imaginària definida sobre U, ho fa només en termes dels atributs, sense cap coneixement de quin atribut apareix ni en quina relació.

A [Sag83] es proposa la *instància representativa* com una representació correcta de les dades emmagatzemades en la base de dades. Aquesta instància representativa es pot obtenir automàticament a partir de les relacions bàsiques i de les seves dependències funcionals. Vegeu la referència esmentada per a conèixer exactament el funcionament d'aquest procediment.

La instància representativa satisfà les dependències funcionals incloses en els diferents esquemes de les relacions. Malauradament, no tots els esquemes de bases de dades tenen una única instància representativa. Sagiv imposa una *condició d'unicitat* per a determinar quan es produeix aquest fet.

Quan l'usuari consulta o actualitza una base de dades, ho fa especificant un subconjunt dels atributs de la seva instància representativa. És a dir, les consultes i les actualitzacions són projeccions d'aquesta instància. Per tant, el problema de traduir una actualització especificada sobre la instància representativa en modificacions de les relacions corresponents està relacionat amb el problema de l'actualització de vistes.

El concepte de '*extension join*' (vegeu [Sag83]) s'utilitza per a calcular eficientment projeccions de la instància representativa en esquemes de bases de dades independents (on la consistència local implica la consistència global). Aquest concepte té un paper molt important en els mètodes que descrivim a continuació.

En aquest context, Langerak [Lan90] proposa algorismes per a traduir actualitzacions de vista en esquemes de bases de dades que satisfan la condició d'unicitat. Es consideren insercions, esborrats i modificacions d'un únic tuple d'una vista definida mitjançant una projecció total de la instància representativa. Langerak demostra que els algorismes que ell proposa tradueixen l'actualització d'un tuple en la vista, i que l'algorisme és mínim i afecta només les relacions bàsiques subjacents.

Una altra constatació important d'aquest treball és que les restriccions que presenta l'enfocament del complement d'una vista (descriu a la secció 1.2.1) provoquen que aquest no es pugui aplicar en el context considerat per Langerak.

En una altra proposta [BV88], s'estudia el problema de l'actualització de vistes en aquells esquemes de base de dades que no satisfan la condició d'unicitat. Brosda i Vossen suggereixen que en la fase de disseny de la base de dades s'especifiquin aquelles combinacions d'atributs (*objectes*) que poden participar en una operació d'actualització. Aquests objectes s'inclouen en l'esquema de la base de dades per a ser utilitzats posteriorment. Es permeten dos tipus d'actualització: "Insert Y" i "Delete Y", on Y és un subconjunt dels atributs de la instància representativa.

El procés de traducció es descompon en dues fases diferents: *comprovació* i *execució*. Durant la fase de comprovació, es verifica si Y es correspon a algun dels objectes especificats en la fase de disseny i en peticions d'inserció, caldrà verificar també que existeix la instància representativa de la base de dades actualitzada. Durant l'etapa d'execució, es realitzen les insercions i els esborrats dels tuples corresponents de manera que la instància representativa es pugui calcular per via de '*extension*'

joins'. Els detalls dels procediments de traducció es poden trobar a [BV88]. Convé destacar que l'algorisme d'esborrat no sempre produeix traduccions satisfactòries.

1.3 Antecedents en bases de dades deductives

Un cop estudiades les propostes que s'han fet en el camp relacional, és interessant conèixer quins mètodes s'han proposat fins al moment actual per a l'actualització de vistes en bases de dades deductives. En aquesta secció descriuim aquestes propostes amb el grau suficient de detall per a que puguin ser comparades exhaustivament amb el Mètode dels Esdeveniments. Aquesta comparació es realitza més endavant, en el capítol 10 d'aquesta tesi.

1.3.1 Mètode de Tomasic [Tom88]

El mètode de Tomasic tracta actualitzacions de vista en bases de dades deductives definides, és a dir, en bases de dades que no admeten condicions negatives en el cos de les regles deductives. En aquest sentit, Tomasic proposa procediments basats en la resolució SLD per a obtenir les traduccions mínimes que satisfan una actualització de vista.

L'absència de condicions negatives redueix considerablement el nombre de vistes que es poden definir i, per tant, les limitacions d'aquest mètode són evidents. Per aquest motiu, i ja que el Mètode dels Esdeveniments admet que les definicions de vista continguin condicions negatives, no hem cregut convenient descriure detalladament el procediment emprat per a efectuar la traducció.

Malgrat aquest fet, Tomasic suggereix una tècnica molt interessant per a reduir el nombre de traduccions obtingudes. Aquesta tècnica, anomenada *anotació de regles*, permet que el dissenyador doni informació addicional mitjançant simples anotacions de les regles i els predicats. Aquestes anotacions s'utilitzen posteriorment per a guiar el procés de traducció.

La idea general d'aquesta tècnica consisteix a explorar només aquelles branques que permeten les anotacions. Així, en una base de dades on totes les regles estiguessin anotades de totes les maneres, el traductor obtindria totes les solucions possibles. En canvi, en una base de dades sense cap anotació el traductor sempre refusaria la petició d'actualització ja que seria incapaç d'obtenir cap traducció.

Les anotacions d'esborrat s'expressen subratllant els predicats apropiats i les d'inserció s'expressen mitjançant una fletxa "→" que apunta al cap de la clàusula anotada. Per exemple, considerem una vista P, definida de la manera següent:

$$\begin{array}{ll} \text{RD.1} & P(X) \leftarrow \underline{Q(X)} \wedge R(X) \\ \text{RD.2} & \rightarrow P(X) \leftarrow \underline{S(X)} \end{array}$$

Aquesta vista es defineix mitjançant dues regles deductives. La regla RD.1 té el literal $Q(X)$ anotat per esborrat. La regla RD.2 està anotada per inserció i té el literal $S(X)$ anotat per esborrat. A causa d'aquestes anotacions, la inserció d'un fet vista $P(k)$ es traduirà en afegir($S(k)$), ja que la regla RD.2 és l'única que està anotada per inserció i l'esborrat d'un fet vista $P(k)$ es traduirà en {esborrar($Q(k)$), esborrar ($S(k)$)} ja que ambdós literals estan anotats per esborrat en les regles corresponents. Totes les traduccions possibles d'una inserció mitjançant la clàusula RD.1 o d'esborrats de $R(X)$ seran ignorades.

1.3.2 Mètode de Decker [Dec90a, Dec90b]

El mètode de Decker, així com la resta de mètodes que estudiarem en aquest capítol, tracta l'actualització de vistes en bases de dades deductives que admeten condicions negatives. En aquest mètode, les traduccions que satisfan un esborrat d'una vista s'obtenen a partir d'una refutació SLDNF. En peticions d'inserció, Decker necessita definir el concepte d'*arbres d'actualització de vista* per obtenir les traduccions. En aquesta secció es descriu quin és el procediment de traducció emprat en cada cas.

Esborrat d'una vista: Sigui D una base de dades deductiva i u una petició d'esborrat d'una vista. Les traduccions que satisfan aquesta petició s'obtenen fent que fracassin les diferents refutacions SLDNF de $D \cup \{\leftarrow u\}$. Això es pot aconseguir esborrant un fet usat com a clàusula d'entrada o bé requerint la inserció d'un àtom d'un literal negatiu seleccionat durant la derivació. Observi's que els literals negatius es tracten com a peticions d'inserció subsidiàries.

Exemple 1.2: Suposem que volem esborrar el fet vista P de la base de dades següent:

$$\begin{array}{ll} \text{F.1} & Q(a) \\ \text{F.2} & R(a) \\ \text{DR.1} & P \leftarrow Q(X) \wedge R(X). \end{array}$$

Les traduccions que satisfan aquesta actualització de vista s'obtenen fent que fracassin totes les branques de l'arbre SLDNF de $D \cup \{\leftarrow P\}$. Aquest arbre es mostra en la figura següent:

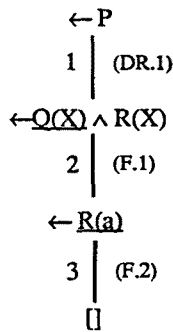


Figura 1.4

Els fets usats com a clàusules d'entrada en aquesta refutació són $Q(a)$ i $R(a)$. Per tant, hi ha dues maneres diferents de fer fracassar l'arbre: $\{\text{esborrar}(Q(a))\}$ o $\{\text{esborrar}(R(a))\}$. Cadascuna d'elles es correspon a una possible traducció de l'actualització de vista demanada.

Inserció en una vista: Sigui D una base de dades deductiva i u una petició d'inserció d'una vista. Per similitud amb el cas anterior, podríem pensar que les traduccions s'obtenen fent que una derivació fracassada de $D \cup \{\leftarrow u\}$ tingui èxit. Això es podria aconseguir afegint una instància de cada literal positiu que apareix en algun objectiu fracassat de la derivació i requerint l'esborrat de l'àtom de tots els literals negatius que apareixen en aquest objectiu.

Exemple 1.3: Suposem que volem inserir el fet vista P en la base de dades següent:

- F.1 $Q(a)$
- F.2 $R(b)$
- DR.1 $P \leftarrow Q(X) \wedge R(X)$.

L'arbre SLDNF de $D \cup \{\leftarrow P\}$ es mostra en la figura 1.5.

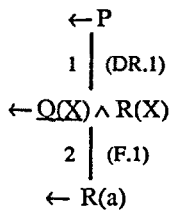


Figura 1.5

Les traduccions que satisfan l'actualització de vista demandada es poden obtenir fent que l'objectiu $\leftarrow R(a)$ tingui èxit. Això s'aconsegueix afegint el fet $R(a)$ a la base de dades. Per tant, s'obté la traducció {afegir $(R(a))$ }.

Malauradament, la funció de selecció utilitzada per la resolució SLDNF pot provocar que no s'obtinguin totes les traduccions que satisfan la inserció de vista demandada. Per exemple, a partir de l'arbre SLDNF de $D \cup \{\leftarrow P\}$, on els literals se seleccionen per la dreta, s'obté la traducció vàlida {afegir $(Q(b))$ }. Aquest arbre es mostra a la figura 1.6.

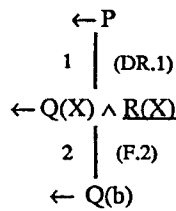


Figura 1.6

Per a poder superar aquesta limitació, és a dir, per a obtenir totes les traduccions que satisfan una petició d'inserció, el mètode de Decker defineix el concepte d'*arbres d'actualització de vista*. Informalment, la idea bàsica d'aquests arbres consisteix a seleccionar i resoldre cada literal de cada objectiu amb cada clàusula d'entrada candidata, encara que en general no cal seleccionar tots els literals d'un objectiu. Un cop assolit l'objectiu fracassat, es fa que tingui èxit de la manera explicada anteriorment. L'arbre d'actualització de vista per a la petició d'inserir P en la base de dades anterior es mostra en la figura 1.7:

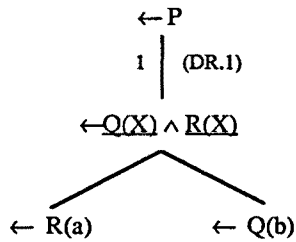


Figura 1.7

Observi's que les dues traduccions que satisfan la inserció de P : {afegir $(R(a))$ } i {afegir $(Q(b))$ } es poden obtenir a partir d'aquest arbre.

No obstant, malgrat que Decker utilitzi els arbres d'actualització de vista, poden existir traduccions que satisfan una actualització de vista però que no són obtingudes pel seu mètode. En el capítol 10 estudiarem quan es produeix aquest fet.

1.3.3 Mètode de Guessoum i Lloyd [GL90a, GL90b]

De la mateixa manera que el mètode de Decker, el mètode de Guessoum i Lloyd utilitza la resolució SLDNF per a obtenir les traduccions que satisfan una actualització de vista. En la primera versió d'aquest mètode [GL90a] es presenten procediments per a esborrar i per a inserir un fet vista en una base de dades deductiva. Aquests procediments tenen la particularitat de ser independents, és a dir, no es criden l'un a l'altre. Per aquest motiu, en general no s'obtenen la majoria de traduccions.

En una versió posterior [GL90b], es generalitzen aquests procediments de tal manera que el d'esborrat crida el d'inserció i a l'inrevés i proporcionen així versions mútuament recursives dels procediments d'actualització. Anem a descriure detalladament el seu funcionament.

Sigui D una base de dades deductiva i u una petició d'esborrat. Les traduccions que satisfan aquesta petició s'obtenen escurçant les branques no fracassades de l'arbre SLDNF de $D \cup \{\leftarrow u\}$. Una branca es pot escurçar suprimint una clàusula usada com a clàusula d'entrada o bé requerint la inserció de l'àtom associat a un literal negatiu seleccionat. Una traducció serà vàlida si un cop modificada la base de dades, l'arbre SLDNF corresponent fracassa finitament.

En peticions d'inserció, les traduccions s'obtenen fent que una derivació fracassada de $D \cup \{\leftarrow u\}$ tingui èxit. Això es pot aconseguir afegint una instància de cada literal positiu que apareix en un objectiu de la derivació i requerint l'esborrat de l'àtom de tots els literals negatius que apareixen en aquest objectiu. Una traducció serà vàlida si un cop modificada la base de dades, hi ha una refutació SLDNF associada a l'actualització de vista demanada.

No és difícil adonar-se que el procediment utilitzat per aquest mètode és molt semblant al que segueix el mètode de Decker. Hi ha, però, tres diferències importants. La primera és que el mètode de Guessoum i Lloyd admet que una traducció estigui formada per modificacions de la base de dades intensional (insercions o esborrats de regles deductives, insercions o supressions de fets vista, etc.).

La segona diferència la trobem en la comprovació que ha d'efectuar el mètode de Guessoum i Lloyd un cop actualitzada la base de dades per a determinar si una solució és vàlida. En general, el cost d'aquesta verificació serà molt elevat ja que gairebé es realitzaran tants accessos a la base de dades com accessos s'hagin efectuat durant el procés de traducció. Malgrat aquest cost addicional, algunes traduccions incorrectes que obté el mètode de Decker són refusades pel mètode de Guessoum i Lloyd gràcies a aquesta comprovació final. Aquest fet es pot corroborar en el capítol 10 d'aquesta tesi.

La tercera diferència és la més important. En peticions d'inserció, Decker defineix els arbres d'actualització de vista per a poder obtenir les traduccions impedides per la funció de selecció utilitzada en la derivació SLDNF. En canvi, Guessoum i Lloyd no proposen cap alternativa a aquest problema. Per tant, generalment aquest darrer mètode no pot obtenir la majoria de traduccions que satisfan una inserció en una vista.

1.3.4 Mètode de Kakas i Mancarella [KM90a, Kak90]

El mètode de Kakas i Mancarella segueix un elegant enfocament abductiu per a tractar l'actualització de vistes en bases de dades deductives. Aquest mètode distingeix clarament dues etapes. En la primera, l'actualització de vista es tradueix mitjançant abducció en diversos conjunts Δ_i . Cada Δ_i és una especificació d'un conjunt de requeriments que la base de dades extensional (EDB) hauria d'acomplir per a satisfer l'actualització de vista demanada. Així, cada conjunt Δ_i es correspon a una solució vàlida. En la segona etapa s'obté una transacció que permet actualitzar Δ_i sobre l'EDB. Aquesta transacció es refereix únicament a les hipòtesis de Δ_i corresponents als predicats bàsics.

En absència de condicions negatives, un *marc abductiu* $\langle T, A, I \rangle$ està format per (vegeu [EK89, KM90a]) una teoria (programa lògic) T, on una col·lecció de predicats de T, A, són designats com a abduïbles, conjuntament amb una teoria d'integritat I. Els predicats abduïbles no tenen cap definició en T però es poden assumir. Donada una observació Q, l'objectiu de l'abducció és *explicar* Q trobant una sèrie d'hipòtesis sobre els abduïbles que, conjuntament amb T, impliquin Q. Intuïtivament, els predicats abduïbles representen la informació incompleta en la teoria i es poden usar en qualsevol moment per a completar-la parcialment.

Com es pot deduir d'aquesta breu descripció, l'abducció i l'actualització de vistes estan molt relacionades. Considerem una base de dades deductiva definida i suposem que la teoria d'integritat és buida. El marc abductiu associat en aquest cas és $\langle IDB, A, \emptyset \rangle$, on IDB es correspon a la base de dades intensional i A és el conjunt de predicats bàsics. Una actualització de vista, per exemple $inserir(P)$, es pot interpretar com la consulta abductiva per a explicar l'observació P en $\langle IDB, A, \emptyset \rangle$. L'exemple següent serveix per a il·lustrar aquestes idees.

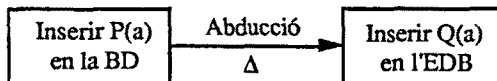
Exemple 1.4: Suposem que volem inserir el fet vista $P(a)$ en la base de dades següent:

IDB: $P(X) \leftarrow Q(X)$

EDB: $Q(c)$

$Q(d)$

En la primera etapa es resol abductivament la consulta $P(a)$, sense tenir en compte el contingut de la base de dades extensional. Raonant endarrera des de $\leftarrow P(a)$ obtenim $Q(a)$, on $Q(a)$ és un abduïble. Aquest abduïble es pot assumir ja que és consistent fer-ho i obtenim així una demostració abductiva $\Delta = \{Q(a)\}$. Per tant, s'ha traduït la petició d'inserir el fet vista $P(a)$ en la base de dades en la petició inserir($Q(a)$) en l'EDB. Esquemàticament:



En la segona etapa s'obté una transacció que permet actualitzar Δ sobre l'EDB. En aquest cas la transacció podria consistir a afegir el fet $Q(a)$ a l'EDB.

Aquest marc abductiu s'ha d'estendre quan apareixen condicions negatives en el cos de les regles deductives ([EK89, KM90a]). Sigui P un programa lògic amb condicions negatives. El marc abductiu associat a P en aquest cas és $\langle P^*, A^*, IC^* \rangle$ on:

- P^* és el programa lògic definit obtingut de P reemplaçant cada literal negatiu, $\neg q(t)$, per un nou literal positiu $q^*(t)$.
- A^* és el conjunt de símbols de predicat, q^* , introduïts en P^* .
- IC^* és el conjunt de denegacions $\leftarrow q(X) \wedge q^*(X)$ i de les sentències $q(X) \vee q^*(X)$ per a tots els símbols de predicat, q^* , introduïts en P^* .

En bases de dades deductives no definides, on P es correspon a la IDB, cal adaptar aquest marc introduint un predicat q^* per tots els predicats de la IDB. A més, es considera que tant els predicats bàsics com la seva negació són abduïbles. Obtenim així el marc abductiu següent:

$$\langle IDB, Ab, IC^* \rangle$$

on Ab és el conjunt de tots els predicats bàsics, més un nou predicat α^* per qualsevol predicat bàsic o vista α . IC^* es defineix de la manera que hem descrit anteriorment per cada predicat α^* . L'exemple següent serveix per a il·lustrar aquest mètode.

Exemple 1.5: Considerem la base de dades deductiva següent, on P i Q són predicats derivats i B és un predicat bàsic:

$$\begin{array}{l} IDB: \quad P(X) \leftarrow \neg Q(X) \\ \quad \quad Q(X) \leftarrow B(X) \\ EDB: \quad B(a) \end{array}$$

el marc abductiu associat a aquesta base de dades és:

$$\begin{array}{l} IDB^*: \quad P(X) \leftarrow Q^*(X) \\ \quad \quad Q(X) \leftarrow B(X) \end{array}$$

Ab: {B, B*, P*, Q*}
 IC*: $\leftarrow Q(X) \wedge Q^*(X)$ $Q(X) \vee Q^*(X)$
 $\leftarrow P(X) \wedge P^*(X)$ $P(X) \vee P^*(X)$
 $\leftarrow B(X) \wedge B^*(X)$ $B(X) \vee B^*(X)$

Suposem que es demana inserir el fet vista P(a) en aquesta base de dades. L'espai de recerca abductiva que es realitza en el primer pas del mètode de Kakas i Mancarella es mostra a la figura següent:

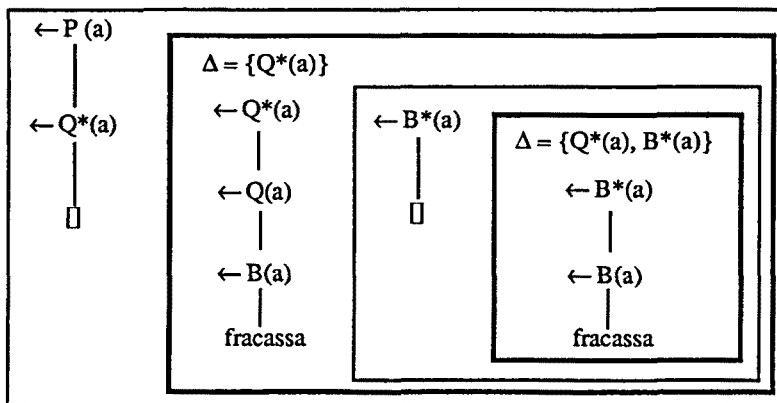


Figura 1.8

Els quadres en negreta mostren la comprovació de la consistència de les noves hipòtesis que s'afegeixen a Δ . Aquesta comprovació es realitza adaptant el Consistency Method [SK88]. Observi's que, en un arbre de consistència, l'èxit a l'hora de demostrar la consistència de l'abduïble requereix que totes les branques fracassin.

En l'arbre principal es veu la necessitat d'incloure l'abduïble $Q^*(a)$ en el conjunt d'hipòtesis Δ per a satisfer l'actualització de vista demanada. Aleshores, cal comprovar la consistència d'aquest abduïble. En el quadre en negreta més gran es verifica que $Q^*(a)$ sigui consistent amb la restricció d'integritat $\leftarrow Q(X) \wedge Q^*(X)$. Perquè aquesta derivació fracassi es requereix que $B(a)$ fracassi. A causa de la restricció d'integritat $B(X) \vee B^*(X)$ això activa la consulta $\leftarrow B^*(a)$, que alhora requereix que $B^*(a)$ sigui afegit a Δ .

Per tant, s'obté la traducció $\Delta = \{Q^*(a), B^*(a)\}$. És a dir, la petició d'actualització de vista original s'ha reduït a fer que l'EDB satisfaci aquest conjunt d'hipòtesis. En la segona etapa del mètode es busca una transacció que permeti actualitzar Δ sobre l'EDB. En aquest cas, això es pot aconseguir esborrant $B(a)$.

Les peticions d'esborrat, com ara esborrar (P), es gestionen transformant-les en peticions d'inserció:

$$\text{esborrar (P)} \Leftrightarrow \text{inserir } (\neg P) \Leftrightarrow \text{inserir (P}^*)$$

Considerem una altra vegada l'exemple anterior. La petició d'esborrar $Q(a)$ d'aquesta base de dades es transformaria en la consulta abductiva $\leftarrow Q^*(a)$. El seu espai de cerca serà el mateix de la figura 1.8 però sense el primer pas de resolució. Per tant, les insercions i els esborrats es tracten de manera uniforme.

Finalment, el mètode de Kakas i Mancarella es pot estendre per a integrar la comprovació de les restriccions d'integritat durant el procés de derivació i s'obtenen així traduccions que satisfan tant l'actualització de vista com la teoria d'integritat de la base de dades.

1.4 El Mètode dels Esdeveniments

En aquesta tesi presentem un nou mètode, anomenat Mètode dels Esdeveniments, per a l'actualització de vistes en bases de dades deductives. Aquest mètode és una aplicació del Model d'Esdeveniments Interns [Oli89] desenvolupat per al disseny de sistemes d'informació a partir de models conceptuals deductius. Altres aplicacions d'aquest model es poden trobar en la comprovació de restriccions d'integritat [Oli91] i en la monitorització de canvis [UO92] en bases de dades deductives, així com en el camp dels models conceptuals deductius, tant per a dissenyar transaccions [San92] com per a raonar a partir d'aquests [CO92].

El Mètode dels Esdeveniments augmenta la base de dades amb un conjunt de regles, anomenades regles de transició i d'esdeveniment, que defineixen explícitament les actualitzacions de vista induïdes per una modificació de la base de dades. Utilitzant aquestes regles, la resolució SLDNF ens permet obtenir totes les traduccions mínimes que satisfan una actualització de vista. Requerim que les solucions siguin mínimes en el sentit que cap subconjunt d'aquestes sigui també una traducció. El problema de com escollir entre diferents traduccions mínimes no és tractat en aquesta tesi. Malgrat aquest fet, les propostes realitzades en el camp relacional [Mas84, Kel86a, Kel86b, SLW88, LS91] es podrien adaptar al nostre cas. Altres comentaris sobre aquest tema es poden trobar a [KM90a, Kak90].

Considerem que les traduccions estan formades únicament per modificacions de la base de dades extensional, és a dir, per insercions i esborrats de fets bàsics. Per aquest motiu, no es tenen en compte canvis en la base de dades intensional (supressió, inserció o modificació de regles

deductives, inserció d'un fet vista, etc). Diversos autors han justificat la conveniència de traduir una actualització de vista d'aquesta manera, vegeu per exemple [Dec90a, KM90a].

Els avantatges principals del Mètode dels Esdeveniments són la seva simplicitat, el tractament uniforme de les insercions i dels esborrats i la incorporació de la comprovació de les restriccions d'integritat en el procés de traducció. Convé destacar també algunes característiques addicionals del nostre mètode com ara la prevenció d'efectes secundaris en altres vistes o el tractament de modificacions del valor d'algun atribut d'un fet vista.

Aquesta tesi s'organitza de la manera següent. En el proper capítol es repassen els conceptes bàsics de les bases de dades deductives. En el capítol 3 es defineix el concepte d'esdeveniment i es descriu el procediment per a derivar les regles de transició i d'esdeveniment. En el capítol 4 es discuteix l'aplicació d'aquestes regles per a l'actualització de vistes considerant un cas simplificat de base de dades deductiva. En el capítol 5 es formalitza el mètode en el cas general i en el capítol 6 es demostra la seva correctesa i completesa. En el capítol 7 es descriu com incorporar la comprovació de les restriccions d'integritat en el nostre mètode. En el capítol 8 es presenten altres característiques del mètode. En el capítol 9 es descriu la seva implementació. En els capítols 10 i 11 es compara el Mètode dels Esdeveniments amb els antecedents proposats fins ara en bases de dades deductives i relacionals, respectivament. El capítol 12 finalitza aquesta tesi amb un sumari i una discussió del treball futur.

Alguns dels materials presentats en aquesta tesi han aparegut a [Ten90], [TO92] i [Ten92].

2. Bases de dades deductives

Una base de dades deductiva D està formada per tres conjunts finits: un conjunt F de fets, un conjunt R de regles deductives i un conjunt I de restriccions d'integritat. Un fet és un àtom de la base. El conjunt de fets s'anomena *base de dades extensional* (EDB) i el conjunt de regles deductives s'anomena *base de dades intensional* (IDB).

Suposarem que els predicats de la base de dades es divideixen en predicats bàsics i predicats vista (també anomenats derivats). Un predicat bàsic apareix només a la base de dades extensional i, eventualment, en el cos de les regles deductives. Un predicat vista apareix només a la base de dades intensional. Qualsevol base de dades deductiva es pot definir d'aquesta manera [BR86].

2.1 Regles deductives

Una regla deductiva és una fórmula del tipus:

$$A \leftarrow L_1 \wedge \dots \wedge L_n \quad \text{amb } n \geq 1$$

on A és un àtom i els L_1, \dots, L_n són literals (o sigui, àtoms o àtoms negats). Totes les variables en A, L_1, \dots, L_n estan universalment quantificades davant la fórmula. 'A' rep el nom de *conclusió* i els L_1, \dots, L_n s'anomenen *condicions*. Els termes de la conclusió i els de les condicions poden ser variables o constants.

Com és habitual, requerim que la base de dades sigui *permissible* (en anglès, *allowed*) abans i després de qualsevol modificació [Llo87]. És a dir, qualsevol variable que ocorre en una regla deductiva té una ocurrència en una condició positiva de la regla. Això assegura que totes les condicions negatives es poden instanciar completament abans de ser avaluades per la regla de negació per fracàs finit.

En aquesta tesi tractem amb bases de dades *estratificades* [ABW88]. Una base de dades és estratificada si el conjunt dels seus símbols de predicat es pot dividir en un conjunt finit de classes, diguem S_0, \dots, S_n , tals que per a cada regla deductiva $P \leftarrow \text{Condicions amb } P \in S_j$:

- (i) si $Q \in S_i$ és el símbol de predicat d'una condició positiva de P , aleshores $i \leq j$, i
- (ii) si $Q \in S_i$ és el símbol de predicat d'una condició negativa de P , aleshores $i < j$.

En el capítol següent donarem una definició alternativa de base de dades estratificada.

2.2 Restriccions d'integritat

Una restricció d'integritat és una fórmula tancada de primer ordre que qualsevol estat de la base de dades ha de satisfer. Nosaltres tractem amb restriccions que tenen forma de *denegació*:

$$\leftarrow L_1 \wedge \dots \wedge L_n \quad \text{amb } n \geq 1$$

on els L_i són literals i totes les variables se suposen universalment quantificades davant la fórmula.

Les restriccions més generals es poden expressar en forma de denegació. Per a fer-ho, primer cal aplicar la transformació en *forma de rang* [Dec89] i, a continuació, utilitzar el procediment descrit a [LIT84]. La transformació en forma de rang és necessària perquè evita que la segona transformació introdueixi *entrebanc* (en anglès, floundering) en alguns casos. Una avaluació s'entrebanca si no es pot continuar el procés de derivació pel fet d'haver assolit un objectiu format únicament per literals negatius no instanciats.

Per motius d'uniformitat, associarem a cada restricció d'integritat un predicat d'inconsistència Icn amb o sense termes (de la mateixa manera que [Kow78]). Així, les restriccions tenen la mateixa forma que les regles deductives i, llavors, les anomenem *regles d'integritat*. Per exemple, la denegació anterior es reescriuria com $Ic1 \leftarrow L_1 \wedge \dots \wedge L_n$. Les restriccions d'integritat també han de ser permissibles.

Exemple 2.1: La base de dades deductiva següent proporciona informació sobre els departaments d'una empresa. Un fet vista $Té_emp(d)$ diu que hi ha algun empleat assignat al departament 'd' i un fet vista $Té_dir(d)$ diu que aquest departament té director. La restricció $Ic1$ indica que una persona no pot estar assignada a un departament que no existeix i $Ic2$ imposa que no hi hagi empleats assignats a un departament que no té director.

Predicats Bàsics

Dept (D)	D és un departament
Assig (E,D)	l'empleat E està assignat al departament D
Dir (E)	E és un directiu

Regles Deductives

$Té_emp(D) \leftarrow Assig(E,D) \wedge \neg Dir(E)$

$Té_dir(D) \leftarrow Assig(E,D) \wedge Dir(E)$

Restriccions d'Integritat

$Ic1(E,D) \leftarrow Assig(E,D) \wedge \neg Dept(D)$

$Ic2(D) \leftarrow Té_emp(D) \wedge \neg Té_dir(D)$

Les restriccions d'integritat que apareixen en aquesta base de dades s'anomenen *estàtiques* ja que s'han de satisfer en qualsevol estat de la base de dades. Les restriccions que inclouen dos estats de la base de dades, anomenades restriccions d'integritat de *transició*, seran considerades més endavant.

3. Regles de transició i d'esdeveniment

En aquest capítol es defineix el concepte d'*esdeveniment* i es descriu el procediment usat per derivar les *regles de transició* i les *regles d'esdeveniment* d'una base de dades. Aquestes regles s'obtenen únicament a partir de les regles deductives i de les restriccions d'integritat i, per tant, són independents dels fets bàsics que estan emmagatzemats a la base de dades. Per aquest motiu, es poden obtenir automàticament en el temps de compilació. En els capítols següents discutirem l'ús d'aquestes regles per a l'actualització de vistes en bases de dades deductives. Els conceptes i terminologia emprats en aquest capítol són una adaptació de [Oli91].

3.1 Esdeveniments

Sigui D una base de dades, U una modificació i D' la base de dades modificada. Diem que U indueix una transició de D (l'estat antic) a D' (l'estat nou). Suposem de moment que U està format per un conjunt no especificat de fets bàsics a afegir i/o a esborrar.

A causa de les regles deductives, U pot induir altres modificacions d'alguns predicats derivats. Sigui P un predicat derivat en D i P' denoti l'avaluació d'aquest mateix predicat en D' . Suposant que el fet $P(k)$ és cert en D , on k és un vector de constants, dos casos són possibles:

- a.1 - $P'(k)$ també és cert en D' (tant $P(k)$ com $P'(k)$ són certs).
- a.2 - $P'(k)$ no és cert en D' ($P(k)$ és cert però $P'(k)$ no ho és).

i suposant que $P'(k)$ és cert en D' , dos casos també són possibles:

- b.1 - $P(k)$ també és cert en D (tant $P(k)$ com $P'(k)$ són certs).
- b.2 - $P(k)$ no és cert en D ($P'(k)$ és cert però $P(k)$ és fals).

En el cas a.2 diem que s'ha produït un esdeveniment d'esborrat durant la transició i el denotem per $\delta P(K)$. En el cas b.2 diem que ha ocorregut un esdeveniment d'inserció i el denotem per $\iota P(K)$.

Així, per exemple, si $Treb$ (Empleat,Dept) és un predicat derivat, $\iota Treb$ (rosar,vendes) denota un esdeveniment d'inserció corresponent al predicat $Treb$: el fet $Treb$ (rosar,vendes) és cert després de la modificació i era fals abans.

Formalment, associem a cada predicat P de la base de dades un esdeveniment d'inserció ιP i un esdeveniment d'esborrat δP , definits com:

$$(1) \quad \forall X (\iota P(X) \leftrightarrow P(X) \wedge \neg P'(X))$$

$$(2) \quad \forall X (\delta P(X) \leftrightarrow P(X) \wedge \neg P'(X))$$

on X és un vector de variables. D'aquestes fórmules s'obtenen les equivalències:

$$(3) \quad \forall X (P'(X) \leftrightarrow (P(X) \wedge \neg \delta P(X)) \vee \iota P(X))$$

$$(4) \quad \forall X (\neg P'(X) \leftrightarrow (\neg P(X) \wedge \neg \iota P(X)) \vee \delta P(X))$$

que relacionen el nou estat amb l'estat antic i els esdeveniments ocorreguts durant la transició.

Si P és un predicat bàsic, els fets ιP i δP representen, respectivament, les insercions i els esborrats de fets bàsics produïts per la modificació. Per tant, a partir d'ara suposarem que U està formada per un conjunt no especificat d'esdeveniments bàsics d'inserció i/o d'esborrat.

Si P és un predicat derivat, aleshores els fets ιP i δP representen, respectivament, les insercions i les supressions de fets derivats induïdes per U .

Si P és un predicat d'inconsistència, aleshores els fets ιP que ocorren durant la transició són violacions de la restricció d'integritat corresponent. Així, per exemple, si una transició indueix el fet $\iota Ic1$, això significa que aquesta transició provoca la violació de la restricció d'integritat $Ic1$. Observi's que, per a predicats d'inconsistència, els fets δP no es poden produir ja que suposem que la base de dades és consistent abans de la modificació.

Exemple 3.1: Consideri's la següent base de dades, on Q , R i S són predicats bàsics.

<i>Fets Bàsics:</i>	$R(a)$ $S(a)$ $S(b)$ $S(c)$
<i>Regles Deductives:</i>	$P(X) \leftarrow Q(X) \wedge R(X)$ $P(X) \leftarrow S(X)$
<i>Restriccions d'Integritat:</i>	$Ic1 \leftarrow P(a)$

Suposem que, en aquest cas, la modificació és el conjunt d'esdeveniments bàsics $U = \{\neg Q(a), \delta S(b), \neg Q(c), \neg R(c)\}$. Els esdeveniments induïts per U en els predicats P i $Ic1$ són $\neg P(a)$, $\delta P(b)$ i $\neg Ic1$. Observi's que $\neg P(c)$ no és induït per U perquè $P(c)$ ja era cert en l'estat anterior de la base de dades i els esdeveniments representen només l'efecte net de la modificació.

3.2 Regles de transició

Sigui P un predicat derivat o d'inconsistència. La definició de P consisteix de les regles de la base de dades que tenen P com a conclusió. Suposem que hi ha m ($m \geq 1$) regles d'aquestes. Pels nostres propòsits, hem de canviar el nom del predicat en la conclusió de les m regles per P_1, \dots, P_m , i afegir el conjunt de clàusules:

$$P \leftarrow P_i \quad i = 1 \dots m$$

Per exemple, si la base de dades conté les regles:

$$\begin{aligned} P(X) &\leftarrow Q(X) \wedge \neg R(X) \\ P(X) &\leftarrow S(X) \wedge T(X) \end{aligned}$$

que defineixen el predicat P , es reescriurà com a:

$$\begin{aligned} P_1(X) &\leftarrow Q(X) \wedge \neg R(X) \\ P_2(X) &\leftarrow S(X) \wedge T(X) \\ P(X) &\leftarrow P_1(X) \\ P(X) &\leftarrow P_2(X) \end{aligned}$$

Consideri's ara una de les regles $P_i(X) \leftarrow L_1 \wedge \dots \wedge L_n$. Quan la regla és avaluada en el nou estat, la seva forma és $P'_i(X) \leftarrow L'_1 \wedge \dots \wedge L'_n$, on L'_r ($r = 1 \dots n$) s'obté reemplaçant el predicat Q de L_r per Q' . Si reemplaçem cada literal del cos per la seva definició equivalent donada a (3) o a (4) obtenim una nova regla, anomenada *regla de transició*, que defineix el predicat P'_i (nou estat) en funció dels predicats de l'estat antic i dels esdeveniments.

Concretament, si L'_r és un literal positiu $Q'_r(X_r)$, apliquem (3) i el substituïm per:

$$(Q_r(X_r) \wedge \neg \delta Q_r(X_r)) \vee \neg Q_r(X_r)$$

i si L'_r és un literal negatiu $\neg Q'_r(X_r)$, apliquem (4) i el substituïm per:

$$(\neg Q_r(X_r) \wedge \neg \neg Q_r(X_r)) \vee \delta Q_r(X_r)$$

Considerem una altra vegada la regla P_1 anterior. En el nou estat de la base de dades la seva forma és $P'_1(X) \leftarrow Q'(X) \wedge \neg R'(X)$. Alèshores, la seva regla de transició és:

$$P'_1(X) \leftarrow [(Q(X) \wedge \neg \delta Q(X)) \vee \iota Q(X)] \wedge [(\neg R(X) \wedge \neg \iota R(X)) \vee \delta R(X)]$$

Serà més fàcil referir-nos a les expressions resultants si denotem per:

$$\begin{aligned} O(L'_r) &= (Q_r(X_r) \wedge \neg \delta Q_r(X_r)) && \text{si } L'_r = Q'_r(X_r) \\ &= (\neg Q_r(X_r) \wedge \neg \iota Q_r(X_r)) && \text{si } L'_r = \neg Q'_r(X_r) \\ N(L'_r) &= \iota Q_r(X_r) && \text{si } L'_r = Q'_r(X_r) \\ &= \delta Q_r(X_r) && \text{si } L'_r = \neg Q'_r(X_r) \end{aligned}$$

Tant $O(L'_r)$ com $N(L'_r)$ expressen condicions per les quals L'_r és cert. $O(L'_r)$ es correspon al cas que L'_r sigui cert perquè L_r ja era cert en l'estat antic (Old en anglès) i no ha estat esborrat, mentre que $N(L'_r)$ es correspon al cas que $N(L'_r)$ és cert perquè és "Nou", induït per la transició, i era fals abans. Observi's que $O(L'_r) \rightarrow L_r$ i que $N(L'_r) \rightarrow \neg L_r$.

Amb aquesta notació, les equivalències (3) i (4) esdevenen:

$$\begin{aligned} (5) \quad &\forall X (P'(X) \leftrightarrow O(P'(X)) \vee N(P'(X))) \\ (6) \quad &\forall X (\neg P'(X) \leftrightarrow O(\neg P'(X)) \vee N(\neg P'(X))) \end{aligned}$$

i aplicant aquestes regles a cadascun dels literals L'_r ($r = 1 \dots n$) obtenim:

$$(7) \quad P'_i(X) \leftarrow \bigwedge_{r=1}^{r=n} [O(L'_r) \vee N(L'_r)]$$

Després de distribuir \wedge sobre \vee , obtenim un conjunt equivalent de 2^{ki} regles de transició, cadascuna d'elles amb la forma:

$$(8) \quad P'_{ij}(X) \leftarrow \bigwedge_{r=1}^{r=n} [O(L'_r) \mid N(L'_r)] \quad \text{amb } j = 1, \dots, 2^{ki}$$

on k_i és el nombre de literals ordinaris en la regla P'_{ij} , i

$$(9) \quad P'_i(X) \leftarrow P'_{ij}(X) \quad j = 1, \dots, 2^{ki}$$

i, per tant,

$$P'_i(X) \leftrightarrow P'_{i,1}(X) \vee \dots \vee P'_{i,2^{ki}}(X)$$

En el conjunt de regles anterior (8), serà útil suposar que la regla corresponent a $j = 1$ és:

$$(10) \quad P'_{i,1}(X) \leftarrow O(L'_1) \wedge \dots \wedge O(L'_n)$$

Exemple 3.2: Sigui P un predicat derivat definit per les regles:

$$P_1(X) \leftarrow T(X)$$

$$P_2(X) \leftarrow Q(X,Y) \wedge \neg R(X)$$

amb:

$$P(X) \leftarrow P_1(X)$$

$$P(X) \leftarrow P_2(X)$$

Les regles de transició per a P_1 i P_2 són:

$$P'_{1,1}(X) \leftarrow T(X) \wedge \neg \delta T(X)$$

$$P'_{1,2}(X) \leftarrow \iota T(X)$$

$$P'_{2,1}(X) \leftarrow Q(X,Y) \wedge \neg \delta Q(X,Y) \wedge \neg R(X) \wedge \neg \iota R(X)$$

$$P'_{2,2}(X) \leftarrow Q(X,Y) \wedge \neg \delta Q(X,Y) \wedge \delta R(X)$$

$$P'_{2,3}(X) \leftarrow \iota Q(X,Y) \wedge \neg R(X) \wedge \neg \iota R(X)$$

$$P'_{2,4}(X) \leftarrow \iota Q(X,Y) \wedge \delta R(X)$$

amb:

$$P'_1(X) \leftarrow P'_{1,j}(X) \quad j = 1, 2$$

$$P'_2(X) \leftarrow P'_{2,j}(X) \quad j = 1, \dots, 4$$

3.3 Regles d'esdeveniment d'inserció

Sigui P un predicat derivat. Les regles d'esdeveniment d'inserció s'han definit a (1) com:

$$\forall X (\iota P(X) \leftrightarrow P'(X) \wedge \neg P(X))$$

Si hi ha m regles pel predicat P, aleshores $P'(X) \leftrightarrow P'_1(X) \vee \dots \vee P'_m(X)$. Per tant tenim que:

$$\iota P(X) \leftarrow P'_i(X) \wedge \neg P(X) \quad \text{per a } i = 1 \dots m$$

Aleshores, si substituïm $P'_i(X)$ per la seva definició equivalent donada a (9) obtenim:

$$(11) \quad \iota P(X) \leftarrow P'_{i,j}(X) \wedge \neg P(X) \quad \text{per a } i = 1 \dots m \text{ i } j = 1 \dots 2^{ki}$$

i com que $P(X) \leftrightarrow P_1(X) \vee \dots \vee P_m(X)$ obtenim:

$$(12) \quad \iota P(X) \leftarrow P'_{i,j}(X) \wedge \neg P_1(X) \wedge \dots \wedge \neg P_m(X) \quad \text{per a } i = 1 \dots m \text{ i } j = 1 \dots 2^{ki}$$

Les regles anteriors (12) s'anomenen *regles d'esdeveniment d'inserció* pel predicat P. Aquestes regles permeten deduir quins fets ιP (insercions induïdes) es produeixen en una transició. Si P és un predicat d'inconsistència, els fets ιP es corresponen a violacions d'una restricció d'integritat.

Les regles corresponents a $j = 1$ (que tenen la forma mostrada a (10)), es poden eliminar de (12). Observi's que aquestes regles no produeixen fets ιP , ja que $P'_{i,1}(X) \rightarrow P_i(X)$. Per tant, el conjunt (12) es redueix a:

$$(13) \quad \iota P(X) \leftarrow P'_{i,j}(X) \wedge \neg P_1(X) \wedge \dots \wedge \neg P_m(X) \quad \text{per } a \ i = 1 \dots m \ i \ j = 2 \dots 2^{ki}$$

A més, hi ha tres importants simplificacions que, quan es poden aplicar, transformen algunes de les regles de (13) en regles equivalents que poden ser avaluades més eficientment.

La primera és la simplificació de la *hipòtesi de consistència*. Es pot aplicar quan P és un predicat d'inconsistència. En aquest cas, els literals $\neg P_k(X)$ ($k=1\dots n$) es poden eliminar de (13) ja que suposem que la base de dades és consistent abans de la modificació i , per això, $P_k(X)$ ha de ser fals, per a tot k i X .

Per a explicar les altres simplificacions, hem d'introduir nova terminologia. Donada una regla $P_i(X) \leftarrow L_1 \wedge \dots \wedge L_n$, distingim dues parts en el seu cos: la part universal i l'existencial. La part universal, denotada per $U(P_i)$, és la conjunció dels literals en el cos les variables dels quals són un subconjunt de X . La part existencial denotada per $E(P_i)$, és la conjunció de literals en els cos que tenen alguna variable que no apareix a X . Aleshores, tenim que $P_i(X) \leftrightarrow U(P_i) \wedge E(P_i)$.

Per exemple, en la regla: $P_1(X,Y) \leftarrow R(X) \wedge Q(X,Y,a) \wedge S(Y,Z) \wedge \neg T(Z)$ tenim:

$$\begin{aligned} U(P_1) &= R(X) \wedge Q(X,Y,a) \\ E(P_1) &= S(Y,Z) \wedge \neg T(Z) \end{aligned}$$

Si $E(P_i)$ no és buit, podem necessitar un predicat auxiliar E_P_i definit per la regla $E_P_i(X) \leftarrow E(P_i)$. En l'exemple anterior, aquest predicat es definiria com:

$$E_P_1(X,Y) \leftarrow S(Y,Z) \wedge \neg T(Z)$$

Les dues simplificacions restants es poden aplicar quan $U(P_i)$ no és buit. Les anomenem *simplificació del fet Nou* i *simplificació del fet Antic*, respectivament. La simplificació del fet Nou es pot aplicar a les regles en (13) on la regla de transició corresponent a $P'_{i,j}(x)$ té algun literal $N(L'_h)$ en la part universal $U(P'_{i,j})$. En aquest cas, podem eliminar el literal $\neg P_i(X)$ i la regla esdevé:

$$(14) \quad \iota P(X) \leftarrow P'_{i,j}(X) \wedge \neg P_1(X) \wedge \dots \wedge \neg P_{i-1}(X) \wedge \neg P_{i+1}(X) \wedge \dots \wedge \neg P_m(X)$$

La simplificació del fet Antic s'aplica a les regles de (13) on la regla de transició corresponent a $P'_{i,j}(X)$ té tots els literals en $U(P'_{i,j})$ de la forma $O(L'_h)$. En aquest cas, podem eliminar $U(P_i)$ de $P_i(X) \leftrightarrow U(P_i) \wedge E(P_i)$ i la regla esdevé:

$$(15) \quad \imath P(X) \leftarrow P'_{ij}(X) \wedge \neg P_1(X) \wedge \dots \wedge \neg P_{i-1}(X) \wedge \neg E_{P_i}(X) \wedge \neg P_{i+1}(X) \wedge \dots \wedge \neg P_m(X)$$

La demostració d'aquestes simplificacions es pot trobar a [Oli91]. L'exemple següent il·lustra la seva aplicació.

Exemple 3.3: Considerem que el predicat P s'ha definit de la mateixa manera que a l'exemple 3.2. Aplicant (13) obtenim les regles d'esdeveniment d'inserció:

$$\begin{aligned} \imath P(X) &\leftarrow P'_{1,2}(X) \wedge \neg P_1(X) \wedge \neg P_2(X) \\ \imath P(X) &\leftarrow P'_{2,j}(X) \wedge \neg P_1(X) \wedge \neg P_2(X) \quad \text{amb } j = 2 \dots 4 \end{aligned}$$

Donat que $U(P'_{1,2})$ té el literal $N(T(X)) = \imath T(X)$, podem aplicar la simplificació del fet Nou a la primera regla i obtenim:

$$\imath P(X) \leftarrow P'_{1,2}(X) \wedge \neg P_2(X)$$

De manera similar, donat que $U(P'_{2,2})$ i $U(P'_{2,4})$ tenen el literal $N(\neg R'(X)) = \delta R(X)$, podem aplicar la mateixa simplificació a les regles corresponents a $P'_{2,2}$ i $P'_{2,4}$, obtenint:

$$\begin{aligned} \imath P(X) &\leftarrow P'_{2,2}(X) \wedge \neg P_1(X) \\ \imath P(X) &\leftarrow P'_{2,4}(X) \wedge \neg P_1(X) \end{aligned}$$

La simplificació del fet Antic es pot usar en la regla corresponent a $P'_{2,3}$, ja que $U(P'_{2,3}) = \neg R(x) \wedge \neg \imath R(X) = O(\neg R(X))$. Obtenim així:

$$\imath P(X) \leftarrow P'_{2,3}(X) \wedge \neg P_1(X) \wedge \neg E_{P_2}(X)$$

amb: $E_{P_2}(X) \leftarrow Q(X, Y)$

3.4 Regles d'esdeveniment d'esborrat

Sigui P un predicat derivat. Les regles d'esdeveniment d'esborrat s'han definit a (2) com:

$$\forall X (\delta P(X) \leftrightarrow P(X) \wedge \neg P'(X))$$

Si hi ha m regles pel predicat P, aleshores tenim:

$$(16) \quad \delta P(X) \leftarrow P_i(X) \wedge \neg P'(X) \quad \text{per a } i = 1 \dots m$$

i, substituint $P'(X)$ per la seva definició equivalent $P'(X) \leftrightarrow P'_1(X) \vee \dots \vee P'_m(X)$, obtenim:

$$(17) \quad \delta P(X) \leftarrow P_i(X) \wedge \neg P'_1(X) \wedge \dots \wedge \neg P'_i(X) \wedge \dots \wedge \neg P'_m(X) \quad \text{per a } i = 1 \dots m$$

Aquestes regles es poden transformar en un conjunt de regles equivalents que poden ser avaluades més eficientment. Sigui $k_{i,u}$ el nombre de literals ordinaris en $U(P_i)$, $k_{i,e}$ el nombre de literals ordinaris en $E(P_i)$ i denotem per $A \setminus B$ A sense B (cert si $A=B$). Aleshores, una regla de (17) és equivalent a les següents $k_{i,u} + k_{i,e}$ regles, una per cadascun dels literals ordinaris $L_{i,j} = [Q_j(X_j) \mid \neg Q_j(X_j)]$ en $U(P_i)$ i en $E(P_i)$:

$$(18) \quad \delta P(X) \leftarrow U(P_i) \setminus L_{i,j} \wedge [\delta Q_j(X_j) \mid \imath Q_j(X_j)] \wedge E(P_i) \wedge \alpha$$

$$\text{per a } i = 1 \dots m, \quad j = 1 \dots k_{i,u}$$

$$(19) \quad \delta P(X) \leftarrow U(P_i) \wedge E(P_i) \setminus L_{i,j} \wedge [\delta Q_j(X_j) \mid \imath Q_j(X_j)] \wedge \neg E_{P'_i}(X) \wedge \alpha$$

$$\text{per a } i = 1 \dots m, \quad j = 1 \dots k_{i,e}$$

on $\alpha \equiv \neg P'_1(X) \wedge \dots \wedge \neg P'_{i-1}(X) \wedge \neg P'_{i+1}(X) \wedge \dots \wedge \neg P'_m(X)$, i on es tria la primera opció de $[\delta Q_j(X_j) \mid \imath Q_j(X_j)]$ si $L_{i,j}$ és positiu i la segona en cas contrari. La demostració d'aquesta transformació es pot trobar a [Oli91].

Exemple 3.4: Considerem una altra vegada el predicat P definit en l'exemple 3.2. Aplicant (18) i (19) obtenim les regles d'esdeveniment d'esborrat:

$$\delta P(X) \leftarrow \delta T(X) \wedge \neg P'_2(X)$$

$$\delta P(X) \leftarrow \imath R(X) \wedge Q(X,Y) \wedge \neg P'_1(X)$$

$$\delta P(X) \leftarrow \neg R(X) \wedge \delta Q(X,Y) \wedge \neg E_{P'_2}(X) \wedge \neg P'_1(X)$$

$$\text{amb: } E_{P'_{2,1}}(X) \leftarrow Q(X,Y) \wedge \neg \delta Q(X,Y)$$

$$E_{P'_{2,2}}(X) \leftarrow \imath Q(X,Y)$$

$$E_{P'_2}(X) \leftarrow E_{P'_{2,j}}(X,Y) \quad j = 1, 2$$

Observi's que no es requereix el literal $\neg P'_1(X)$ en el cos de la primera regla perquè $\delta T(X) \rightarrow \neg T'(X) \rightarrow \neg P'_1(X)$. De manera similar, no es requereix el literal $\neg P'_2(X)$ en la segona regla perquè $\imath R(X) \rightarrow R'(X) \rightarrow \neg P'_2(X)$. En la tercera regla podem eliminar únicament $\neg U(P'_2(X))$.

3.5 Base de dades augmentada

Sigui D una base de dades deductiva. Anomenem *base de dades augmentada*, o $A(D)$, a la base de dades formada per D, les seves regles de transició i les seves regles d'esdeveniment. En el proper capítol discutirem el rol cabdal que té $A(D)$ en el nostre mètode d'actualització de vistes, però abans

analitzarem les propietats d'A(D) respecte les propietats de D. En concret, donat que el nostre mètode està basat en la resolució SLDNF, ens interessen les propietats sintàctiques d'A(D) relacionades amb la completesa d'aquesta resolució.

Aquestes propietats es defineixen en funció del *graf de dependències* de D. Seguirem la terminologia i les definicions donades a [DC89]. Els nodes del graf de dependències són els fets i les regles de D. Per cada parell de nodes F, F', hi ha una fletxa de F' a F si hi ha un àtom A en el cos de F tal que els predicats en A i en la conclusió de F' són els mateixos. La fletxa es marca positivament (resp. negativament) si A és positiu (resp. negatiu) en F.

Si F i F' són dos nodes del graf de dependències, diem que:

- a) F *depèn* de F' si hi ha un camí de F' a F.
- b) F *depèn positivament* (resp. *negativament*) de F' si hi ha un camí de F' a F que no conté cap fletxa negativa (resp., que conté com a mínim una fletxa negativa).
- c) F *depèn en forma parell* (resp. *senar*) de F' si hi ha un camí de F' a F que conté un nombre parell (resp. senar) de fletxes negatives.
- d) F *depèn recursivament d'ell mateix* si hi ha un camí de F a F de longitud superior a 0.

Aquestes definicions s'usen per a caracteritzar les següents propietats d'una base de dades D:

- a) D és *jeràrquica* si cap node del graf de dependències de D depèn recursivament d'ell mateix.
- b) D és *estratificada* si cap node del graf de dependències de D depèn negativament d'ell mateix.
- c) D és *consistent per crida* si cap node del graf de dependències de D depèn en forma senar d'ell mateix.
- d) D és *estricta* si no hi ha cap parell de nodes F, F' en el graf de dependències de D tal que F depèn en forma parell i senar de F'.
- e) D és *parell* si no hi ha cap parell de nodes F, F' en el graf de dependències de D tal que F depèn en forma parell i senar de F' i F' depèn recursivament d'ell mateix.

En general, la resolució SLDNF és incompleta, però hi ha moltes classes de programes lògics (bases de dades deductives) i objectius pels quals és completa. Així, Clark [Cla78] ha demostrat la completesa per bases de dades jeràrquiques i permissibles. Cavedon i Lloyd [CL87] l'han demostrat per bases de dades i objectius que són permissibles, estrictes i estratificats, i Kunen [Kun89] per bases de dades i objectius que són permissibles, estrictes i consistents per crida. Més recentment, Decker i Cavedon [DC89] han demostrat la completesa per bases de dades i objectius que són consistents per crida, parells i cobertes recursivament (una generalització de permissible).

La següent relació entre les propietats de D i les d' $A(D)$ es pot demostrar fàcilment:

- a) Si D és jeràrquica (resp. consistent per crida) aleshores $A(D)$ és jeràrquica (resp. consistent per crida).
- b) Si D és estratificada aleshores $A(D)$ és consistent per crida.
- c) Si $D \cup \{\leftarrow \text{Icn}(X)\}$ és estricta (resp. parell) aleshores $A(D) \cup \{\leftarrow \text{Icn}(X)\}$ també és estricta (resp. parell). El significat d'aquests objectius es veurà més clar en els capítols següents.
- b) Si D és permisible aleshores $A(D)$ també és permisible.

Per tant, aquests resultats ens permeten demostrar que, si les propietats d'una base de dades D són tals que la resolució SLDNF és completa, aleshores la resolució també serà completa per a la base de dades augmentada $A(D)$.

4. Mètode dels Esdeveniments: cas simplificat

El mètode que presentem en aquesta tesi, anomenat Mètode dels Esdeveniments, utilitza les regles de transició i d'esdeveniment per obtenir les traduccions que satisfan una actualització de vista. En el capítol anterior hem definit el procediment usat per derivar aquestes regles. Per tant, ja ens trobem en disposició de descriure el funcionament del nostre mètode.

Per simplificar la presentació, començarem descrivint el Mètode dels Esdeveniments en un cas particular de bases de dades deductives. Concretament, suposarem que totes les variables que apareixen en el cos d'una regla deductiva també ocorren en la seva conclusió. És a dir, i seguint la terminologia definida en el capítol anterior, les regles que no tenen part existencial (on $E(P_i) = \emptyset$). Per exemple, la regla $P(X) \leftarrow Q(X) \wedge R(X,Y) \wedge \neg S(Y)$ no satisfà aquesta condició, ja que $E(P) = R(X,Y) \wedge \neg S(Y)$. En canvi, la regla $P(X,Y) \leftarrow Q(X,Y) \wedge \neg R(Y)$ sí que la satisfà.

Aquest capítol es divideix en tres seccions. A la secció 4.1 es descriu la idea general del nostre mètode. A la secció 4.2 s'estudia el seu funcionament mitjançant alguns exemples. Finalment, a la secció 4.3 es presenta la formalització del cas simplificat del Mètode dels Esdeveniments.

4.1 Descripció del mètode

El Mètode dels Esdeveniments segueix l'enfocament dels traductors automàtics. Les entrades del mètode són la base de dades augmentada i la petició d'actualització de vista. El resultat que s'obté és el conjunt de traduccions mínimes que satisfan l'actualització demanada. Requerim que les solucions siguin mínimes en el sentit que cap subconjunt d'elles sigui també una traducció.

En general, poden existir diverses traduccions mínimes. En aquesta tesi no es tracta el problema d'escollir quina és la traducció més adient en cada cas. Malgrat aquest fet, es podrien adaptar les

propostes realitzades en el camp relacional [Mas84, Kel86a, Kel86b,SLW88, LS91] i en el deductiu [Kak90, KM90a, KM90b] per prendre aquesta decisió.

Nosaltres considerem que les traduccions estan formades únicament per modificacions de la base de dades extensional. És a dir, per insercions i per esborrats de fets bàsics. Per aquest motiu, no tenim en compte canvis en la base de dades intensional (esborrat, inserció o modificació de regles, inserció de fets derivats, etc.). Diversos autors han justificat la conveniència de traduir una actualització de vista només d'aquesta manera: vegeu per exemple [Dec90a, Kak90].

En aquest capítol i en el següent suposarem que les actualitzacions de vista es redueixen a insercions i a esborrats. Més endavant, en el capítol 8, tractarem el cas de peticions de modificació. En el Mètode dels Esdeveniments, una inserció de vista (resp. esborrat) u es correspon a un fet $\iota P(k)$ (resp. $\delta P(k)$), on $P(k)$ és el fet que ha de ser cert (resp. fals) en el nou estat de la base de dades. Per exemple, la inserció del fet vista $P(a)$ es correspon a l'esdeveniment $\iota P(a)$. Una traducció $d'u$, anomenada T , defineix un conjunt d'insercions i/o d'esborrats de fets bàsics tal que, si u és una inserció (resp. esborrat), $P(k)$ és (resp. no és) una conseqüència lògica de la compleció de la base de dades modificada segons T .

Per la pròpia definició del concepte d'esdeveniment, es requereix que si $\iota P(k)$ és una petició d'inserció en una vista (resp. $\delta P(k)$ d'esborrat d'una vista), aleshores $P(k)$ no ha de ser cert en la base de dades actual (resp. $P(k)$ ha de ser cert en la base de dades actual).

Sigui T una traducció formada per un conjunt d'esdeveniments bàsics. T satisfà l'actualització de vista u si, usant la resolució SLDNF, l'objectiu $\{\leftarrow u\}$ té èxit considerant el conjunt d'entrada $A(D) \cup T$. Per tant, T s'obté fent que alguna derivació fracassada d' $A(D) \cup \{\leftarrow u\}$ tingui èxit. Això s'aconsegueix mitjançant la inclusió en el conjunt d'entrada una instància de cada esdeveniment bàsic positiu que ocorre en l'objectiu fracassat.

Si s'assoleix la clàusula buida, aleshores el conjunt de traducció T estarà format per aquests esdeveniments bàsics. Les diferents possibilitats en què una derivació fracassada pot tenir èxit es corresponen a les múltiples traduccions T_i que satisfan l'actualització de vista.

Si no s'obté cap traducció T , aleshores l'actualització de vista no es pot satisfer canviant únicament la base de dades extensional. En aquest cas, s'haurien de realitzar canvis en la base de dades intensional, és a dir, modificacions de les regles deductives i/o de les restriccions d'integritat.

4.2 Exemples

Exemple 4.1: Vista definida mitjançant una única regla deductiva

Sigui D una base de dades deductiva que conté dos predicats bàsics (ED i DM) i un predicat derivat (EDM). Un fet bàsic $ED(e,d)$ indica que l'empleat e està assignat al departament d , mentre que un fet bàsic $DM(d,m)$ ens diu que l'individu m dirigeix el departament d . El predicat vista EDM ens diu donat un empleat, qui és el seu director per cada departament.

Suposem que l'estat actual de la base de dades és el següent:

- F.1 $ED(joan, d1)$
- F.2 $ED(pere, d2)$
- F.3 $DM(d2, maria)$
- DR.1 $EDM(E,D,M) \leftarrow ED(E,D) \wedge DM(D,M)$

Les regles de transició i d'esdeveniment associades a aquesta base de dades són les següents:

- T.1 $EDM'_{1,1}(E,D,M) \leftarrow ED(E,D) \wedge \neg\delta ED(E,D) \wedge DM(D,M) \wedge \neg\delta DM(D,M)$
- T.2 $EDM'_{1,2}(E,D,M) \leftarrow ED(E,D) \wedge \neg\delta ED(E,D) \wedge \imath DM(D,M)$
- T.3 $EDM'_{1,3}(E,D,M) \leftarrow \imath ED(E,D) \wedge DM(D,M) \wedge \neg\delta DM(D,M)$
- T.4 $EDM'_{1,4}(E,D,M) \leftarrow \imath ED(E,D) \wedge \imath DM(D,M)$

- I.1...3 $\imath EDM(E,D,M) \leftarrow EDM'_{1,j}(E,D,M) \quad j = 2 \dots 4$

- D.1 $\delta EDM(E,D,M) \leftarrow ED(E,D) \wedge \delta DM(D,M)$
- D.2 $\delta EDM(E,D,M) \leftarrow \delta ED(E,D) \wedge DM(D,M)$

Eborrat d'una vista: Suposem que volem esborrar el fet $EDM(pere,d2,maria)$. Les traduccions que satisfan aquesta actualització de vista s'obtenen fent que una derivació SLDNF fracassada d' $A(D) \cup \{\leftarrow\delta EDM(pere,d2,maria)\}$ tingui èxit. Això es mostra a la Figura 4.2.1, on les etiquetes encerclades són referències a les regles del mètode definides a la secció 4.3.

En canvi, si que s'obté una traducció a partir de la branca esquerra. Els passos 1, 2 i 3 d'aquesta branca són passos de resolució SLDNF on les clàusules d'A(D) actuen com a clàusules d'entrada. Després del pas 3, s'assoleix l'objectiu $\leftarrow \neg\delta ED(joan,d1) \wedge \iota DM(d1,anna)$. Aquest objectiu serà satisfet si s'inclou l'esdeveniment $\iota DM(d1,anna)$ en el conjunt d'entrada T i es garanteix que no s'hi afegirà $\delta ED(joan,d1)$. Aquestes accions es realitzen en els passos 4 i 5, respectivament.

En el pas 4, l'esdeveniment bàsic seleccionat $\iota DM(d1,anna)$ s'inclou en el conjunt de traducció T. En el pas 5 se selecciona el literal $\neg\delta ED(joan,d1)$. Per a què aquesta branca assoleixi la clàusula buida, $\delta ED(joan,d1)$ ha de fracassar. És a dir, aquest esdeveniment no pot pertànyer al conjunt de traducció. Utilitzem un conjunt de condició C per garantir que, més endavant durant el procés de traducció, $\delta ED(joan,d1)$ no s'inclourà en T. Així, en el pas 5, $\delta ED(joan,d1)$ s'afegeix al conjunt de condició C.

Després del pas 5, el procés finalitza a l'haver-se assolit la clàusula buida. Aleshores, el conjunt de traducció T conté els esdeveniments bàsics que satisfan l'actualització demanada. En aquest cas, s'ha obtingut $T = \{\iota DM(d1,anna)\}$. Dit amb altres paraules, en Joan tindrà com a directora l'Anna en el departament d1 si afegim el fet que l'Anna dirigeix aquest departament. Observi's que aquesta és l'única traducció mínima vàlida possible.

En general, el conjunt de condició C conté aquells esdeveniments bàsics que no poden formar part de la traducció T. La seva finalitat és la d'assegurar que la derivació fracassi realment, és a dir, d'impendir que s'efectuïn accions contradictòries en una mateixa branca. El conjunt C s'obté també durant el procés de derivació i conté els àtoms corresponents als esdeveniments bàsics negatius seleccionats durant aquest.

A causa del significat oposat de T i C, abans d'incloure un esdeveniment bàsic en T (resp. en C) cal comprovar que aquest no pertany al subconjunt de C (resp. de T) ja determinat. Si hi pertany, això indica que s'ha produït una contradicció en la branca actual i, per tant, cap traducció vàlida es pot obtenir a partir d'aquesta. En l'exemple anterior, no es produeix cap contradicció a l'incloure esdeveniments bàsics en T ni en C.

Exemple 4.2: Vista definida mitjançant més d'una regla deductiva

El Mètode dels Esdeveniments tracta de la mateixa manera totes les vistes, independentment del nombre de regles deductives usades en la seva definició. Malgrat aquest fet, hem considerat convenient presentar un exemple d'aquest darrer cas ja que això ens permet il·lustrar aspectes del mètode que no han aparegut anteriorment.

Sigui D una base de dades deductiva que conté per tres predicats bàsics (Est, Cr i Ciu) i un predicat derivat (Dr). El significat dels predicats bàsics i derivat és:

Est (X)	X és estranger
Cr (X)	X té algun registre criminal
Ciu (X)	X és un ciutadà
Dr (X)	X té dret de residència. El poden tenir tots els ciutadans més els estrangers que no tenen cap registre criminal.

Suposem que el contingut actual de la base de dades és el següent:

F.1	Ciu (toni)
DR.1	$Dr(X) \leftarrow Est(X) \wedge \neg Cr(X)$
DR.2	$Dr(X) \leftarrow Ciu(X)$

Algunes regles de transició i d'esdeveniment associades a aquesta base de dades són:

T.1	$Dr'_{1,1}(X) \leftarrow Est(X) \wedge \neg \delta Est(X) \wedge \neg Cr(X) \wedge \neg \iota Cr(X)$
T.2	$Dr'_{1,2}(X) \leftarrow Est(X) \wedge \neg \delta Est(X) \wedge \delta Cr(X)$
T.3	$Dr'_{1,3}(X) \leftarrow \iota Est(X) \wedge \neg Cr(X) \wedge \neg \iota Cr(X)$
T.4	$Dr'_{1,4}(X) \leftarrow \iota Est(X) \wedge \delta Cr(X)$
T.5	$Dr'_{2,1}(X) \leftarrow Ciu(X) \wedge \neg \delta Ciu(X)$
T.6	$Dr'_{2,2}(X) \leftarrow \iota Ciu(X)$
T.7..10	$Dr'_1(X) \leftarrow Dr'_{1,j}(X) \quad j = 1..4$
T.11, 12	$Dr'_2(X) \leftarrow Dr'_{2,j}(X) \quad j = 1, 2$
D.1	$\delta Dr(X) \leftarrow \delta Est(X) \wedge \neg Cr(X) \wedge \neg Dr'_2(X)$
D.2	$\delta Dr(X) \leftarrow Est(X) \wedge \iota Cr(X) \wedge \neg Dr'_2(X)$
D.3	$\delta Dr(X) \leftarrow \delta Ciu(X) \wedge \neg Dr'_1(X)$

Suposem que volem esborrar el fet vista Dr(toni), és a dir, volem que en Toni deixi de tenir dret de residència. Les possibles traduccions que satisfan aquesta petició s'obtenen fent que una derivació fracassada d' $A(D) \cup \{\leftarrow \delta Dr(\text{toni})\}$ tingui èxit. Procedim de la mateixa manera que en l'exemple anterior. Això es mostra a la Figura 4.2.3, on les etiquetes encerclades són referències a les regles del mètode definides a la propera secció.

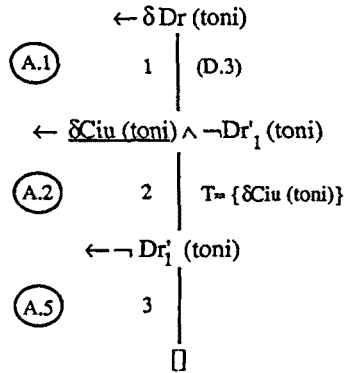


Figura 4.2.3

Seleccionant les clàusules D.1 i D.2 en el pas 1 d'aquesta derivació, obtenim derivacions fracassades a les quals no es pot fer tenir èxit. Per tant, no s'obté cap traducció a partir d'elles (aquestes derivacions no es mostren en la figura anterior).

En canvi, si que es pot aconseguir que la derivació resultant de seleccionar la clàusula D.3 tingui èxit. El pas 1 d'aquesta derivació és un pas de resolució SLDNF on $A(D)$ actua com conjunt d'entrada. En el pas 2, l'esdeveniment bàsic seleccionat s'inclou en el conjunt de traducció T . Després d'aquest pas s'assoleix l'objectiu $\leftarrow \neg Dr'_1(\text{toni})$. Aquest objectiu se satisfarà si l'arbre subsidiari associat al literal $Dr'_1(\text{toni})$ fracassa finitament. A la figura 4.2.4 es mostra una part d'aquest arbre subsidiari:

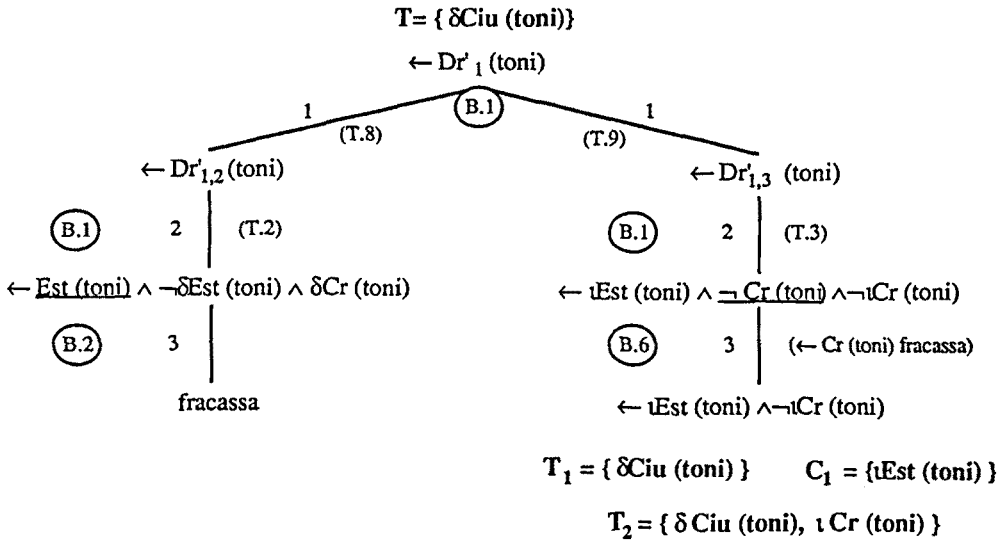


Figura 4.2.4

Els passos 1 i 2 de la branca esquerra són passos de resolució SLDNF on A(D) actua com a conjunt d'entrada. En el pas 3, aquesta derivació fracassa perquè no es pot unificar el literal seleccionat Est(toni) amb cap clàusula de la base de dades.

Per motius similars, la selecció de les clàusules T.6 i T.10 en el pas 1 de l'arbre anterior condueix a derivacions fracassades (aquestes derivacions no es mostren en l'exemple).

Els passos 1 i 2 de la branca dreta són passos de resolució SLDNF on A(D) actua com a conjunt d'entrada. Després d'aquest pas s'assoleix l'objectiu $\leftarrow \neg \text{Est}(\text{toni}) \wedge \neg \text{Cr}(\text{toni})$. Aleshores, cal considerar totes les alternatives per a fer-lo fracassar. Obtindrem tantes traduccions com possibilitats hi hagi de garantir el fracàs d'aquest objectiu. En aquest cas, hi ha dues opcions: incloure $\neg \text{Cr}(\text{toni})$ en el conjunt de traducció T o bé garantir que $\neg \text{Est}(\text{toni})$ no hi serà inclòs.

Considerem la segona possibilitat. La inclusió de l'esdeveniment bàsic $\neg \text{Est}(\text{toni})$ en el conjunt de condició C_1 assegura que $\neg \text{Est}(\text{toni})$ no serà inclòs en T. Aleshores, una traducció possible és $T_1 = \{\delta \text{Ciu}(\text{toni})\}$. Per tant, l'esborrat del fet Ciu(toni) satisfarà la petició d'esborrar Dr(toni). És a dir, l'eliminació de la condició de ciutadà d'en Toni farà que aquest deixi de tenir dret de residència.

L'altra possibilitat es correspon a la inclusió de l'esdeveniment $\neg \text{Cr}(\text{toni})$ en T. Aleshores s'obté la traducció $T_2 = \{\delta \text{Ciu}(\text{toni}), \neg \text{Cr}(\text{toni})\}$. O sigui, l'actualització demanada també se satisfarà si s'esborra el fet Ciu(toni) de la base de dades i s'hi insereix el fet Cr(toni). Observi's que T_2 no és una traducció mínima, ja que existeix un subconjunt d'ella (T_1) que també és una traducció. Com que nosaltres considerem només solucions mínimes, l'única que tindrem en compte és T_1 .

4.3 Formalització del Mètode dels Esdeveniments

Un cop il·lustrat el Mètode dels Esdeveniments mitjançant alguns exemples, en aquesta secció definim formalment el procediment usat per a obtenir les traduccions en el cas particular de bases de dades considerat en aquest capítol. Com hem pogut comprovar en els exemples anteriors, el Mètode dels Esdeveniments és una alternança de dues activitats: 1) satisfer l'actualització de la vista incloent esdeveniments bàsics en el conjunt de traducció i 2) comprovar que les actualitzacions de vista induïdes per aquests esdeveniments bàsics no són contradictòries amb l'actualització demanada. Aquestes dues activitats es realitzen, respectivament, durant les derivacions constructiva i de consistència definides a continuació.

Sigui u una petició d'actualització de vista. T serà una traducció d'u si existeix una derivació constructiva des de $(\leftarrow u \ \emptyset \ \emptyset)$ fins a $(\square T \ C)$. Els esdeveniments bàsics continguts en el conjunt de traducció T es corresponen a les modificacions (insercions i/o esborrats) que cal efectuar en la base

de dades actual per a satisfer l'actualització de la vista. En general, existiran diferents derivacions constructives. Cadascuna es correspon a una traducció diferent. Quan no n'hi ha cap, l'actualització de la vista no es pot satisfer canviant únicament la base de dades extensional.

El Mètode dels Esdeveniments pot obtenir solucions no mínimes. Per tant, un cop generades totes les traduccions T_i , un mecanisme de purga se n'encarregarà d'escollir únicament aquelles traduccions que són mínimes.

El Mètode dels Esdeveniments gestiona de la mateixa manera les peticions d'inserció i les d'esborrat. Per tant, també admet **actualitzacions de vista múltiples** (més d'una actualització alhora) i **mixtes** (incloent peticions d'inserció i d'esborrat) Una actualització múltiple i mixta de la forma:

inserir(P_1) i ... i inserir(P_n) i esborrar(Q_1) i ... i esborrar(Q_m)

es correspon a l'objectiu inicial $\leftarrow \neg P_1 \wedge \dots \wedge \neg P_n \wedge \delta Q_1 \wedge \dots \wedge \delta Q_m$. A partir d'aquest objectiu les traduccions s'obtenen seguint el procediment definit en aquesta secció.

Els predicats que apareixen en la base de dades augmentada $A(D)$ poden ser dels tipus següents (hi incloem exemples de la darrera secció):

- predicats bàsics antics (de l'estat anterior de la base de dades): Est, Ciu, Cr
- predicats derivats antics (de l'estat anterior de la base de dades): Dr
- esdeveniments bàsics: δ Ciu, \neg Cr, δ Est
- esdeveniments derivats: δ Dr, \neg Cr
- predicats nous (del nou estat de la base de dades): $Dr'_{1,1}$, $Dr'_{1,2}$

Els esdeveniments bàsics representen les accions que es poden realitzar sobre els predicats bàsics de la base de dades. Així $\neg T$, essent T un predicat bàsic, representa la inserció de T mentre que δT representa l'esborrat de T. Els esdeveniments derivats representen les actualitzacions induïdes sobre els predicats derivats que es produeixen durant la transició. Els predicats nous representen l'avaluació del predicat del mateix nom en la base de dades modificada, p.ex. P' denota l'avaluació del predicat derivat P en la base de dades actualitzada.

Fins ara, en aquesta tesi hem usat el símbol ' \wedge ' per a expressar una conjunció. A partir d'aquest moment usarem aquest símbol i el símbol ',' indistintament. O sigui, $P \leftarrow L_1, \dots, L_k$ i $P \leftarrow L_1 \wedge \dots \wedge L_k$ expressaran la mateixa regla deductiva.

Per poder definir les derivacions constructiva i de consistència necessitem expressar la convenció següent:

- Sigui G_i un objectiu que té la forma $\leftarrow L_1, \dots, L_k$. Denotem per $G_i \setminus L_j$, aquest objectiu G_i , on el literal L_j ha sigut eliminat. Per tant $G_i \setminus L_j = \leftarrow L_1, \dots, L_{j-1}, L_{j+1}, \dots, L_k$. Observi's que, si $G_i = \leftarrow L_j$, aleshores $G_i \setminus L_j = \square$.

Descripció de la Derivació Constructiva

Una derivació constructiva des de $(G_1 T_1 C_1)$ fins a $(G_n T_n C_n)$ per via d'una regla de càlcul segura R és una tal seqüència:

$$(G_1 T_1 C_1), (G_2 T_2 C_2), \dots, (G_n T_n C_n)$$

que per a tot $i > 1$, G_i té la forma $\leftarrow L_1, \dots, L_k$, $R(G_i) = L_j$ i $(G_{i+1} T_{i+1} C_{i+1})$ s'obté segons una de les regles següents:

- A1) Si L_j és positiu i no és un esdeveniment bàsic, aleshores $G_{i+1} = S$, $T_{i+1} = T_i$ i $C_{i+1} = C_i$, on S és el resolvent d'alguna clàusula en $A(D)$ amb G_i en el literal seleccionat L_j .
- A2) Si L_j és un esdeveniment bàsic positiu " ιP " (resp. " δP "), $L_j \notin C_i$ i el fet P no és cert (resp. és cert) en la base de dades actual, aleshores $G_{i+1} = G_i \setminus L_j$, $T_{i+1} = T_i \cup \{L_j\}$ i $C_{i+1} = C_i$.
- A3) Si L_j és un predicat bàsic o derivat, negatiu i antic " $\neg P$ ", aleshores $G_{i+1} = G_i \setminus L_j$, $T_{i+1} = T_i$ i $C_{i+1} = C_i$, si l'objectiu " $\leftarrow P$ " té èxit. " $\leftarrow P$ " té èxit si $\leftarrow P$ fracassa finitament.
- A4) Si L_j és un esdeveniment bàsic negatiu " $\neg \iota P$ " (resp. " $\neg \delta P$ ") i $\iota P \in T_i$ (resp. $\delta P \in T_i$), aleshores $G_{i+1} = G_i \setminus L_j$, $T_{i+1} = T_i$ i $C_{i+1} = C_i \cup \{\iota P\}$ (resp. $C_{i+1} = C_i \cup \{\delta P\}$).
- A5) Si L_j és un predicat negatiu, nou o esdeveniment derivat, " $\neg P$ " i existeix una derivació de consistència des de $(\{\leftarrow P\} T_i C_i)$ a $(\{T' C')$ aleshores $G_{i+1} = G_i \setminus L_j$, $T_{i+1} = T'$ i $C_{i+1} = C'$.

Els passos corresponents a les regles A1) i A3) són passos de resolució SLDNF. En el cas A2) s'afegeixen esdeveniments bàsics al conjunt de traducció T , per aconseguir que una derivació fracassada de $A(D) \cup \{\leftarrow G_1\}$ tingui èxit. En el cas A4) s'afegeixen esdeveniments bàsics al conjunt de condició C per assegurar que no seran inclosos posteriorment en el conjunt de traducció T . En A5) es comprova la consistència de les actualitzacions de vista induïdes i la petició d'actualització. En alguns casos això requerirà la inclusió d'altres esdeveniments bàsics en T i/o en C .

Descripció de la Derivació de Consistència

Una derivació de consistència des de $(F_1 T_1 C_1)$ fins a $(F_n T_n C_n)$ per via d'una regla de càlcul segura R és una tal seqüència:

$$(F_1 T_1 C_1), (F_2 T_2 C_2), \dots, (F_n T_n C_n)$$

que per a tot $i > 1$, F_i té la forma $\{\leftarrow L_1, \dots, L_k\} \cup F'_i$ i per a algun $j=1\dots k$, $(F_{i+1} T_{i+1} C_{i+1})$ s'obté segons una de les regles següents:

- B1) Si L_j és positiu i no és un esdeveniment bàsic, S' és el conjunt de tots els resolvents de les clàusules en $A(D)$ amb F_i en el literal L_j i $[\] \notin S'$, aleshores $F_{i+1} = S' \cup F'_i$, $T_{i+1} = T_i$ i $C_{i+1} = C_i$.
- B2) Si L_j és positiu, no és un esdeveniment bàsic i no hi ha cap clàusula en $A(D)$ que es pugui unificar amb L_j , aleshores $F_{i+1} = F'_i$, $T_{i+1} = T_i$ i $C_{i+1} = C_i$.
- B3) Si L_j és un esdeveniment bàsic positiu i $L_j \in C_i$, aleshores $F_{i+1} = F'_i$, $T_{i+1} = T_i$ i $C_{i+1} = C_i$.
- B4) Si L_j és un esdeveniment bàsic positiu, $L_j \in T_i$, $L_j \notin C_i$ i $k > 1$, aleshores $F_{i+1} = F_i \setminus L_j \cup F'_i$, $T_{i+1} = T_i$ i $C_{i+1} = C_i$.
- B5) Si L_j és un esdeveniment bàsic positiu " $\text{!}P$ " (resp. " δP "), $L_j \in T_i$ i $L_j \notin C_i$, aleshores $F_{i+1} = F'_i$, $T_{i+1} = T_i$ i $C_{i+1} = C_i \cup \{L_j\}$.
- B6) Si L_j és un predicat antic negatiu, bàsic o derivat " $\neg P$ " i $k > 1$, aleshores $F_{i+1} = F_i \setminus L_j \cup F'_i$, $T_{i+1} = T_i$ i $C_{i+1} = C_i$, si l'objectiu " $\leftarrow P$ " té èxit. " $\leftarrow P$ " té èxit si l'objectiu $\leftarrow P$ fracassa finitament.
- B7) Si L_j és un predicat antic negatiu, bàsic o derivat " $\neg P$ " i existeix una refutació SLDNF de $A(D) \cup \{\leftarrow P\}$, aleshores $F_{i+1} = F'_i$, $T_{i+1} = T_i$ i $C_{i+1} = C_i$.
- B8) Si L_j és un esdeveniment bàsic negatiu i $\neg L_j \in T_i$, aleshores $F_{i+1} = F'_i$, $T_{i+1} = T_i$ i $C_{i+1} = C_i$.
- B9) Si L_j és un esdeveniment bàsic negatiu, $\neg L_j \in C_i$, $\neg L_j \notin T_i$ i $k > 1$, aleshores $F_{i+1} = F_i \setminus L_j \cup F'_i$, $T_{i+1} = T_i$ i $C_{i+1} = C_i$.
- B10) Si L_j és un esdeveniment bàsic negatiu " $\neg \text{!}P$ " (resp. " $\neg \delta P$ "), $\text{!}P \notin C_i$ (resp. $\delta P \notin C_i$), $\text{!}P \notin T_i$ (resp. $\delta P \notin T_i$) i el fet P no és cert (resp. és cert) en la base de dades actual, aleshores $F_{i+1} = F'_i$, $T_{i+1} = T_i \cup \{\text{!}P\}$ (resp. $T_{i+1} = T_i \cup \{\delta P\}$) i $C_{i+1} = C_i$.
- B11) Si L_j és un predicat negatiu, nou o esdeveniment derivat, " $\neg P$ ", $k > 1$ i existeix una derivació de consistència des de $(\{\leftarrow P\} T_i C_i)$ fins a $(\{\} T' C')$, aleshores $F_{i+1} = F_i \setminus L_j \cup F'_i$, $T_{i+1} = T'$ i $C_{i+1} = C'$.
- B12) Si L_j és un predicat negatiu, nou o esdeveniment derivat, " $\neg P$ " i existeix una derivació constructiva des de $(\{\leftarrow P\} T_i C_i)$ fins a $([\] T' C')$, aleshores $F_{i+1} = F'_i$, $T_{i+1} = T'$ i $C_{i+1} = C'$.

Els passos corresponents a les regles B1), B2), B6) i B7) són passos de resolució SLDNF on les clàusules d'A(D) actuen com a clàusules d'entrada. En el cas B3) (resp. B8) se suprimeix la branca actual de la derivació de consistència perquè el subconjunt de C (resp. de T) ja determinat ens assegura el fracàs d'aquesta branca. En el cas B4) (resp. B9), el literal seleccionat se suprimeix de l'objectiu perquè ja pertany al conjunt de traducció (resp. al conjunt de condició).

En el cas B5) (resp. B10) s'afegeixen nous esdeveniments bàsics a C (resp. a T), que es corresponen a les diferents formes en què l'objectiu pot fracassar. En ambdós casos es pot suprimir la branca actual perquè ja s'ha assegurat el seu fracàs. En els casos B11) i B12) es continuarà la derivació per la branca actual o es prescindirà d'aquesta segons hi hagi una derivació de consistència o constructiva associada a la negació del literal seleccionat, respectivament.

Les derivacions de consistència no depenen de l'ordre en què la regla de selecció R selecciona els literals ja que, en general, cal explorar totes les possibilitats per fer fracassar una conjunció $\leftarrow L_1, \dots, L_k$. Cadascuna d'elles es correspon a una traducció diferent.

5. Mètode dels Esdeveniments: cas general

En el capítol anterior hem descrit el Mètode dels Esdeveniments per al cas particular en què totes les variables que apareixen en el cos d'una regla deductiva apareixen també en la seva conclusió. A causa d'aquesta simplificació hem pogut considerar que el conjunt de condició C estava format únicament per esdeveniments bàsics, la qual cosa permetia reduir el procés de verificació de la consistència a una comprovació de pertinença d'un element a un conjunt. En el cas general, és a dir, quan apareixen variables locals en la definició dels predicats vista, cal incloure condicions més complexes en C . Per aquest motiu, el mètode ha de ser estès per poder verificar la consistència d'aquestes noves condicions.

En aquest capítol presentem l'extensió del Mètode dels Esdeveniments per al cas general. Concretament, a la secció 5.1 es presenta una descripció informal d'aquesta extensió, que s'il·lustra amb alguns exemples a la secció 5.2. Finalment, a la secció 5.3 es formalitza el Mètode dels Esdeveniments pel cas general.

5.1 Descripció del mètode

El procediment seguit pel Mètode dels Esdeveniments en el cas general és, bàsicament, el que s'ha descrit en el capítol anterior. Així, sigui D una base de dades, $A(D)$ aquesta base de dades augmentada i u una petició d'actualització de vista, aleshores, una traducció T és un conjunt d'esdeveniments bàsics tals que, usant la resolució SLDNF, l'objectiu $\{\leftarrow u\}$ té èxit considerant el conjunt d'entrada $A(D) \cup T$.

De la mateixa manera que en el cas anterior, una traducció T s'obté fent que alguna derivació fracassada d' $A(D) \cup \{\leftarrow u\}$ tingui èxit. Això s'aconsegueix inclouent en el conjunt d'entrada T una instància de cada esdeveniment bàsic positiu que ocorre en l'objectiu fracassat. Les diferents

possibilitats en què una derivació fracassada pot tenir èxit es corresponen a les múltiples traduccions T_i que satisfan l'actualització de vista demanada.

La diferència més important ve donada per la necessitat d'estendre el tipus de les condicions que s'han d'incloure en el conjunt de condició C . Aquest fet provoca que la verificació de consistència ja no es pugui reduir a una simple comprovació de pertinença d'un element a un conjunt tal i com es feia en el cas simplificat. L'exemple següent serveix per a justificar la necessitat d'incloure condicions més complexes en C .

Exemple 5.1: Considerem una base de dades que conté únicament una regla deductiva i dos fets:

$Q(a)$
 $R(a)$
 $P \leftarrow Q(X) \wedge R(X)$

Algunes regles de transició i d'esdeveniment associades a aquesta base de dades són:

$P'_{1,1} \leftarrow Q(X) \wedge \neg\delta Q(X) \wedge R(X) \wedge \neg\delta R(X)$
 $P'_{1,2} \leftarrow \imath Q(X) \wedge \imath R(X)$
 $P'_1 \leftarrow P'_{1,j} \quad j = 1 \dots 4$
 $\delta P \leftarrow Q(X) \wedge \delta R(X) \wedge \neg P'_1$

Per a obtenir traduccions que satisfan l'esborrat de la vista P , l'arbre subsidiari SLDNF associat al literal $\neg P'_1$ ha de fracassar finitament. Fixem-nos en una branca d'aquest arbre:

$$\begin{array}{c} \leftarrow P'_1 \\ | \\ \leftarrow P'_{1,2} \\ | \\ \leftarrow \imath Q(X) \wedge \imath R(X) \end{array}$$

Assegurar el fracàs d'aquesta branca requereix que no existeixi cap individu 'x' pel qual $\imath Q(x)$ i $\imath R(x)$ siguin certs. En aquest cas, hem d'afegir la condició $\neg\exists X (\imath Q(X) \wedge \imath R(X))$ al conjunt de condició C per a poder verificar que, posteriorment durant el procés de traducció, noves inclusions en T no provoquin que aquesta derivació tingui èxit.

En general, C estarà format per aquells objectius assolits durant la derivació de consistència, la consistència dels quals no pugui ser garantida completament donat el subconjunt de traducció T determinat fins aquell moment. En general, això es produirà quan el literal seleccionat sigui un esdeveniment bàsic que no està plenament instanciat o bé quan sigui un esdeveniment bàsic que no pertany al conjunt de traducció.

traduccions que satisfan aquesta petició s'obtenen fent que tingui èxit l'arbre SLDNF finitament fracassat de $A(D) \cup \{\leftarrow \text{iE}(\text{marc})\}$.

Intuitivament, no és difícil adonar-se que hi ha dues traduccions mínimes que satisfan l'actualització demanada. O bé inserim que en Marc practica els escacs, o bé per a algun valor 'z' concret inserim que 'z' és un esport i que en Marc el practica. Ambdues traduccions s'obtenen a partir de la derivació constructiva de la figura 5.2.1, on les etiquetes encerclades són referències a les regles del mètode definides posteriorment a la secció 5.3.

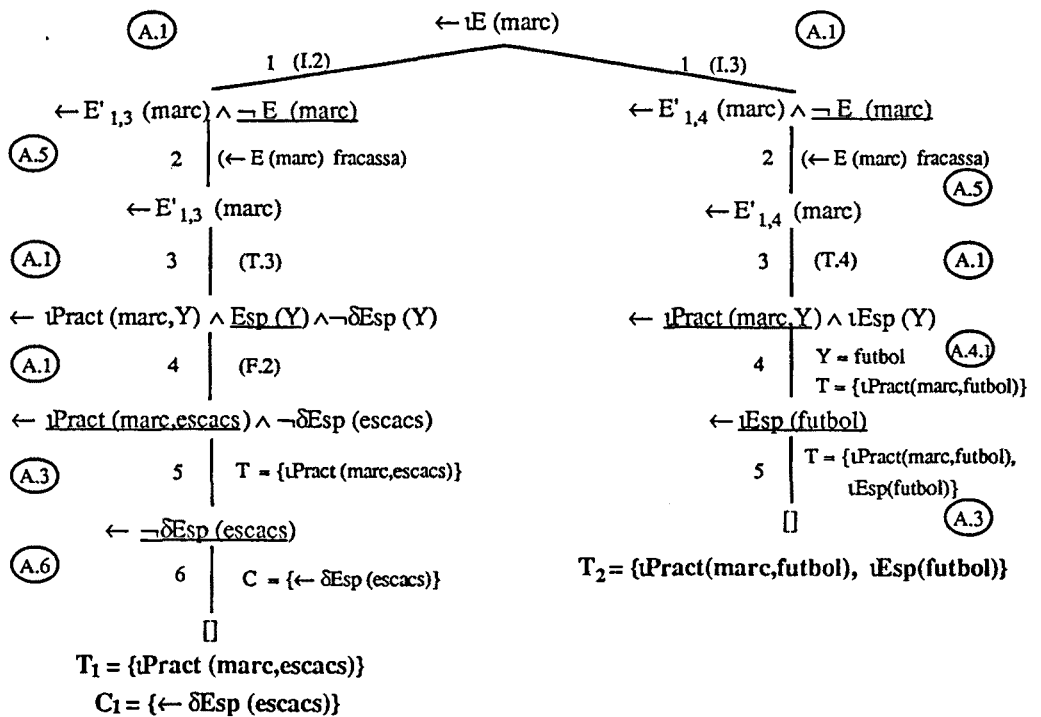


Figura 5.2.1

Els passos 1, 2, 3 i 4 de la branca de l'esquerra són passos de resolució SLDNF on $A(D)$ actua com a conjunt d'entrada. En el pas 5, l'esdeveniment bàsic positiu seleccionat $\text{iPract}(\text{marc}, \text{escacs})$ s'inclou en el conjunt de traducció T per a aconseguir que aquesta derivació tingui èxit. En el pas 6, el literal seleccionat és $\neg \delta\text{Esp}(\text{escacs})$. Aquesta derivació tindrà èxit si garantim que l'esdeveniment bàsic $\delta\text{Esp}(\text{escacs})$ no s'inclourà posteriorment en T . Per això, la condició $\leftarrow \delta\text{Esp}(\text{escacs})$ és afegida a C .

El procés de traducció finalitza quan s'assoleix la clàusula buida. Aleshores, la traducció obtinguda en aquesta branca és $T_1 = \{\neg \text{Pract}(\text{marc}, \text{escacs})\}$. És a dir, podem deduir que en Marc és un esportista si afegim a la base de dades el fet que en Marc practica els escacs.

El procediment que s'ha seguit fins ara no comporta cap novetat respecte al que s'ha descrit en el capítol anterior. Fixem-nos, però, en la branca de la dreta. Els passos 1, 2 i 3 també són passos de resolució SLDNF on $A(D)$ actua com a conjunt d'entrada.

Després del pas 3 s'assoleix l'objectiu $\neg \text{Pract}(\text{marc}, Y) \wedge \text{Esp}(Y)$. Aquest objectiu es pot satisfer si instanciem la variable Y a algun valor 'z' pel qual es compleixi que tant $\neg \text{Pract}(\text{marc}, z)$ com $\text{Esp}(z)$ són certs. Aquesta instanciació es pot realitzar assignant valors per defecte, o bé demanant-los a l'usuari. En el pas 4 de la figura 5.2.1, el valor 'futbol' s'assigna a la variable Y mitjançant alguna d'aquestes tècniques i que l'esdeveniment $\neg \text{Pract}(\text{marc}, \text{futbol})$ s'afegeix al conjunt de traducció. En cas que es desitjés obtenir totes les solucions, l'usuari hauria de donar tots els valors possibles, amb la qual cosa sortirien tantes branques com valors diferents hi hagués.

En els pas 5, l'esdeveniment bàsic seleccionat s'inclou en T per aconseguir que aquesta derivació tingui èxit.

Un cop assolit l'èxit en aquesta derivació s'ha obtingut una altra traducció: $T_2 = \{\neg \text{Pract}(\text{marc}, \text{futbol}), \text{Esp}(\text{futbol})\}$. És a dir, l'actualització de vista també se satisfarà inserint el fet que el Futbol és un esport i que en Marc el practica. Observi's que les traduccions obtingudes pel Mètode dels Esdeveniments es corresponen a les que hem esmentat anteriorment.

Seleccionant la clàusula I.1 en el pas 1 de la figura 5.2.1 s'obté una derivació fracassada a la qual no es pot fer tenir èxit i, per tant, aquesta derivació no genera cap altra traducció. Aquesta derivació no es mostra en la figura anterior.

Esborrat d'una vista: Considerem un altre cop la mateixa base de dades i suposem que volem esborrar el fet que en David és un esportista. Intuïtivament, això es pot aconseguir si esborrem el fet que en 'david' practica els escacs o bé si deixem de considerar aquests com un esport.

Les traduccions que satisfan aquesta petició s'obtenen fent que una derivació fracassada d' $A(D) \cup \{\neg \text{Esp}(\text{david})\}$ tingui èxit. Això es mostra a la figura 5.2.2, on les etiquetes encerclades són referències a les regles del mètode definides a la secció 5.3.

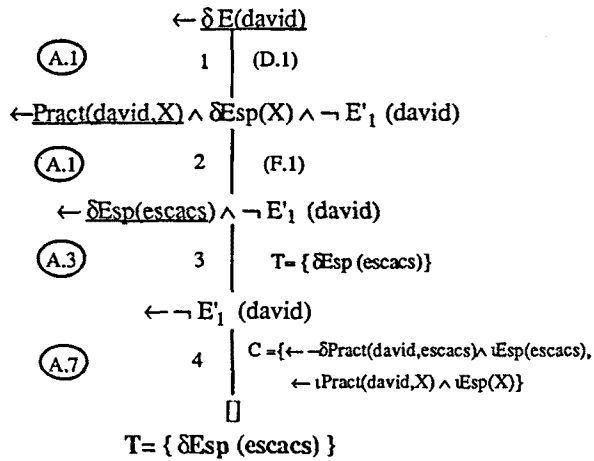


Figura 5.2.2

Els passos 1 i 2 són passos de resolució SLDNF on A(D) actua com a conjunt d'entrada. En el pas 3 se selecciona el literal $\delta \text{Esp}(\text{escacs})$. Com que es tracta d'un esdeveniment bàsic positiu, aquest literal és inclòs en T. En el pas 4, el literal seleccionat és $\neg E'_1(\text{david})$. Perquè aquesta derivació tingui èxit, l'arbre subsidiari SLDNF per a $A(D) \cup T \cup \{ \leftarrow E'_1(\text{david}) \}$ ha de fracassar finitament. A la figura 5.2.3 es mostra el fracàs d'aquest arbre.

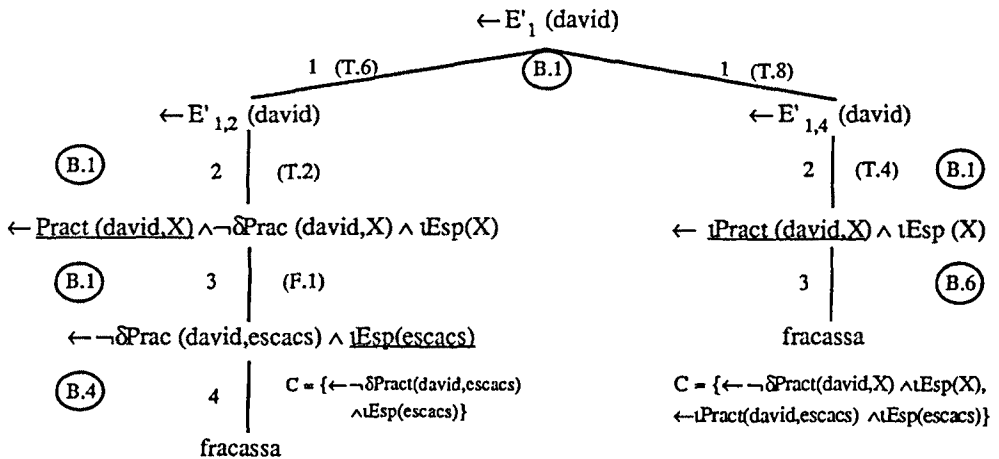


Figura 5.2.3

Els passos 1, 2 i 3 de la branca esquerra són passos de resolució SLDNF on A(D) actua com a conjunt d'entrada. En el pas 4, la branca fracassa perquè no hi ha cap fet en el conjunt d'entrada T que es pugui resoldre amb el literal seleccionat. L'objectiu actual s'afegeix al conjunt de condició C per a garantir que, posteriorment durant el procés de derivació, noves inclusions en T no facin que aquesta branca tingui èxit.

En la branca dreta, els passos 1 i 2 són passos de resolució SLDNF on les clàusules d'A(D) actuen com a clàusules d'entrada. En el pas 3, la branca fracassa perquè no hi ha cap fet en el conjunt d'entrada T que es pugui resoldre amb el literal seleccionat. Aleshores, la condició $\leftarrow \neg \text{Pract}(\text{david}, X) \wedge \neg \text{Esp}(X)$ s'afegeix a C pels mateixos motius que en el cas anterior.

Les branques resultants de seleccionar les clàusules T.5 i T.7 en el pas 1 de la figura anterior, fracassen sense modificar el contingut del conjunt de traducció T ni el del conjunt de condició C. Aquestes branques no es mostren en l'exemple.

Un cop finalitzat l'arbre de la figura 5.2.3, la derivació inicial de la figura 5.2.2 assoleix la clàusula buida i, per tant, s'acaba el procés de traducció. En aquest cas, s'ha obtingut la traducció $T = \{\delta \text{Esp}(\text{escacs})\}$. Per tant, l'esborrat del fet $\text{Esp}(\text{escacs})$ satisfà l'actualització de vista demanada. És a dir, en David deixarà de ser un esportista si ja no considerem que els escacs són un esport.

Seleccionant la clàusula D.2 en el pas 1 de la figura 5.2.2, s'obté una altra solució $T = \{\delta \text{Pract}(\text{david}, \text{escacs})\}$. Aquesta derivació no es mostra en la figura anterior. Observeu que les traduccions obtingudes pel Mètode dels Esdeveniments es corresponen a les esmentades anteriorment.

Exemple 5.3: Com hem esmentat a la secció 4.3, el Mètode dels Esdeveniments permet tractar actualitzacions de vista múltiples i mixtes. En aquest exemple, una actualització d'aquest tipus ens serveix per a comprovar com una inclusió d'un esdeveniment bàsic en el conjunt de traducció T pot provocar que la derivació de consistència associada a una condició de C tingui èxit.

Sigui D una base de dades que conté dos predicats bàsics (El i Cont) i dos predicats derivats (Dr i Atur). El significat d'aquests predicats és el següent:

El (X)	X està en edat laboral
Cont (X)	X està contractat per alguna empresa
Der (X)	X ha de fer la declaració de renda. Suposarem que només fan declaració de renda aquelles persones que estan en edat laboral.
Atur	Hi ha atur. Entenent que hi ha atur si alguna persona en edat laboral no està contractada.

El contingut actual d'aquesta base de dades és:

F.1	El (marta)
DR.1	Der (X) \leftarrow El (X)
DR.2	Atur \leftarrow El (X) \wedge \neg Cont (X)

Algunes regles de transició i d'esdeveniment associades a aquesta base de dades són:

- T.1 $Der'_{1,1}(X) \leftarrow El(X) \wedge \neg \delta El(X)$
T.2 $Der'_{1,2}(X) \leftarrow \imath El(X)$
T.3 $Atur'_{1,1} \leftarrow El(X) \wedge \neg \delta El(X) \wedge \neg Cont(X) \wedge \neg \imath Cont(X)$
T.4 $Atur'_{1,2} \leftarrow El(X) \wedge \neg \delta El(X) \wedge \delta Cont(X)$
T.5 $Atur'_{1,3} \leftarrow \imath El(X) \wedge \neg Cont(X) \wedge \neg \imath Cont(X)$
T.6 $Atur'_{1,4} \leftarrow \imath El(X) \wedge \delta Cont(X)$
T.7..10 $Atur'_1 \leftarrow Atur'_{1,j} \quad j=1...4$
I.1 $\imath Der(X) \leftarrow Der'_{1,2}(X)$
D.1 $\delta Atur \leftarrow El(X) \wedge \imath Cont(X) \wedge \neg Atur'_1$
D.2 $\delta Atur \leftarrow \delta El(X) \wedge \neg Cont(X) \wedge \neg Atur'_1$

Suposem que volem esborrar el fet que hi ha Atur i volem inserir que l'Elm ha de fer declaració de renda. Les traduccions que satisfan aquesta actualització de vista múltiple i mixta s'obtenen fent que una derivació fracassada d' $A(D) \cup \{\leftarrow \delta Atur \wedge \imath Der(elm)\}$ tingui èxit. Aquesta derivació es mostra a la Figura 5.2.4, on les etiquetes encerclades són referències a les regles del mètode definides a la propera secció.

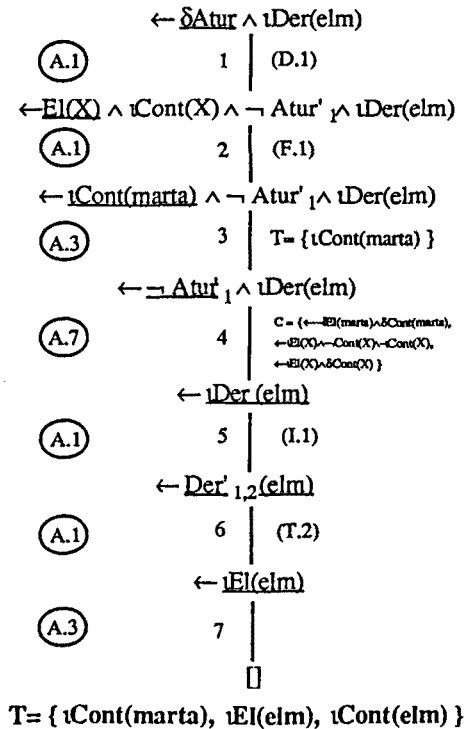


Figura 5.2.4

Observi's que la inclusió de $\iota El(elm)$ en el conjunt de traducció T en el pas 7 de la figura 5.2.4 es contradiu amb l'esborrat del fet vista $Atur$. L'existència del conjunt de condició C ens ha permès detectar aquesta contradicció (mitjançant la violació de la condició $\leftarrow \iota El(X) \wedge \neg Cont(X) \wedge \neg \iota Cont(X)$) i realitzar les accions correctores necessàries (fet que ha provocat la inclusió de $\iota Cont(elm)$ en T).

Si se selecciona la regla D.2 en el pas 1 de la figura 5.2.4, s'obté una altra traducció $T = \{\delta El(marta), \iota El(elm), \iota Cont(elm)\}$. Aquesta derivació no es mostra en la figura anterior.

5.3 Formalització del Mètode dels Esdeveniments

En aquesta secció descrivim formalment el Mètode dels Esdeveniments per a l'actualització de vistes en bases de dades deductives. Com ja hem esmentat a la secció 4.3, aquest mètode és una alternança de dues activitats: 1) satisfer l'actualització de la vista incloent esdeveniments bàsics en el conjunt de traducció i 2) comprovar que les actualitzacions de vista induïdes per aquests esdeveniments bàsics no són contradictòries amb l'actualització demanada. Aquestes dues activitats es realitzen, respectivament, durant les derivacions constructiva i de consistència descrites a continuació.

Sigui u una petició d'actualització de vista. T serà una traducció d' u si existeix una derivació constructiva des de $(\leftarrow u \ \emptyset \ \emptyset)$ fins a $(\sqcup T \ C)$. Els esdeveniments bàsics continguts en el conjunt de traducció T es corresponen a les modificacions (insercions i/o esborrats) que cal efectuar en la base de dades actual per a satisfer l'actualització demanada. En general, existiran diverses derivacions constructives. Cadascuna es correspon a una traducció diferent. Quan no n'hi ha cap, l'actualització de la vista no es pot satisfer canviant únicament la base de dades extensional.

Descripció de la Derivació Constructiva

Una derivació constructiva des de $(G_1 \ T_1 \ C_1)$ fins a $(G_n \ T_n \ C_n)$ per via d'una regla de càlcul segura R és una tal seqüència:

$$(G_1 \ T_1 \ C_1), (G_2 \ T_2 \ C_2), \dots, (G_n \ T_n \ C_n)$$

que per a tot $i > 1$, G_i té la forma $\leftarrow L_1, \dots, L_k$, $R(G_i) = L_j$ i $(G_{i+1} \ T_{i+1} \ C_{i+1})$ s'obté segons una de les regles següents:

A1) Si L_j és positiu, no és un esdeveniment bàsic, i S és el resolvent d'alguna clàusula en $A(D)$ amb G_i en el literal seleccionat L_j , aleshores $G_{i+1} = S$, $T_{i+1} = T_i$ i $C_{i+1} = C_i$.

A2) Si L_j és un esdeveniment bàsic positiu instanciat i $L_j \in T_i$, aleshores $G_{i+1} = G_i \setminus L_j$, $T_{i+1} = T_i$ i $C_{i+1} = C_i$.

A3) Si L_j és un esdeveniment bàsic positiu instanciat "iP" (resp. "δP"), $L_j \in T_i$, i P no és cert (resp. P és cert) en la base de dades.

Si $C_i = \{\leftarrow Q_1, \dots, \leftarrow Q_k, \dots, \leftarrow Q_n\}$ i existeixen derivacions de consistència des de $(\{\leftarrow Q_1\} T_i \cup \{L_j\} C_i)$ fins a $(\{ \} T^1 C^1)$,, des de $(\{\leftarrow Q_k\} T^{k-1} C^{k-1})$ fins a $(\{ \} T^k C^k)$,, des de $(\{\leftarrow Q_n\} T^{n-1} C^{n-1})$ fins a $(\{ \} T^n C^n)$

aleshores $G_{i+1} = G_i \setminus L_j$, $T_{i+1} = T^n$ i $C_{i+1} = C^n$.

Observi's que si $C_i = \emptyset$, aleshores $G_{i+1} = G_i \setminus L_j$, $T_{i+1} = T_i \cup \{L_j\}$ i $C_{i+1} = C_i$.

A4) Si L_j és un esdeveniment bàsic positiu no instanciat, i σ és una tal substitució que:

A4.1) Si L_j és un esdeveniment bàsic positiu no instanciat "iP", aleshores σ és una substitució de les variables de P de manera que $P\sigma$ no és a la base de dades.

A4.2) Si L_j és un esdeveniment bàsic positiu no instanciat "δP", aleshores σ és alguna substitució de les variables de P de manera que $P\sigma$ és a la base de dades.

Si $C_i = \{\leftarrow Q_1, \dots, \leftarrow Q_k, \dots, \leftarrow Q_n\}$, i existeixen derivacions de consistència des de $(\{\leftarrow Q_1\} T_i \cup \{L_j\sigma\} C_i)$ fins a $(\{ \} T^1 C^1)$,, des de $(\{\leftarrow Q_k\} T^{k-1} C^{k-1})$ fins a $(\{ \} T^k C^k)$,, des de $(\{\leftarrow Q_n\} T^{n-1} C^{n-1})$ fins a $(\{ \} T^n C^n)$

aleshores $G_{i+1} = (G_i \setminus L_j)\sigma$, $T_{i+1} = T^n$ i $C_{i+1} = C^n$.

Observi's que si $C_i = \emptyset$, aleshores $G_{i+1} = (G_i \setminus L_j)\sigma$, $T_{i+1} = T_i \cup \{L_j\sigma\}$ i $C_{i+1} = C_i$.

A5) Si L_j és un predicat bàsic o derivat, negatiu i antic " $\neg P$ ", aleshores $G_{i+1} = G_i \setminus L_j$, $T_{i+1} = T_i$ i $C_{i+1} = C_i$, si l'objectiu " $\leftarrow \neg P$ " té èxit. " $\leftarrow \neg P$ " té èxit si l'objectiu $\leftarrow P$ fracassa finitament, és a dir, si per alguna regla de càlcul R' l'espai de recerca SLDNF d' $A(D) \cup \{\leftarrow P\}$ fracassa finitament.

A6) Si L_j és un esdeveniment bàsic negatiu instanciat " $\neg P$ " i $P \notin T_i$, aleshores $G_{i+1} = G_i \setminus L_j$, $T_{i+1} = T_i$ i $C_{i+1} = C_i \cup \{\leftarrow P\}$.

A7) Si L_j és un predicat negatiu, nou o d'esdeveniment i existeix una derivació de consistència des de $(\{\leftarrow L_j\} T_i C_i)$ fins a $(\{ \} T' C')$, aleshores $G_{i+1} = G_i \setminus L_j$, $T_{i+1} = T'$ i $C_{i+1} = C'$.

Els passos corresponents a les regles A1) i A5) són passos de resolució SLDNF on les clàusules d' $A(D)$ actuen com a clàusules d'entrada. El pas corresponent a la regla A2) és un pas SLDNF on el conjunt de traducció T actua com a conjunt d'entrada.

En el cas A3), l'esdeveniment bàsic seleccionat s'inclou en el conjunt de traducció T_i si es pot assegurar la consistència de totes les condicions de C_i . En alguns casos això pot provocar la inclusió

de nous elements en T_i i/o en C_i . Observi's que si $C_i = \emptyset$, les verificacions de consistència no s'han d'efectuar ja que no existeix cap condició per satisfer.

En el cas A4) se selecciona un esdeveniment bàsic positiu que no està plenament instanciat. Aleshores, cal instanciar-lo per a delimitar exactament l'esdeveniment bàsic al qual fa referència (subcasos A4.1 i A4.2). En el cas A4.1) la instanciació es pot efectuar assignant valors per defecte o bé demanant-los a l'usuari. En A4.2) hi haurà tantes alternatives com fets de la base de dades es puguin unificar amb P. Quan l'esdeveniment bàsic ja està completament delimitat, es procedeix de la mateixa manera que en el cas A3).

En el cas A6), l'esdeveniment bàsic seleccionat s'inclou en el conjunt de condició C_i per a assegurar que noves incorporacions en el conjunt de traducció T_i , no provoquin que aquesta branca fracassi.

En el cas A7), s'assoleix l'objectiu següent si es pot garantir la consistència del literal seleccionat. Això també pot provocar noves inclusions en T_i i/o en C_i .

Descripció de la Derivació de Consistència

Una derivació de consistència des de $(F_1 T_1 C_1)$ fins a $(F_n T_n C_n)$ per via d'una regla de càlcul segura R és una tal seqüència:

$$(F_1 T_1 C_1), (F_2 T_2 C_2), \dots, (F_n T_n C_n)$$

que per a tot $i \geq 1$, F_i té la forma $\{\leftarrow L_1, \dots, L_k\} \cup F'_i$ i per a algun $j=1 \dots k$, $(F_{i+1} T_{i+1} C_{i+1})$ s'obté segons una de les regles següents:

- B1) Si L_j és positiu, no és un esdeveniment bàsic, S' és el conjunt de tots els resolvents de les clàusules en $A(D)$ amb F_i en el literal L_j i $[\] \notin S'$, aleshores $F_{i+1} = S' \cup F'_i$, $T_{i+1} = T_i$ i $C_{i+1} = C_i$.
- B2) Si L_j és positiu, no és un esdeveniment bàsic i no hi ha cap clàusula en $A(D)$ que es pugui unificar amb L_j , aleshores $F_{i+1} = F'_i$, $T_{i+1} = T_i$ i $C_{i+1} = C_i$.
- B3) Si L_j és un esdeveniment bàsic positiu instanciat, $L_j \in T_i$ i $k > 1$, aleshores $F_{i+1} = F_i \setminus L_j \cup F'_i$, $T_{i+1} = T_i$ i $C_{i+1} = C_i$.
- B4) Si L_j és un esdeveniment bàsic positiu instanciat i $L_j \notin T_i$, aleshores $F_{i+1} = F'_i$, $T_{i+1} = T_i$ i $C_{i+1} = C_i \cup \{F_j\}$.

- B5) Si L_j és un esdeveniment bàsic positiu no instanciat, S' és el conjunt de tots els resolvents en T_i amb F_i en el literal L_j i $[\] \notin S'$, aleshores $F_{i+1} = S' \cup F'_i$, $T_{i+1} = T_i$ i $C_{i+1} = C_i \cup \{F_i\}$.
- B6) Si L_j és un esdeveniment bàsic positiu no instanciat, i no hi ha fets en T_i que es puguin unificar amb L_j , aleshores $F_{i+1} = F'_i$, $T_{i+1} = T_i$ i $C_{i+1} = C_i \cup \{F_i\}$.
- B7) Si L_j és un predicat antic negatiu, bàsic o derivat, " $\neg P$ " i $k > 1$, aleshores $F_{i+1} = F_i \setminus L_j \cup F'_i$, $T_{i+1} = T_i$ i $C_{i+1} = C_i$, si l'objectiu " $\leftarrow \neg P$ " té èxit. " $\leftarrow \neg P$ " té èxit si l'objectiu $\leftarrow P$ fracassa finitament, és a dir, si per alguna regla de càlcul R' l'espai de recerca SLDNF d' $A(D) \cup \{\leftarrow P\}$ fracassa finitament.
- B8) Si L_j és un predicat antic negatiu, bàsic o derivat " $\neg P$ " i existeix una refutació SLDNF d' $A(D) \cup \{\leftarrow P\}$, aleshores $F_{i+1} = F'_i$, $T_{i+1} = T_i$ i $C_{i+1} = C_i$.
- B9) Si L_j és un esdeveniment bàsic negatiu instanciat i $\neg L_j \in T_i$, aleshores $F_{i+1} = F'_i$, $T_{i+1} = T_i$ i $C_{i+1} = C_i$.
- B10) Si L_j és un esdeveniment bàsic negatiu instanciat, $\neg L_j \in T_i$ i $k > 1$, aleshores $F_{i+1} = F_i \setminus L_j \cup F'_i$, $T_{i+1} = T_i$ i $C_{i+1} = C_i$.
- B11) Si L_j és un esdeveniment bàsic negatiu instanciat, $\neg L_j \in T_i$ i existeix una derivació constructiva des de $(\{\leftarrow \neg L_j\} T_i C_i)$ fins a $([\] T' C')$, aleshores $F_{i+1} = F'_i$, $T_{i+1} = T'$ i $C_{i+1} = C'$.
- B12) Si L_j és un predicat negatiu, nou o esdeveniment derivat, $k > 1$, i existeix una derivació de consistència des de $(\{\leftarrow \neg L_j\} T_i C_i)$ fins a $(\{\}\ T' C')$, aleshores $F_{i+1} = F_i \setminus L_j \cup F'_i$, $T_{i+1} = T'$ i $C_{i+1} = C'$.
- B13) Si L_j és un predicat negatiu, nou o esdeveniment derivat, i existeix una derivació constructiva des de $(\{\leftarrow \neg L_j\} T_i C_i)$ fins a $([\] T' C')$, aleshores $F_{i+1} = F'_i$, $T_{i+1} = T'$ i $C_{i+1} = C'$.

Els passos corresponents a les regles B1), B2), B7) i B8) són passos de resolució SLDNF on les clàusules d' $A(D)$ actuen com a clàusules d'entrada. Els passos corresponents a les regles B3), B9) i B10) són passos SLDNF on el conjunt de traducció T actua com a conjunt d'entrada.

En el cas B4), la branca actual ja fracassa a causa del contingut actual del conjunt de traducció T_i . A més, l'objectiu F_i s'inclou en el conjunt de condició C_i per a assegurar que noves inclusions en T_i no provoquin que aquesta branca tingui èxit.

En els casos B5) i B6) s'afegeixen noves condicions a C_i si es pot garantir la seva consistència a causa del subconjunt de T ja determinat. Concretament, en els cas B5) es verifica la consistència mentre que en el cas B6) la branca actual ja fracassa i, per tant, en podem prescindir.

En el cas B11), es prescindeix de la branca actual si existeix una derivació constructiva per la negació del literal seleccionat.

En els casos B12) i B13) es continuarà la derivació per la branca actual o se'n podrà prescindir segons si existeix una derivació de consistència o bé una de constructiva per a la negació del literal seleccionat.

Les derivacions de consistència no depenen de l'ordre en què la regla de càlcul R selecciona els literals ja que, en general, cal explorar totes les possibilitats per a fer fracassar una conjunció $\leftarrow L_1, \dots, L_k$. Cal tenir en compte que cadascuna pot conduir a una traducció diferent.

A la figura 5.3.1 es pot veure, de manera resumida, quin és el pas que s'aplica en les derivacions constructiva i de consistència en funció del tipus de literal seleccionat.

Constructiva	Positiu	Negatiu	Consistència	Positiu	Negatiu
Pred. antic bàsic	A1	A5	Pred. antic bàsic	B1, B2	B7, B8
Pred. antic deriv.			Pred. antic deriv.		
Esdev. bàsic	A2, A3, A4	A6	Esdev. bàsic	B3, B4, B5, B6	B9, B10, B11
Esdev. derivat	A1	A7	Esdev. derivat	B1, B2	B12, B13
Predicat nou			Predicat nou		

Figura 5.3.1

6. Correctesa i Completesa

En aquest capítol es demostra la correctesa i la completesa del Mètode dels Esdeveniments segons la formalització definida per les regles de la secció 5.3.

6.1 Correctesa

La nostra primera tasca consisteix a donar una definició precisa dels conceptes de derivació constructiva de nivell k i de derivació de consistència de nivell k .

Definició: Sigui G un objectiu qualsevol, T i T' conjunts de traducció i C i C' conjunts de condició. Una *derivació de consistència de nivell 0* de $(G \ T \ C)$ a $(\{\} \ T' \ C')$ és una derivació de consistència que no crida a cap derivació constructiva ni a cap derivació de consistència.

Definició: Sigui G un objectiu qualsevol, T i T' conjunts de traducció i C i C' conjunts de condició. Una *derivació constructiva de nivell 0* de $(G \ T \ C)$ a $(\{\} \ T' \ C')$ és una derivació constructiva que no crida a cap derivació de consistència, o bé que crida únicament a derivacions de consistència de nivell 0.

Definició: Sigui G un objectiu qualsevol, T i T' conjunts de traducció i C i C' conjunts de condició. Una *derivació de consistència de nivell $k+1$* de $(G \ T \ C)$ a $(\{\} \ T' \ C')$ és una derivació de consistència que crida a alguna derivació constructiva o de consistència de nivell k .

Definició: Sigui G un objectiu qualsevol, T i T' conjunts de traducció i C i C' conjunts de condició. Una *derivació constructiva de nivell $k+1$* de $(G \ T \ C)$ a $(\{\} \ T' \ C')$ és una derivació constructiva que crida a alguna derivació de consistència de nivell $k+1$.

Observi's que el concepte de nivell no es correspon exactament al rang d'una derivació SLDNF definit per Lloyd [Llo87].

Per demostrar la correctesa del Mètode dels Esdeveniments necessitem demostrar el lema 6.1. Sigui u una actualització de vista. Aquest lema estableix que hi ha una refutació d' $A(D) \cup T \cup \{\leftarrow u\}$ per a tota traducció T obtinguda pel nostre mètode.

Lema 6.1: Sigui D una base de dades deductiva, $A(D)$ la base de dades augmentada, u una actualització de vista i T una traducció mínima tal que existeix una derivació constructiva des de $(\leftarrow u) \cup \emptyset$ fins a $(\sqcup T C)$. Aleshores hi ha una refutació SLDNF d' $A(D) \cup T \cup \{\leftarrow u\}$.

Demostració: Hem de demostrar que els passos usats en les derivacions constructives i en les de consistència es corresponen a passos de resolució SLDNF, on les clàusules d' $A(D) \cup T$ actuen com a clàusules d'entrada. La demostració la farem per inducció sobre el nivell k de les derivacions constructiva i de consistència.

Sigui G un objectiu qualsevol, T i T' conjunts de traducció i C i C' conjunts de condició. Per començar, considerem el cas $k=0$. Primer demostrarem que a una derivació de consistència li correspon un arbre SLDNF finitament fracassat. Aquest resultat s'usa després per demostrar que a una derivació constructiva li correspon una refutació SLDNF.

(1) Sigui CS una *derivació de consistència* de nivell 0 de $(G T C)$ a $(\{\} T' C')$. Llavors l'arbre de recerca SLDNF d' $A(D) \cup T' \cup \{G\}$ fracassa finitament. Observi's que aquestes derivacions no alteren el conjunt de traducció, és a dir, es compleix que $T=T'$.

- Els passos B1), B2), B7) i B8) són passos de resolució SLDNF on les clàusules d' $A(D)$ actuen com a clàusules d'entrada.

- Als passos B3), B4), B5), B6), B9) i B10) els correspon un pas de resolució SLDNF on T' actua com a conjunt d'entrada.

(2) Sigui CT una *derivació constructiva* de nivell 0 de $(G T C)$ a $(\sqcup T' C')$. Llavors hi ha una refutació SLDNF d' $A(D) \cup T' \cup \{G\}$.

Cas1: No es crida cap derivació de consistència.

- Els passos A1) i A5) són passos de resolució SLDNF on les clàusules d' $A(D)$ actuen com a clàusules d'entrada.

- Als passos A2), A3) i A6) els correspon un pas de resolució SLDNF on T' actua com a conjunt d'entrada. Observi's que, en aquest cas, en el pas A3) no s'efectua cap derivació de consistència.

- En el pas A4), l'esdeveniment bàsic positiu seleccionat s'inclou en el conjunt de traducció un cop instanciat. Per tant, a aquest pas de la derivació constructiva li correspon un pas de resolució SLDNF on les clàusules de T' actuen com a clàusules d'entrada.

Cas2: Es crida una o més derivacions de consistència de nivell 0.

- Els passos A1) i A5) són passos de resolució SLDNF, on les clàusules d' $A(D)$ actuen com a clàusules d'entrada.

- Als passos A2) i A6) els correspon un pas de resolució SLDNF on T' actua com a conjunt d'entrada.

- Sigui L_j el literal seleccionat i $C_i = \{Q_1, \dots, Q_n\}$ el conjunt de condició determinat quan s'aplica el pas A3) o A4). S'assoleix el següent objectiu en la derivació constructiva si existeixen les derivacions de consistència de nivell 0 des de $(Q_k T_i \cup \{L_j\} C_i)$ fins a $(\{T_i\} \cup \{L_j\} C_j)$ per a $k=1 \dots n$. Per tant, aquests passos són equivalents a:

· un pas SLDNF on L_j o $L_j\sigma$ actua com a clàusula d'entrada.

· n passos d'aplicació de la regla de negació per fracàs finit, un per cada Q_k , $k=1 \dots n$. Observi's que només s'afegirà l'esdeveniment bàsic a T si aquesta inclusió no altera el fracàs de les derivacions de consistència considerades anteriorment.

- En el pas A7) es comprova que hi hagi una derivació de consistència de nivell 0 des de $(P T_i C_i)$ fins a $(\{T_i\} C_j)$, on $\neg P$ és el literal seleccionat. Com hem demostrat anteriorment a (1), l'existència d'aquesta derivació es correspon al fracàs de l'objectiu $\leftarrow P$ i, per tant, el pas A7) es correspon a la regla de negació per fracàs finit de la resolució SLDNF.

Un cop demostrat el cas bàsic, suposem que el resultat és cert per a derivacions de nivell k . Anem a demostrar que també ho és per a derivacions de nivell $k+1$.

(3) Sigui CS una derivació de consistència de nivell $k+1$ de $(G T C)$ a $(\{T\} T' C')$. Llavors l'arbre de recerca SLDNF d' $A(D) \cup T' \cup \{G\}$ fracassa finitament.

- Els passos B1), B2), B7) i B8) són passos de resolució SLDNF on les clàusules d' $A(D)$ actuen com a clàusules d'entrada.

- Als passos B3), B9) i B10) els correspon un pas de resolució SLDNF on T' actua com a conjunt d'entrada.

- Sigui G_s l'objectiu sobre el que s'apliquen els passos B4), B5) o B6) i T_s i C_s els conjunts de traducció i de condició determinats en aquest moment. Anem a demostrar que a aquests passos els correspon un pas de resolució SLDNF on les clàusules de T' actuen com a clàusules d'entrada:

· per la pròpia definició d'aquests passos, els correspon un pas SLDNF on les clàusules de T_s actuen com a clàusules d'entrada.

· sigui $T_p = T' - T_s$. És a dir, T_p conté els esdeveniments bàsics afegits al conjunt de traducció després d'haver aplicat aquests passos. Aquests esdeveniments bàsics s'hauran inclòs en T_p en el pas A3) o A4) d'alguna derivació constructiva de nivell k o inferior. Com que en els passos B4), B5) i B6) G_s s'afegeix al conjunt de condició C_s , en A3) i en A4) es verifica que aquesta condició fracassi. Per tant, per la hipòtesi d'inducció, als passos B4), B5) i B6) es correspon un pas de resolució SLDNF on les clàusules de T_p actuen com a clàusules d'entrada.

Com que $T' = T_s \cup T_p$, a aquests passos també els correspon un pas de resolució SLDNF on T' actua com a conjunt d'entrada.

- En els passos B11) i B13) es crida una derivació constructiva de nivell k . Per la hipòtesi d'inducció, hi haurà una refutació de l'arbre subsidiari associat al literal seleccionat i , per tant, fracassa la branca actual.

- En el pas B12) es crida una derivació de consistència de nivell k . Per la hipòtesi d'inducció, l'arbre subsidiari associat al literal seleccionat fracassa finitament i , per tant, aquest pas es correspon a la regla de negació per fracàs finit de la resolució SLDNF.

(4) Sigui CT una derivació constructiva de nivell $k+1$ de $(G \ T \ C)$ a $(\{\} \ T' \ C')$. Llavors hi ha una refutació SLDNF d' $A(D) \cup T' \cup \{G\}$.

- Els passos A1) i A5) són passos de resolució SLDNF, on les clàusules d' $A(D)$ actuen com a clàusules d'entrada.

- Al passos A2) i A6) els correspon un pas de resolució SLDNF on T' actua com a conjunt d'entrada.

- Sigui L_j el literal seleccionat i $C_i = \{Q_1, \dots, Q_n\}$ el conjunt de condició determinat que s'aplica el pas A3) o A4). S'assoleix el següent pas en la derivació constructiva si existeixen les derivacions de consistència de nivell $k+1$

des de $(Q_1 \ T_i \cup \{L_j \sigma\} \ C_i)$ fins a $(\{\} \ T^1 \ C^1)$

.....

des de $(Q_n \ T^{n-1} \ C^{n-1})$ fins a $(\{\} \ T^n \ C^n)$

Com hem demostrat a (3), l'existència d'aquestes derivacions es correspon al fracàs dels objectius Q_k , amb $k=1 \dots n$ i, per tant, els passos A3) i A4) es corresponen a un pas SLDNF on L_j o $L_j \sigma$ actua com a clàusula d'entrada i a n passos d'aplicació de la regla de negació per fracàs finit de la resolució SLDNF.

- En el pas A7) es comprova que hi hagi una derivació de consistència de nivell $k+1$ des de $(P \ T_i \ C_i)$ fins a $(\{\} \ T_j \ C_j)$, on P és el literal seleccionat. Com hem demostrat anteriorment a (3), l'existència d'aquesta derivació es correspon al fracàs de l'objectiu P i, per tant, el pas A4) es correspon a la regla de negació per fracàs finit de la resolució SLDNF.

Q.E.D.

Teorema 6.1: (Correctesa del Mètode dels Esdeveniments)

Sigui D una base de dades deductiva, $A(D)$ la base de dades augmentada i u una actualització de vista de manera que u no és una conseqüència lògica de $\text{comp}(A(D))$. Sigui T una traducció qualsevol obtinguda mitjançant el Mètode dels Esdeveniments. Aleshores, u és una conseqüència lògica de $\text{comp}(A(D) \cup T)$.

Demostració: Del Lema 6.1 es dedueix que si existeix una derivació constructiva des de $(\leftarrow u \ \emptyset)$ fins a $([] \ T \ C)$, aleshores hi ha una refutació SLDNF d' $A(D) \cup T \cup \{\leftarrow u\}$. Per tant, per la correctesa de la resolució SLDNF, se segueix que u és una conseqüència lògica de $\text{comp}(A(D) \cup T)$.

Q.E.D.

6.2 Completesa

En aquesta secció, demostrem la completesa del Mètode dels Esdeveniments en bases de dades estratificades. Sigui D una base de dades estratificada, u una inserció (resp. esborrat) en una vista i T una traducció mínima tal que u és cert (resp. no ho és) en la base de dades modificada. Aleshores diem que u se segueix d' $A(D) \cup T$. Això vol dir que hi ha una refutació SLDNF de $A(D) \cup T \cup \{\leftarrow u\}$. Per tant, hem de demostrar que el nostre mètode és capaç d'obtenir tots els conjunts de traducció T que satisfan aquesta condició.

Aquesta és la nostra idea de completesa. Qualsevol traducció T tal que usant la resolució SLDNF produeix una refutació de $A(D) \cup T \cup \{\leftarrow u\}$ és obtinguda pel Mètode dels Esdeveniments. Això és equivalent a dir que el nostre mètode obté totes les modificacions de la base de dades T que fan que la base de dades actualitzada impliqui u .

Com ja hem esmentat en el capítol 3, si una base de dades D és estratificada, aleshores la base de dades augmentada $A(D)$ és consistent per crida. Com que la resolució SLDNF és completa en aquest tipus de bases de dades (el que vol dir que per a qualsevol d'aquests T , la resolució SLDNF produeix una refutació), aleshores el Mètode dels Esdeveniments és complet en bases de dades estratificades.

En el teorema 6.5 es demostra que existeix una derivació constructiva des de $(\leftarrow u \ \emptyset \ \emptyset)$ fins a $([] \ T \ C)$ per tots els conjunts de traducció T tals que usant la resolució SLDNF produeixen una refutació d' $A(D) \cup T \cup \{\leftarrow u\}$. Aquest teorema es correspon als resultats de completesa explicats anteriorment. Els teoremes 6.2, 6.3 i 6.4 són necessaris per demostrar el teorema 6.5.

Teorema 6.2: Per demostrar aquest teorema, usarem el concepte de rang d'una refutació SLDNF i rang d'un arbre SLDNF finitament fracassat definit a [Llo87]. Sigui D una base de dades,

$A(D)$ la base de dades augmentada, G un objectiu qualsevol, T , T' i T'' conjunts de traducció i C , C' i C'' conjunts de condició. Aleshores els dos resultats següents són certs:

a) Sigui una refutació SLDNF de rang n d' $A(D) \cup T \cup \{G\}$ llavors

- $\forall T'$ tal que $T \supseteq T'$, i

- $\forall C'$ tal que $\forall C \in C'$ l'arbre SLDNF d' $A(D) \cup T \cup \{C\}$ fracassa finitament i és de rang $n-1$

existeix una derivació constructiva de $(G \ T' \ C')$ a $(\square \ T'' \ C'')$ i $T \supseteq T''$ i $\forall C \in C''$, l'arbre SLDNF per a $A(D) \cup T'' \cup \{C\}$ fracassa finitament i és de rang $n-1$.

b) Sigui un arbre SLDNF de rang n que fracassa finitament per a $A(D) \cup T \cup \{A\}$ llavors

- $\forall T'$ tal que $T \supseteq T'$, i

- $\forall C'$ tal que $\forall C \in C'$ l'arbre SLDNF d' $A(D) \cup T \cup \{C\}$ fracassa finitament i és de rang $n-1$

existeix una derivació de consistència de $(F \ T' \ C')$ a $(\{\} \ T'' \ C'')$ i $T \supseteq T''$ i $\forall C \in C''$, l'arbre SLDNF per a $A(D) \cup T'' \cup \{C\}$ fracassa finitament i és de rang n .

Demostració: Anem a demostrar aquest teorema per inducció sobre el rang n de la refutació o de l'arbre fracassat finitament.

Cas $n = 0$:

a) Com que $n = 0$, en els objectius de la refutació només hi ha literals positius. Partim de $(G \ T' \ \emptyset)$ i anem a veure quin resultat obtenim en funció del literal L_j seleccionat en la derivació SLDNF.

- L_j no és un esdeveniment bàsic. Obtindrem $(S \ T_i \ \emptyset)$. (A1)

- L_j és un esdeveniment bàsic instanciat:

· Si $L_j \in T_i$, obtindrem $(G_i \setminus L_j \ T_i \ \emptyset)$. (A2)

· Si $L_j \notin T_i$, obtindrem $(G_i \setminus L_j \ T_i \cup \{L_j\} \ \emptyset)$. (A3)

(observi's que $L_j \in T_i$, per tant, $T \supseteq T_i \cup \{L_j\}$)

- L_j és un esdeveniment bàsic no instanciat, que s'instanciï amb substitució σ .

Obtindrem $((G_i \setminus L_j) \sigma \ T_i \cup \{L_j \sigma\} \ \emptyset)$. (A4)

(observi's que $L_j \sigma \in T_i$, per tant, $T \supseteq T_i \cup \{L_j \sigma\}$)

La derivació acaba amb \square .

b) Com que $n = 0$, en els objectius de l'arbre només hi ha literals positius. Partim de $(F \ T' \ \emptyset)$ i anem a veure quin resultat obtenim en funció del literal L_j seleccionat en la derivació SLDNF.

- L_j no és un esdeveniment bàsic. Obtindrem $(S' \cup F'_i \ T' \ C_i)$ o $(F'_i \ T' \ C_i)$ segons es pugui resoldre (B1) o no (B2), respectivament.

- L_j és un esdeveniment bàsic instanciat:

· Si $L_j \in T_i$, i $k > 1$, obtindrem $(F_i \setminus L_j \cup F'_i \ T' \ C_i)$. (B3)

· Si $L_j \notin T_i$, obtindrem $(F'_i \ T' \ C_i \cup \{F_i\})$. (B4)

(observi's que F_i és de rang 0)

- L_j és un esdeveniment bàsic no instanciat. Obtenim $(S' \cup F'_i \ T' \ C_i \cup \{F_i\})$ o $(F'_i \ T' \ C_i \cup \{F_i\})$ segons es pugui resoldre (B5) o no (B6), respectivament.

(observi's que F_i és de rang 0)

La derivació acaba amb $\{\}$.

Cas general:

Suposem que el teorema és cert per a refutacions SLDNF i arbres SLDNF finitament fracassats de rang n-1, anem a demostrar que també ho és per a refutacions i arbres finitament fracassats de rang n.

a) Partim de $(G \ T' \ C')$ i anem a veure quin resultat obtenim en funció del literal L_j seleccionat en la derivació SLDNF.

L_j és positiu:

- L_j no és un esdeveniment bàsic. Obtindrem $(S \ T_i \ C_i)$. (A1)

- L_j és un esdeveniment bàsic instanciat:

· Si $L_j \in T_i$, obtindrem $(G_i \setminus L_j \ T_i \ C_i)$. (A2)

· Si $L_j \notin T_i$. Afegim L_j a T_i i anem a verificar que hi ha una derivació de consistència per a $(C_k \ T_i \cup \{L_j\} \ C_i)$, $k = 1 \dots n$, (una per cada $C_k \in C_i$). Com que l'arbre SLDNF per $A(D) \cup T \cup \{C_k\}$ és de rang n-1 i fracassa finitament, aleshores per la hipòtesi d'inducció hi ha una derivació de consistència des de $(C_k \ T_i \cup \{L_j\} \ C_i)$ fins a $(\{\} \ T' \ C')$ amb $T \supseteq T'$ i per a tot $C \in C'$, l'arbre SLDNF per a $A(D) \cup T' \cup \{C\}$ fracassa finitament i és de rang n-1. (A3)

- L_j és un esdeveniment bàsic no instanciat, procedim de manera similar que en el cas anterior un cop s'ha instanciat aquest esdeveniment (A4)

L_j és negatiu:

- L_j és un predicat antic, bàsic o derivat. Obtindrem $(G_i \setminus L_j \ T_i \ C_i)$. (A5)

- L_j és un esdeveniment bàsic instanciat. Obtindrem $(G_i \setminus L_j \ T_i \ C_i \cup \{\leftarrow L_j\})$. Observi's que $A(D) \cup T \cup \{\leftarrow L_j\}$ és un arbre de rang 0 (i, per tant, de rang n-1). (A6)

- L_j és un predicat nou o un esdeveniment derivat. L'arbre SLDNF per a $A(D) \cup T \cup \{L_j\}$ fracassa finitament i és de rang n-1. Aleshores, per la hipòtesi d'inducció existeix una derivació de consistència de $(\leftarrow \neg L_j \ T_j \ C_j)$ a $(\{\} \ T' \ C')$ on T' i C' satisfan les condicions del teorema. (A7)

La derivació acaba amb \square .

b) Partim de $(F \ T' \ C')$ i anem a veure quin resultat obtenim en funció del literal L_j seleccionat en la derivació SLDNF.

L_j és positiu:

- L_j no és un esdeveniment bàsic. Obtindrem $(S' \cup F'_i \ T_i \ C_i)$ o $(F'_i \ T_i \ C_i)$ segons es pugui resoldre (B1) o no (B2), respectivament.

- L_j és un esdeveniment bàsic instanciat:

· Si $L_j \in T_i$ i $k > 1$, obtindrem $(F_i \setminus L_j \cup F'_i \ T_i \ C_i)$. (B3)

· Si $L_j \notin T_i$, obtindrem $(F'_i \ T_i \ C_i \cup \{F_i\})$. (B4)

(observi's que F_i és de rang n)

- L_j és esdeveniment bàsic no instanciat. Obtenim $(S' \cup F'_i \ T_i \ C_i \cup \{F_i\})$ o $(F'_i \ T_i \ C_i \cup \{F_i\})$ segons es pugui resoldre (B5) o no (B6), respectivament.

(observi's que F_i és de rang n)

L_j és negatiu:

- L_j és un predicat antic, basic o derivat. Obtindrem $(F_i \setminus L_j \cup F'_i \ T_i \ C_i)$ o $(F'_i \ T_i \ C_i)$ segons l'arbre subsidiari SLDNF fracassi finitament (B7) o no (B8).

- L_j és un esdeveniment bàsic instanciat.

· Si $L_j \in T_i$, obtindrem $(F'_i \ T_i \ C_i)$. (B9)

· Si $L_j \notin T_i$ i $k > 1$, obtindrem $(F'_i \ T_i \ C_i \cup \{F_i\})$. (B10)

(observi's que F_i és de rang n)

· Si $L_j \notin T_i$. La refutació SLDNF d' $A(D) \cup T \cup \{\leftarrow L_j\}$ és de rang $n-1$. Per tant, per la hipòtesi d'inducció, existeix la derivació constructiva de $(\{\leftarrow L_j\} \ T_i \ C_i)$ a $(\square \ T' \ C')$ on T' i C' satisfan les condicions del teorema. (B11)

- L_j és un predicat nou o un esdeveniment derivat.

· L'arbre SLDNF d' $A(D) \cup T \cup \{\leftarrow L_j\}$ fracassa finitament i és de rang $n-1$.

Per tant, per la hipòtesi d'inducció existeix la derivació de consistència de $(\{\leftarrow L_j\} \ T_i \ C_i)$ a $(\{\} \ T' \ C')$ on T' i C' satisfan les condicions del teorema. (B12)

· Hi ha una refutació SLDNF d' $A(D) \cup T \cup \{\leftarrow L_j\}$. Aquesta refutació és de rang $n-1$. Per tant, per la hipòtesi d'inducció existirà la derivació constructiva de $(\{\leftarrow L_j\} \ T_i \ C_i)$ a $(\square \ T' \ C')$ on T' i C' satisfan les condicions del teorema. (B13)

La derivació acaba amb $\{\}$.

Q.E.D.

Teorema 6.3: Sigui T una traducció mínima. Totes les refutacions (principal i auxiliars) que apareixen a l'espai de cerca SLDNF d' $A(D) \cup T \cup \{\leftarrow u\}$ són assolides per la derivació constructiva de $(\{\leftarrow u\} \ \emptyset \ \emptyset)$ a $(\square \ T \ C)$.

Demostració: Anem a demostrar aquest teorema per reducció a l'absurd. Suposem que hi ha una refutació SLDNF d' $A(D) \cup T \cup \{G_s\}$ que no és assolida per aquesta derivació constructiva.

- No pot ser la refutació principal ja que, per la correctesa del nostre mètode, a la derivació constructiva de $(\{\leftarrow u\} \emptyset \emptyset)$ a $(\square T C)$ li correspon una refutació SLDNF per a $A(D) \cup T \cup \{\leftarrow u\}$.

- Per tant, ha de ser una refutació auxiliar. Sigui A l'arbre SLDNF fracassat finitament de $A(D) \cup T \cup \{G_k\}$ que crida aquesta refutació. Pel teorema 6.2, a aquest arbre li correspon una derivació de consistència de $(G_k T_k C_k)$ a $(\{\} T' C')$. En aquesta derivació no s'assoleix G_s perquè el pas aplicat a algun literal L_j seleccionat anteriorment ho impedeix. Els passos del nostre mètode que prescindeixen de la branca actual són:

· B2), B8) i B9). No pot ser cap d'aquests perquè són passos SLDNF on les clàusules d' $A(D)$ o de T actuen com a clàusules d'entrada.

· B11), B13). En aquest cas es prescindeix de la branca actual perquè hi ha una derivació constructiva per a la negació del literal seleccionat. Però per la correctesa del nostre mètode, a aquesta derivació li correspon una refutació SLDNF. Per tant, tampoc pot ser cap d'aquests passos.

· B4) i B6). Es prescindeix de la branca actual perquè L_j és un esdeveniment bàsic que no pertany a T_i , on T_i és el conjunt de traducció determinat quan s'aplica aquest pas. Aquest esdeveniment no és afegit més tard a T_i , ja que sinó la derivació de consistència continuaria. Llavors, si treiem L_j de T , l'arbre continuarà fracassant i $T - \{L_j\}$ serà també una solució, la qual cosa implica que T no era mínima.

Q.E.D.

Teorema 6.4: Sigui $A(D) \cup T \cup \{\leftarrow u\}$ una refutació SLDNF de manera que T és una traducció mínima. Llavors, per a tot $t \in T$, t s'utilitza en la refutació principal o en alguna refutació auxiliar que apareix en l'espai de recerca SLDNF d' $A(D) \cup T \cup \{\leftarrow u\}$.

Demostració: Suposem que t només s'utilitza en els passos de resolució dels arbres auxiliars que fracassen finitament. Llavors, si treiem t de T els arbres continuaran fracassant i $T - \{t\}$ serà també una solució, la qual cosa implica que T no era mínima.

Q.E.D.

Teorema 6.5: Sigui T una traducció mínima de manera que usant la resolució SLDNF produeix una refutació d' $A(D) \cup T \cup \{\leftarrow u\}$. Aleshores, existeix una derivació constructiva des de $(\{\leftarrow u\} \emptyset \emptyset)$ fins a $(\square T C)$ per a algun conjunt de condició C .

Demostració: Una conseqüència directa del teorema 6.2 és que hi ha una derivació constructiva des de $(\{\leftarrow u\} \emptyset \emptyset)$ fins a $(\square T' C')$ amb $T \supseteq T'$. Pel teorema 6.3, sabem que aquesta derivació constructiva assoleix totes les refutacions que apareixen en l'espai de recerca SLDNF d' $A(D) \cup T \cup$

$\{\leftarrow u\}$. Per tant, T' inclou tots els fets de T usats en aquestes refutacions. Finalment, pel teorema 6.4 sabem que tots els elements de T s'usen en alguna d'aquestes refutacions. Per tant, $T' = T$.

Q.E.D.

Teorema 6.6: (*Completesa del Mètode dels Esdeveniments*)

Sigui D una base de dades estratificada, $A(D)$ la base de dades augmentada i u una actualització de vista qualsevol tal que u no és una conseqüència lògica de $\text{comp}(A(D))$. Aleshores, hi ha una derivació constructiva des de $(\leftarrow u \ \emptyset \ \emptyset)$ fins a $(\prod T \ C)$ per a tota traducció T que satisfà que u és una conseqüència lògica de $\text{comp}(A(D) \cup T)$.

Demostració: De la completesa de la resolució SLDNF se segueix que per a tota traducció T tal que u és una conseqüència lògica de $\text{comp}(A(D) \cup T)$ existeix una refutació SLDNF de $A(D) \cup T \cup \{\leftarrow u\}$. Del teorema 6.5 es dedueix que, si existeix una refutació SLDNF d' $A(D) \cup T \cup \{\leftarrow u\}$, aleshores hi ha una derivació constructiva des de $(\leftarrow u \ \emptyset \ \emptyset)$ fins a $(\prod T \ C)$. Com ja hem esmentat en el capítol 3, si una base de dades D és estratificada, aleshores la base de dades augmentada $A(D)$ és consistent per crida. Donat que la resolució SLDNF és completa en aquest tipus de bases de dades, aleshores el Mètode dels Esdeveniments és complet en bases de dades estratificades.

Q.E.D.

Dels teoremes 6.1 i 6.6 se'n poden deduir dues conclusions importants. Sigui u una petició d'actualització de vista. La correctesa del Mètode dels Esdeveniments ens assegura que si existeix una derivació constructiva des de $(\leftarrow u \ \emptyset \ \emptyset)$ fins a $(\prod T \ C)$, aleshores la base de dades modificada segons T satisfà l'actualització de vista demanada. Altrament, la completesa del Mètode dels Esdeveniments ens assegura que si la derivació constructiva que té com a objectiu inicial fracassa finalment, aleshores l'actualització de vista demanada no es pot satisfer canviant únicament la base de dades extensional.

7. Satisfacció de les restriccions d'integritat

Les restriccions d'integritat són condicions que qualsevol estat de la base de dades ha de satisfer. El seu objectiu principal és el d'impedir que la base de dades contingui informació errònia o inconsistent. Per tant, es necessita algun mecanisme per comprovar que no siguin violades per una transacció.

En els darrers anys, la satisfacció de les restriccions d'integritat ha estat un tema àmpliament estudiat tant en el camp relacional com en el deductiu (vegeu [Oli91] i les referències que en ell s'hi esmenten). Quan una transacció viola una restricció, la integritat de la base de dades es pot recuperar de diferents maneres. Les alternatives més conegudes consisteixen en refusar la transacció (anomenada genèricament *comprovació de les restriccions d'integritat*) o bé en impulsar accions correctores que mitjançant modificacions addicionals de la base de dades en reparin la seva consistència (coneguda amb el nom de *imposició de les restriccions d'integritat*).

Les traduccions d'una actualització de vista són modificacions de la base de dades i, per tant, es corresponen a transaccions. Aleshores, alguna traducció pot ser invalidada pel fet de violar alguna restricció d'integritat. Per aquest motiu, és necessari integrar l'actualització de vistes amb la satisfacció de les restriccions d'integritat per refusar aquelles traduccions que portarien a un estat inconsistent de la base de dades.

Aquesta integració es pot efectuar de dues maneres diferents. Una primera opció consisteix en considerar que ambdós processos són completament independents. És a dir, primer s'obtenen totes les traduccions que satisfan l'actualització de vista demanada i posteriorment es verifica que cadascuna d'elles satisfaci les restriccions d'integritat. Aquest enfocament es correspondria amb el que anteriorment hem anomenat *comprovació de les restriccions d'integritat*.

La segona possibilitat consisteix en incorporar la satisfacció de les restriccions d'integritat durant el procés de traducció. Aleshores, en una única etapa s'obtenen traduccions que satisfan tant

l'actualització de vista demanada com les restriccions d'integritat. Aquest enfocament es correspondria amb el que anteriorment hem anomenat imposició de les restriccions d'integritat. Els avantatges i inconvenients d'ambdós enfocaments es presenten, respectivament, a les seccions 7.1 i 7.2.

7.1 Imposició de les restriccions d'integritat

El Mètode dels Esdeveniments permet considerar els dos enfocaments descrits anteriorment. En aquesta secció s'estudia com es pot incorporar la satisfacció de les restriccions d'integritat en el procés de traducció.

En el nostre mètode, una traducció viola alguna restricció d'integritat si inclou esdeveniments bàsics que provoquen que algun fet $\uparrow Ic_n$ esdevingui cert. O sigui, si fan que s'insereixi algun fet d'inconsistència. A causa del significat natural del concepte d'esdeveniment, augmentant l'objectiu arrel es pot aconseguir detectar quan es produeix un fet d'aquest tipus. Aleshores, aquest objectiu es defineix com $\{\leftarrow u \wedge \neg \uparrow Ic\}$, on u és l'actualització de vista demanada i $\neg \uparrow Ic$ indica que no es pot violar cap restricció d'integritat. El predicat $\uparrow Ic$ es defineix com $\uparrow Ic \leftarrow \uparrow Ic_1, \dots, \uparrow Ic \leftarrow \uparrow Ic_n$, on n és el nombre de restriccions d'integritat i cada $\uparrow Ic_j$ té els seus arguments corresponents.

El procediment per a obtenir les traduccions que satisfan tant l'actualització demanada com les restriccions d'integritat és el mateix que s'ha definit en el capítol 5. T és una traducció si existeix una *derivació constructiva* des de $(\{\leftarrow u \wedge \neg \uparrow Ic\} \emptyset \emptyset)$ fins a $(\square T C)$. Durant aquesta derivació se selecciona el literal $\neg \uparrow Ic$. Com que es tracta d'un esdeveniment derivat negatiu (cas A7), s'assolirà el proper pas de la derivació constructiva si hi ha una *derivació de consistència* des de $(\{\leftarrow \uparrow Ic\} T C)$ fins a $(\{ \} T' C')$, on T i C són els conjunts de traducció i condició ja determinats quan el literal $\neg \uparrow Ic$ és seleccionat. L'existència d'aquesta derivació equival a garantir la consistència de totes les restriccions d'integritat. L'exemple següent serveix per il·lustrar aquesta idea.

Exemple 7.1: Considerem una base de dades que conté dos predicats bàsics (Cont i Atur) i un predicat vista (Treb) i una restricció d'integritat (Ic1). Un fet bàsic Cont(x) indica que l'individu x està contractat, mentre que un fet bàsic Atur(y) ens diu que y es troba a l'atur. El predicat vista Treb ens defineix com a treballadors en actiu totes aquelles persones que estiguin contractades. La restricció d'integritat Ic1 diu que no pot ser que una persona estigui contractada i es trobi a l'atur al mateix temps.

Suposem que l'estat actual de la base de dades és el següent:

F.1 Atur (laia)

- DR.1 $Treb(X) \leftarrow Cont(X)$
 IC $Ic1(X) \leftarrow Cont(X) \wedge Atur(X)$

Algunes regles de transició i d'esdeveniment associades a aquesta base de dades són:

- T.1 $Treb'_{1,1}(X) \leftarrow Cont(X) \wedge \neg\delta Cont(X)$
 T.2 $Treb'_{1,2}(X) \leftarrow \imath Cont(X)$
 T.3 $Ic'_{1,1}(X) \leftarrow Cont(X) \wedge \neg\delta Cont(X) \wedge Atur(X) \wedge \neg\delta Atur(X)$
 T.4 $Ic'_{1,2}(X) \leftarrow Cont(X) \wedge \neg\delta Cont(X) \wedge \imath Atur(X)$
 T.5 $Ic'_{1,3}(X) \leftarrow \imath Cont(X) \wedge Atur(X) \wedge \neg\delta Atur(X)$
 T.6 $Ic'_{1,4}(X) \leftarrow \imath Cont(X) \wedge \imath Atur(X)$
 I.1 $\imath Treb(X) \leftarrow Treb'_{1,2}(X)$
 I.2..4 $\imath Ic1(X) \leftarrow Ic'_{1,j}(X) \quad j = 2 \dots 4$

Suposem que es vol inserir que la Laia és una treballadora en aquesta base de dades. Aquesta actualització de vista es correspon al fet $\imath Treb(laia)$. Per a assegurar-nos de què les traduccions obtingudes satisfaran les restriccions d'integritat, l'objectiu inicial ha d'incloure el literal $\neg \imath Ic$. En aquest cas, com que només hi ha una restricció d'integritat, el predicat $\imath Ic$ es defineix com $\imath Ic \leftarrow \imath Ic1(X)$.

T serà una traducció si hi ha una derivació constructiva des de $\{\{\leftarrow \imath Treb(laia) \wedge \neg \imath Ic\} \emptyset \emptyset\}$ fins a $\{\} T C$. Aquesta derivació es mostra a la figura 7.1, on les etiquetes encadrades són referències a les regles del mètode definides a la secció 5.3.

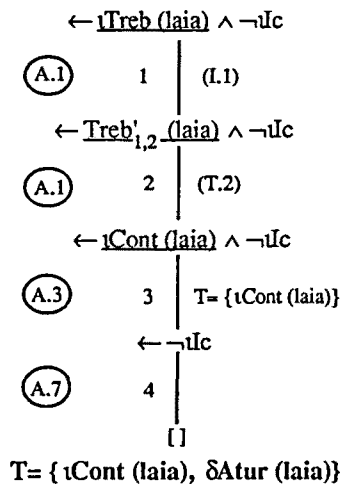


Figura 7.1

Els passos 1 i 2 són passos de resolució SLDNF on A(D) actua com a conjunt d'entrada. En el pas 3, el literal seleccionat és $\neg \text{Cont}(\text{laia})$. Com que es tracta d'un esdeveniment bàsic positiu, s'inclou en el conjunt de traducció T. En el pas 4, se selecciona el literal $\neg \text{Ic}$. Per tant, s'assolirà el següent pas de la derivació constructiva si existeix una derivació de consistència des de $(\{\neg \text{Ic}\} \cup T \cup C)$ fins a $(\{\} \cup T' \cup C')$. Aquesta derivació es mostra a la figura 7.2.

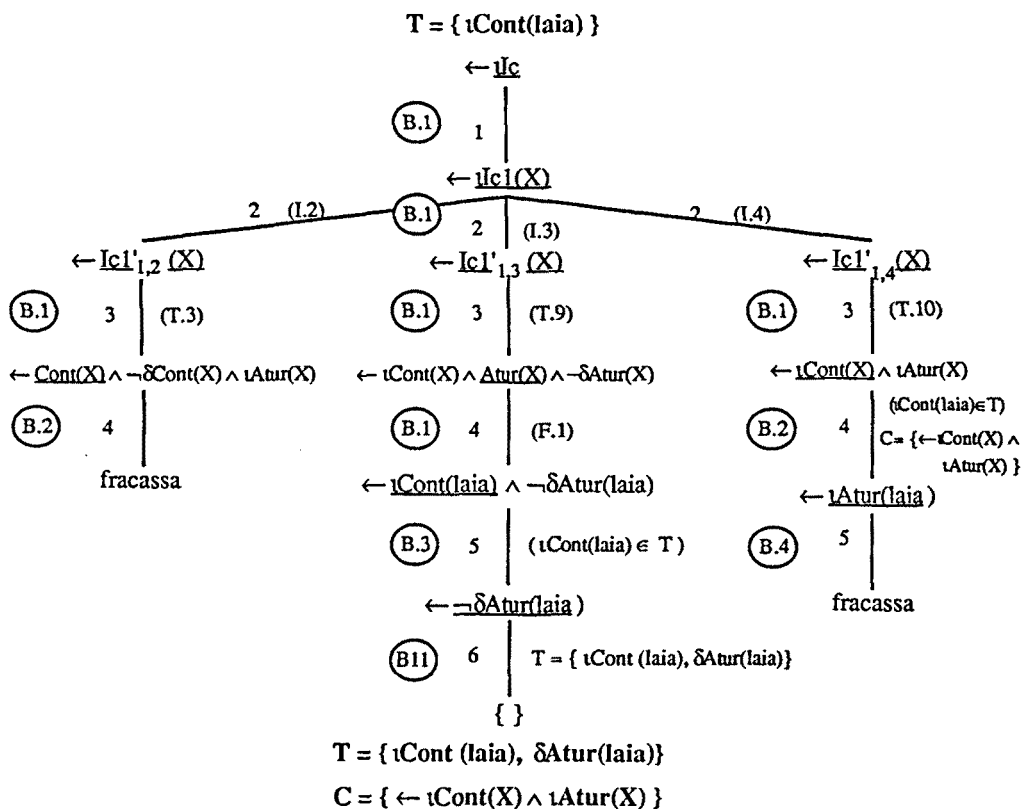


Figura 7.2

La derivació de l'esquerra fracassa perquè no hi ha cap clàusula en el conjunt d'entrada que es pugui unificar amb el literal seleccionat $\text{Cont}(X)$.

Els passos 1 fins al 4 de la derivació central són passos de resolució SLDNF on les clàusules d'A(D) actuen com a clàusules d'entrada. El pas 5 també es un pas de resolució SLDNF, però en aquest cas es resol amb un fet del conjunt de traducció T. En el pas 6, es pot prescindir d'aquesta branca perquè hi ha una derivació constructiva associada al literal $\leftarrow \delta \text{Atur}(\text{laia})$. Aquesta derivació no es mostra en la figura anterior, però provoca la inclusió de l'esdeveniment bàsic $\delta \text{Atur}(\text{laia})$ en el

conjunt de traducció T. Observi's que aquesta inclusió ha estat ocasionada per la necessitat de no violar cap restricció d'integritat.

La derivació de la dreta fracassa en el pas 5 perquè l'esdeveniment bàsic $\neg \text{Atur}(\text{laia})$ no pertany al conjunt d'entrada $A(D) \cup T$.

Un cop fracassat l'arbre de la figura 7.2, la derivació inicial de la figura 7.1 assoleix la clàusula buida i, per tant, acaba el procés de traducció. Aleshores, la traducció obtinguda és $T = \{\neg \text{Cont}(\text{laia}), \delta \text{Atur}(\text{laia})\}$. O sigui, la inserció del fet $\text{Cont}(\text{laia})$ i l'esborrat del fet $\text{Atur}(\text{laia})$ satisfan l'actualització de vista demanada i la restricció d'integritat de la base de dades.

7.2 Comprovació de les restriccions d'integritat

Com hem esmentat anteriorment, l'altre enfocament per a la satisfacció de les restriccions d'integritat també es pot aplicar mitjançant el Mètode dels Esdeveniments. En aquest enfocament es considera que la comprovació de les restriccions d'integritat és un pas addicional, independent de l'actualització de vistes.

En el nostre cas, es tractaria d'obtenir primer totes les traduccions. Això es pot aconseguir aplicant el Mètode dels Esdeveniments de la manera descrita en el capítol 5. A continuació, caldrà refusar aquelles solucions que violin alguna restricció d'integritat. Aquesta comprovació es podria realitzar, per exemple, mitjançant el mètode presentat a [Oli91]. Aquest mètode també està basat en el concepte d'esdeveniment i, per tant, es pot aplicar directament.

Considerem una altra vegada la base de dades anterior i suposem que es demana la mateixa actualització de vista. Veiem quin és el resultat obtingut amb aquest segon enfocament.

El primer que s'ha de fer és obtenir totes les traduccions que satisfan la inserció de $\text{Treb}(\text{laia})$. T és una traducció si hi ha una derivació constructiva des de $(\{\neg \text{Treb}(\text{laia})\} \emptyset \emptyset)$ fins a $(\square T C)$. Aquesta derivació es gairebé la mateixa de la figura 7.1, però eliminant el literal $\neg \text{Ic}$ de tots els objectius. Per tant, després del pas 3 s'assoleix la clàusula buida i finalitza el procés de traducció. En aquest cas, s'obté una única traducció $T = \{\neg \text{Cont}(\text{laia})\}$.

A continuació, cal comprovar que la traducció obtinguda satisfaci les restriccions d'integritat. No és difícil adonar-se del fet que aquesta traducció $T = \{\neg \text{Cont}(\text{laia})\}$ viola la restricció Ic1 ja que, si fos aplicada, la Laia tindria contracte i estaria a l'atur al mateix temps. Per tant, en aquest enfocament es refusa la traducció i no s'obté cap solució.

En aquest exemple, s'ha pogut comprovar que l'enfocament de la imposició de les restriccions d'integritat presenta alguns avantatges importants respecte l'enfocament descrit en aquesta secció. Per un costat, representa un guany en *eficiència* ja que les solucions inconsistentes són refusades durant la seva generació. Per l'altre, també hi ha una millora de l'*eficàcia* ja que, com s'ha pogut veure a l'exemple anterior, en alguns casos s'obtenen solucions que no són generades si es considera el segon enfocament. Aquest resultat és especialment interessant perquè assenyala la conveniència d'incloure la satisfacció de les restriccions d'integritat en el procés de traducció sempre que això sigui possible.

Per acabar aquest capítol, convé remarcar que el Mètode dels Esdeveniments permet tractar també *restriccions d'integritat de transició*. Aquestes restriccions d'integritat imposen condicions en més d'un estat de la base de dades. És a dir, són restriccions que han de satisfer les transicions. Un exemple de restricció d'integritat d'aquest tipus podria ser: "els salaris dels empleats no poden disminuir". Vegeu [Oli90, Oli91] pels detalls de les regles de transició i d'esdeveniment en aquest cas.

8. Característiques addicionals del mètode

8.1 Modificacions de vista

En els capítols anteriors hem considerat que les actualitzacions de vista permeses en el Mètode dels Esdeveniments es reduïen a insercions i a esborrats. En aquesta secció veurem que el nostre mètode també pot gestionar peticions de modificació de vista. Convé remarcar que, malgrat haver-se fet alguna proposta per traduir peticions d'aquest tipus en bases de dades relacionals [FSS79, DB82, Kel85, Dat86], cap dels mètodes proposats fins el moment actual en bases de dades deductives admet aquest tipus d'actualitzacions.

Entenem una *modificació de vista* com una petició per reemplaçar el valor d'algun atribut d'un fet vista per un altre valor diferent. De fet, una modificació de vista es pot entendre com un esborrat d'un fet vista seguit per una inserció d'un altre fet en el mateix predicat vista, on ha estat substituït el valor dels atributs desitjats. El significat natural del concepte d'esdeveniment ens permet definir un objectiu on totes aquestes accions hi estiguin considerades. L'exemple següent serveix per il·lustrar aquesta idea.

Exemple 8.1: Sigui D una base de dades deductiva que conté dos predicats bàsics (EP i PL) i un predicat derivat (EPL). Un fet bàsic EP(e,p) indica que l'empleat e treballa en el projecte p i un fet bàsic PL(p,l) ens diu que el projecte p està localitzat a la població l. El predicat vista EPL ens diu, donat un empleat, quina és la seva localització per cada projecte en què treballa. Suposem que l'estat actual de la base de dades és el següent:

F.1	EP (anna, p1)
F.2	PL (p1, vic)
F.3	PL (p2, vic)
DR.1	EPL (E,P,L) \leftarrow EP (E,P) \wedge PL (P,L)

Les regles de transició i d'esdeveniment associades a aquesta base de dades són:

Els passos 1, 2, 4, 5 i 6 de la branca esquerra són passos de resolució SLDNF on les clàusules d'A(D) actuen com a clàusules d'entrada. En els passos 3 i 7, l'esdeveniment bàsic seleccionat s'inclou en el conjunt de traducció T. En el pas 8, la condició $\leftarrow \delta PL(p2, vic)$ s'afegeix al conjunt de condició C per a garantir l'èxit de la derivació constructiva.

Un cop s'assoleix la clàusula buida, s'obté la traducció $T_1 = \{\delta PL(p1, vic), \iota EP(anna, p2)\}$. És a dir, assignant l'Anna al projecte p2 i esborrant el fet que el projecte p1 estigui localitzat a Vic se satisfarà l'actualització de vista demanada.

L'explicació dels passos de la branca dreta és exactament la mateixa que en el cas anterior. Un cop acabada la derivació constructiva, s'obté la traducció $T_2 = \{\delta EP(anna, p1), \iota EP(anna, p2)\}$. És a dir, l'actualització de vista també se satisfarà canviant l'assignació de l'Anna del projecte p1 al p2.

A partir de les derivacions resultants de seleccionar la clàusula I.3 en el pas 4 de qualsevol de les dues branques anteriors, s'obtenen dues traduccions més: $T_3 = \{\delta PL(p1, vic), \iota EP(anna, p5), \iota PL(p5, vic)\}$ i $T_4 = \{\delta EP(anna, p1), \iota EP(anna, p5), \iota PL(p5, vic)\}$, on p5 és el nom d'un projecte proporcionat per l'usuari. Aquestes derivacions no es mostren a la figura anterior. Observi's que les traduccions obtingudes es corresponen amb les solucions que satisfan la petició de modificació de la vista EPL.

8.2 Prevenció d'efectes secundaris en altres vistes

A causa de les regles deductives, quan es modifica una base de dades segons les traduccions obtingues es poden induir actualitzacions no demanades sobre altres vistes. És a dir, la base de dades pot no haver canviat completament d'acord amb la petició de l'usuari. Quan es manifesta aquest fet diem que es produeixen *efectes secundaris* en altres vistes. Els efectes secundaris són indesitjables i s'haurien d'evitar sempre que es pugui.

El Mètode dels Esdeveniments proporciona un mecanisme per a prevenir els efectes secundaris. Aquests es poden si s'indiquen els predicats vista pels quals no volem que es produeixin insercions i/o esborrats induïts. Les traduccions obtingudes satisfaran l'actualització de vista demanada i no induiran cap inserció ni cap esborrat pel conjunt de predicats vista donats.

L'objectiu arrel s'ha de modificar per a incorporar la prevenció d'efectes secundaris en el procés de traducció. L'objectiu arrel es defineix com $\{\leftarrow u \wedge \neg \iota P_1 \wedge \dots \wedge \neg \iota P_n \wedge \neg \delta Q_1 \wedge \dots \wedge \neg \delta Q_m\}$, on u és l'actualització de vista demanada i per a tot i (resp. j), $1 \leq i \leq n$ (resp. $1 \leq j \leq m$), $\neg \iota P_i$ (resp. $\neg \delta Q_j$) indica que no volem que es produeixin insercions induïdes (resp. esborrats induïts) sobre la

vista P_i (resp. Q_j). No ens ha de sorprendre que l'enfocament seguit per prevenir efectes secundaris sigui similar al tractament que es fa per satisfer les restriccions d'integritat. De fet, una violació d'una restricció d'integritat és un cas particular d'efecte secundari sobre predicats d'inconsistència.

Les traduccions s'obtenen a partir de la derivació constructiva que té com a objectiu inicial $\{\leftarrow u \wedge \neg \imath P_1 \wedge \dots \wedge \neg \imath P_n \wedge \neg \delta Q_1 \wedge \dots \wedge \neg \delta Q_m\}$. Durant aquesta derivació, els literals $\neg \imath P_i$ i $\neg \delta Q_j$ seran seleccionats. Com que es tracta d'esdeveniments derivats negatius (cas A7), s'assolirà el pas següent de la derivació constructiva si existeix una derivació de consistència des de $\{\{\leftarrow \imath P_i\} \text{ o } \{\leftarrow \delta Q_j\} \text{ T C}\}$ fins a $\{\{\} \text{ T' C'}\}$, on T i C són els conjunts de traducció i condició ja determinats quan aquests literals siguin seleccionats, respectivament. L'existència d'aquesta derivació equival a assegurar que no s'induirà cap actualització de vista no demanada quan s'apliqui la traducció. Per tant, la prevenció d'efectes secundaris s'inclou com una part de la derivació de consistència.

Exemple 8.2: Sigui D una base de dades deductiva que conté dos predicats bàsics (Prof i Asist) i dos predicats derivats (Clas i Aten). Un fet bàsic Prof(x) indica que x és un professor i un fet bàsic Asist(x) ens diu que x és un analista de sistemes. El predicat vista Clas ens indica qui fa classes, entenent que tots els professors fan classes. El predicat vista Aten ens indica totes les persones que tenen horari d'atenció als alumnes, on tant els professors com els analistes de sistemes en tenen. Suposem que l'estat actual de la base de dades és el següent:

F.1 Prof (roser)
 DR.1 Clas (X) \leftarrow Prof (X)
 DR.2 Aten (X) \leftarrow Prof (X)
 DR.3 Aten (X) \leftarrow Asist (X)

Algunes regles de transició i d'esdeveniment associades a aquesta base de dades són:

T.1 Aten'_{1,1} (X) \leftarrow Prof (X) \wedge $\neg \delta$ Prof (X)
 T.2 Aten'_{1,2} (X) \leftarrow \imath Prof (X)
 T.3 Aten'_{2,1} (X) \leftarrow Asist (X) \wedge $\neg \delta$ Asist (X)
 T.4 Aten'_{2,2} (X) \leftarrow \imath Asist (X)
 T.5, 6 Aten'₁ (X) \leftarrow Aten'_{1,j} (X) j = 1, 2
 T.7, 8 Aten'₂ (X) \leftarrow Aten'_{2,j} (X) j = 1, 2
 D.1 δ Clas (X) \leftarrow δ Prof (X)
 D.2 δ Aten (X) \leftarrow δ Prof (X) \wedge \neg Aten'₂ (X)
 D.3 δ Aten (X) \leftarrow δ Asist (X) \wedge \neg Aten'₁ (X)

Suposem que volem esborrar que la Roser és una professora i volem mantenir cert el fet que la Roser té horari d'atenció als alumnes. És a dir, volem actualitzar la vista Clas i impedir que es

produueixin efectes secundaris sobre l'individu Roser a la vista Aten. Les traduccions que satisfan aquesta petició s'obtenen a partir de la derivació constructiva que té com a objectiu arrel $\{\leftarrow \delta \text{Clas}(\text{roser}) \wedge \neg \delta \text{Aten}(\text{roser})\}$.

Aquesta derivació es mostra a la figura 8.2, on les etiquetes encerclades són referències a les regles del mètode definides a la secció 5.3.

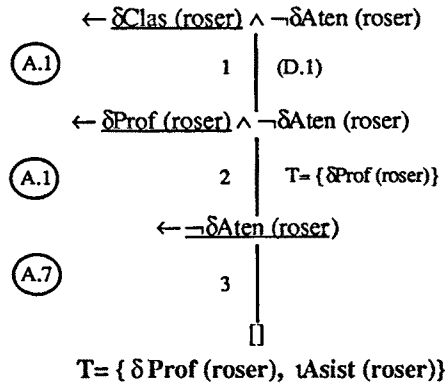


Figura 8.2

El pas 1 és un pas de resolució SLDNF on les clàusules d'A(D) actuen com a clàusules d'entrada. En el pas 2, l'esdeveniment bàsic $\delta \text{Prof}(\text{roser})$ és seleccionat i s'inclou en el conjunt de traducció T. En el pas 3 se selecciona el literal $\neg \delta \text{Aten}(\text{roser})$. S'assolirà l'objectiu següent de la derivació constructiva si hi ha una derivació de consistència des de $(\delta \text{Aten}(\text{roser}) \text{ T C})$ fins a $(\{\} \text{ T}' \text{ C}')$. Una part d'aquesta derivació constructiva es mostra a la figura 8.3.

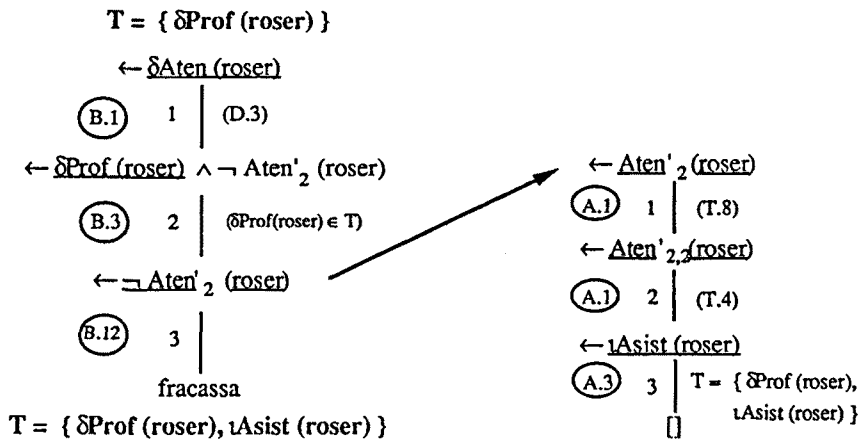


Figura 8.3

Els passos 1 i 2 són passos de resolució SLDNF on les clàusules d'A(D) i les de T actuen com a clàusules d'entrada. La derivació de consistència de la figura 8.3 finalitzarà en el pas 3 si hi ha una derivació constructiva associada al literal $\{\leftarrow \text{Aten}'_2(\text{roser})\}$. Aquesta derivació es mostra a la part inferior dreta de la figura anterior i provoca que l'esdeveniment bàsic $\text{Asist}(\text{roser})$ s'inclouï en el conjunt de traducció T. Observi's que aquesta inclusió ha estat provocada per la necessitat de no produir efectes secundaris.

Un cop fracassada la derivació de consistència (figura 8.3), la derivació constructiva inicial (figura 8.2) assoleix la clàusula buida i, per tant, finalitza el procés de traducció. Aleshores, s'obté la traducció $T = \{\delta \text{Prof}(\text{roser}), \text{Asist}(\text{roser})\}$. Per tant, l'actualització de vista demanada se satisfarà sense produir efectes secundaris especificats si esborrem el fet $\text{Prof}(\text{roser})$ i inserim el fet $\text{Asist}(\text{roser})$. És a dir, el canvi en la qualificació professional de la Roser de professora a analista de sistemes farà que deixi de fer classes però que segueixi mantenint l'horari d'atenció als alumnes.

8.3 Anotació de regles

Com ja hem esmentat anteriorment, un dels problemes que presenta l'actualització de vistes és el de l'existència de múltiples traduccions que satisfan una actualització. Aquest fet denota ambigüïtat semàntica, en el sentit que manca informació addicional per a escollir la traducció més adient.

Tomasic [Tom88] suggereix una tècnica molt interessant per a reduir el nombre de traduccions obtingudes. Aquesta tècnica, anomenada *anotació de regles*, ha estat descrita al capítol 1 d'aquesta tesi i permet que el dissenyador expressi informació mitjançant simples anotacions de les regles i dels predicats. Aquestes anotacions s'usen posteriorment per guiar el procés de traducció. La idea general consisteix a explorar únicament aquelles branques que permeten les anotacions.

L'anotació de regles proposada per Tomasic es pot adaptar per reduir el nombre de traduccions obtingudes pel Mètode dels Esdeveniments. En aquest mètode, les regles d'esdeveniment d'inserció i les d'esborrat s'utilitzen per traduir peticions d'actualització de vista. Aleshores, tant les anotacions d'inserció com les d'esborrat es poden expressar mitjançant una fletxa " \rightarrow " que apunti el cap de la regla d'esdeveniment anotada.

Exemple 8.3: Considerem la base de dades deductiva següent:

F.1	Q
F.2	R
F.3	S
DR.1	$P \leftarrow Q \wedge R \wedge S$

Les regles d'esdeveniment d'esborrat associades a aquesta base de dades són:

- D.1 $\delta P \leftarrow Q \wedge R \wedge \delta S$
- D.2 $\delta P \leftarrow Q \wedge \delta R \wedge S$
- D.3 $\rightarrow \delta P \leftarrow \delta Q \wedge R \wedge S$

En aquest exemple, la vista P té una única anotació d'esborrat en la clàusula D.3. Això vol dir que les peticions d'esborrat d'aquesta vista s'intentaran satisfer usant únicament aquesta clàusula. En cas que això no sigui possible, no s'obtindrà cap solució.

Suposem que es demana esborrar el fet P d'aquesta base de dades. Les traduccions que satisfan aquesta petició s'obtenen a partir de la derivació constructiva associada a $\{\leftarrow \delta P\}$, però en aquest cas es resol únicament amb aquelles regles d'esdeveniment que estan anotades. Aquesta derivació es mostra a la figura 8.4, on les etiquetes encerclades són referències a les regles del mètode definides a la secció 5.3.

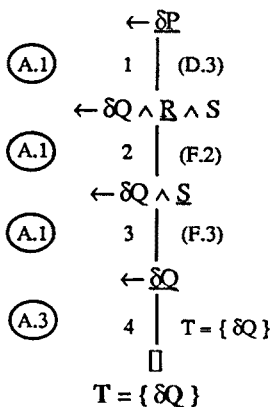


Figura 8.4

En el pas 1 es resol únicament amb la clàusula D.3 perquè aquesta és l'única clàusula anotada. Els passos 2 i 3 són passos de resolució SLDNF on les clàusules d'A(D) actuen com a clàusules d'entrada. En el pas 4, l'esdeveniment bàsic seleccionat, δQ , s'inclou en el conjunt de traducció T.

Un cop finalitzada la derivació constructiva s'ha obtingut una única traducció: $T = \{ \delta Q \}$. O sigui, l'actualització de vista es pot satisfer únicament esborrant el fet Q de la base de dades. Si totes les regles d'esdeveniment d'esborrat haguessin estat anotades, s'haurien obtingut les traduccions $T_1 = \{ \delta S \}$, $T_2 = \{ \delta R \}$ i $T_3 = \{ \delta Q \}$. Així doncs, en aquest exemple es veu com l'anotació de regles permet reduir el nombre de solucions obtingudes.

En el Mètode dels Esdeveniments, l' anotació de regles presenta un avantatge respecte a la proposada per Tomasic. A causa de les regles d'esdeveniment, en el nostre mètode es poden realitzar tantes anotacions d'inserció com regles d'esdeveniment d'inserció hi hagi. En canvi, el mètode de Tomasic permet únicament una anotació d'inserció per cada regla deductiva. Per tant, en el nostre mètode es pot ser més precís a l'hora d'efectuar les anotacions.

9. Implementació

La implementació que s'ha realitzat comprèn dos mòduls diferents: el *Conversor* i el *Traductor*. El mòdul *Conversor* genera les regles de transició i d'esdeveniment a partir d'una base de dades deductiva qualsevol segons el procediment descrit al capítol 3. O sigui, transforma automàticament una base de dades deductiva D en la seva base de dades augmentada $A(D)$. La implementació del mòdul *Conversor* ha estat realitzada per M. A. Delhez [Del91] i s'ha utilitzat el llenguatge Pascal del VAX.

L'altre mòdul, el *Traductor*, és el més important. El mòdul *Traductor* s'encarrega de traduir una actualització de vista en tants conjunts de traducció T_i (formats per esdeveniments bàsics) com possibilitats hi hagi de satisfer la petició. Per tant, aquest mòdul es correspon pròpiament a la implementació del Mètode dels Esdeveniments presentat en aquesta tesi. El mòdul *Traductor* és un metaintèrpret per sobre del Pop/Prolog en un VAX. La seva implementació és molt semblant a la formalització del mètode descrita a la secció 5.3. El codi complet d'aquest mòdul es pot trobar a l'Apèndix d'aquesta tesi.

La integració del mòdul *Conversor* i del *Traductor* permet disposar d'una eina per traduir actualitzacions de vista en bases de dades deductives. Donada una base de dades D , el mòdul *Conversor* obté automàticament la base de dades augmentada $A(D)$. D'aquesta manera, el Mètode dels Esdeveniments es pot aplicar directament. Aquesta conversió s'efectua en temps de compilació ja que les regles de transició i d'esdeveniment només depenen del contingut de la base de dades intensional.

En temps d'execució, l'usuari efectuarà peticions d'actualització de vista sobre aquesta base de dades. Aleshores, el mòdul *Traductor* obté totes les traduccions mínimes que satisfan l'actualització demanada. Per a fer-ho, utilitza la base de dades augmentada $A(D)$ generada pel *Conversor* en temps de compilació. El funcionament d'ambdós mòduls es descriu a la figura següent:

Temps de compilació



Temps d'execució

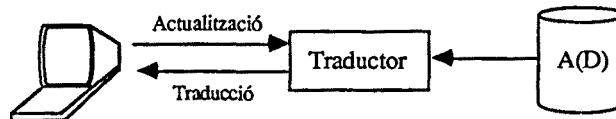


Figura 9.1

La implementació del Mètode dels Esdeveniments es correspon al mòdul Traductor. Per aquest motiu, en aquest capítol només es descriu la implementació d'aquest mòdul. El capítol s'estructura de la manera següent. A la secció 8.1 es presenten els aspectes generals de la implementació. A la secció 8.2 s'explica com s'ha implementat la derivació constructiva. Finalment, a la secció 8.3 es descriu la implementació de la derivació de consistència.

9.1 Aspectes generals

Una base de dades deductiva es representa com un conjunt de clàusules escrites en llenguatge Prolog.

Exemple 9.1: la base de dades deductiva següent

```
Atur (laia)
Treb (X) ← Cont (X)
Ic1 (X) ← Cont (X) ∧ Atur (X)
```

es representa com:

```
atur (laia).
treb (X) :- cont (X).
ic1 (X) :- cont (X), atur (X).
```

Les regles d'esdeveniment i de transició també es representen com a clàusules escrites en llenguatge Prolog. Per a representar els esdeveniments hem escollit la convenció següent:

- un esdeveniment bàsic ιP (resp. δP) es representa per $\text{ins}(p)$ (resp. $\text{del}(p)$).
- un esdeveniment derivat ιQ (resp. δQ) es representa per iq (resp. dq).

Aquesta distinció de nomenclatura ens ha permès simplificar la implementació, ja que els únics literals que es tracten de manera especial són els esdeveniments bàsics.

Exemple 9.2: Sigui R un predicat bàsic i P i Q dos predicats derivats. Les regles de transició següents:

$$\begin{aligned} P'_{1,1} &\leftarrow Q \wedge \neg\delta Q \wedge R \wedge \neg\delta R \\ P'_{1,2} &\leftarrow Q \wedge \neg\delta Q \wedge \iota R \\ P'_{1,3} &\leftarrow \iota Q \wedge R \wedge \neg\delta R \\ P'_{1,4} &\leftarrow \iota Q \wedge \iota R \end{aligned}$$

es representen com:

$$\begin{aligned} p_{11} &:- q, \text{not}(\text{dq}), r, \text{not}(\text{del}(r)). \\ p_{12} &:- q, \text{not}(\text{dq}), \text{ins}(r). \\ p_{13} &:- \text{iq}, r, \text{not}(\text{del}(r)). \\ p_{14} &:- \text{iq}, \text{ins}(r). \end{aligned}$$

Un cop comentats tots els aspectes referents a la representació de les clàusules en la nostra implementació, anem a descriure el funcionament del procediment de traducció. Sigui u una petició d'actualització de vista. L'avaluació del predicat $\text{translate}(u)$ produeix com a resultat totes les traduccions mínimes que satisfan l'actualització de vista demanada. Esquemàticament, el funcionament d'aquest predicat és el següent:

- Obtenir una traducció T , fent que la derivació constructiva associada a l'actualització de vista demanada tingui èxit.
- Verificar que la solució obtinguda satisfaci totes les condicions. Aquesta verificació pot provocar la inclusió de nous esdeveniments bàsics en el conjunt de traducció T .
- Estudiar si la solució obtinguda està subsumida per alguna traducció trobada anteriorment. Si és així, la traducció actual no és mínima i per tant no ha de ser tinguda en compte.
- Estudiar si la solució obtinguda subsumeix alguna traducció anterior. Caldrà prescindir de totes les traduccions trobades anteriorment que siguin subsumides per la traducció actual ja que no són traduccions mínimes.
- Repetir el procés fins que s'hagin obtingut totes les solucions.
- Presentar per pantalla les traduccions mínimes obtingudes.

Hi ha dos aspectes importants que cal destacar. En primer lloc, la verificació de condicions es realitza com un únic pas al final del procés de derivació, en comptes de fer-se per cada inclusió d'un esdeveniment bàsic en el conjunt de traducció T tal com s'havia definit en el capítol 5.

Aquesta optimització permet que el procés de verificació sigui més eficient, ja que les condicions només es comproven una vegada en comptes de verificar-se tants cops com inclusions d'esdeveniments bàsic en T hi hagi. El predicat *check_conds* (*T*, *C*, *T'*) efectua la verificació de les condicions, essent T i C els conjunts de traducció i condició i T' la traducció resultant de la verificació. Recordeu que el procés de verificació de condicions pot provocar noves inclusions en T.

El segon aspecte important que cal destacar és que l'eliminació de les traduccions que no són mínimes la realitza el predicat *study_subsumption*(*Soluc*, *List*). Una traducció mínima és aquella que no està subsumida per cap altra traducció. És a dir, no existeix cap subconjunt d'ella que també sigui una traducció. *Study_subsumption* (*Soluc*, *List*) estudia si la traducció *Soluc* subsumeix o és subsumida per alguna traducció obtinguda anteriorment (contingudes a la llista *List*).

La part més interessant de la implementació és la que es refereix al procés d'obtenció de traduccions. En el Mètode dels Esdeveniments, les traduccions s'obtenen fent que la derivació constructiva associada a la petició d'actualització de vista tingui èxit. La implementació de la derivació constructiva es descriu a la propera secció.

9.2 Implementació de la derivació constructiva

Ens hem basat en el predicat *solve*(*A*) [SS86] per implementar la derivació constructiva. Aquest metaintèrpret ha estat proposat per simular el model computacional de la resolució SLDNF i, segons Sterling i Shapiro, la seva granularitat és la més adient per a gestionar la majoria d'aplicacions. El seu codi complet es mostra a la figura següent:

```
solve (true).
solve ((A,B)) :- solve(A), solve(B).
solve (A) :- clause(A,B), solve(B).
solve (not(A)) :- not (solve(A)).
```

Figura 9.2

Declarativament, aquest intèrpret es llegeix de la manera següent. La constant *true* és certa. La conjunció (A,B) és certa si A i B també ho són. Un objectiu A és cert si hi ha una clàusula A :- B en

el programa interpretat i B és cert. Un objectiu $\text{not}(A)$ és cert si l'objectiu A no ho és (negació per fracàs finit).

Sigui u una actualització de vista. En el Mètode dels Esdeveniments, una traducció T s'obté fent que una derivació fracassada de $A(D) \cup \{\leftarrow u\}$ tingui èxit. Per tant, no ens ha de sorprendre que la implementació escollida sigui una extensió d'aquest metaintèrpret. Més concretament, en el nostre mètode una traducció T és un conjunt d'esdeveniments bàsics en què existeix una derivació constructiva des de $(\leftarrow u \ \emptyset \ \emptyset)$ fins a $([] \ T \ C)$.

El predicat $\text{solve}(\text{Goal}, T_i, T_f, C_i, C_f)$ realitza les derivacions constructives. El paràmetre Goal es correspon a l'objectiu actual de la derivació, T_i i C_i són els conjunts de traducció i condició inicials i T_f i C_f els conjunts de traducció i condició resultants de l'aplicació d'una de les regles A1), ..., A7) definides a la secció 5.3.

L'objectiu inicial de la derivació constructiva es correspon a la petició d'actualització de vista. Per tant, T és una traducció vàlida si s'obté com a resultat de l'avaluació del predicat $\text{solve}(u, [], T, [], C)$. Observi's que això és equivalent a dir que hi ha una derivació constructiva des de $(\leftarrow u \ \emptyset \ \emptyset)$ fins a $([] \ T \ C)$.

La nostra implementació estén el predicat $\text{solve}(A)$ [SS86] en dues direccions diferents. D'una banda, convé destacar el tractament particular que hem d'efectuar quan el literal seleccionat és un esdeveniment bàsic. Això està motivat per la necessitat d'aconseguir que una derivació SLDNF finitament fracassada tingui èxit. De l'altra, també és diferent el tractament de l'arbre subsidiari associat als literals negatius, ja que en el nostre cas aquests arbres es corresponen a les derivacions de consistència. La implementació d'aquestes derivacions es descriu detalladament a la propera secció.

El codi complet de la derivació constructiva es presenta a l'Apèndix d'aquesta tesi. Convé, però, fer algunes puntualitzacions importants:

- Com ja hem dit, el procés de comprovació de les condicions s'efectua com un únic pas al final de la derivació constructiva. Per tant, les regles A3) i A4) no incorporen aquesta comprovació.

- Quan el literal seleccionat és un esdeveniment bàsic positiu que no està plenament instanciat (cas A.4.1), cal instanciar-lo per a delimitar exactament l'esdeveniment al que es refereix. Hem suposat que aquesta instanciació la realitza l'usuari assignant valors a les variables del literal seleccionat.

9.2 Implementació de la derivació de consistència

Quan en la derivació constructiva se selecciona un literal negatiu ' $\neg L_i$ ', corresponent a un predicat nou o d'esdeveniment, s'assoleix l'objectiu següent si existeix una derivació de consistència des de $(\leftarrow L_i \ T \ C)$ fins a $(\{ \} \ T' \ C')$, on T i C són els subconjunts de traducció i de condició determinats quan L_i és seleccionat. També s'utilitza la derivació de consistència per comprovar que el conjunt de traducció T satisfaci totes les condicions de C.

El predicat *do_fail*(Goal, T_i , C_i , T_f , C_f) realitza les derivacions de consistència. El paràmetre Goal es correspon a l'objectiu inicial de la derivació, T_i i C_i són els conjunts de traducció i condició inicials i T_f i C_f els conjunts de traducció i condició resultants de l'aplicació d'una de les regles B1), ... , B13) definides a la secció 5.3.

El codi complet del predicat *do_fail* es pot trobar a l'Apèndix d'aquesta tesi. En aquest cas es fa difícil identificar clarament les regles B1), ... , B13) a causa de tres motius principals:

- En alguns casos, es pot prescindir d'alguna regla (per exemple, B10) ja que la comprovació de condicions es realitza únicament al final de la derivació constructiva.

- En el codi s'hi inclou el fet que cal tenir en compte totes les possibilitats de fer fracassar una conjunció $\leftarrow L_1, \dots, L_k$.

- S'utilitza un predicat auxiliar *fer_fallar_llista* quan es vol efectuar la derivació de consistència d'un conjunt d'objectius.

10. Comparació amb els mètodes de bases de dades deductives

En aquest capítol es compara exhaustivament el Mètode dels Esdeveniments amb els altres mètodes proposats fins el moment actual per l'actualització de vistes en bases de dades deductives. Tots aquests mètodes tracten amb el mateix tipus de base de dades que el nostre. Tanmateix, cap d'ells admet peticions de modificació de vista, ni inclou la comprovació de restriccions d'integritat de transició, ni pot prevenir efectes secundaris en altres vistes.

Aquest capítol s'estructura de la manera següent. A la secció 10.1 el Mètode dels Esdeveniments es compara amb el mètode de Decker. A la secció 10.2 es compara amb el mètode de Guessoum i Lloyd. Finalment, a la secció 10.3 es compara amb el mètode de Kakas i Mancarella. Tots aquests mètodes han estat descrits en el capítol 1 d'aquesta tesi.

10.1 Mètode de Decker [Dec90a, Dec90b]

En el mètode de Decker les traduccions s'obtenen a partir de derivacions SLDNF, de la manera següent:

Esborrat d'una vista: Sigui D una base de dades deductiva i u una petició d'esborrat d'una vista. Les traduccions que satisfan aquesta petició s'obtenen fent que fracassin les diferents refutacions SLDNF de $D \cup \{\leftarrow u\}$. Això es pot aconseguir esborrant un fet usat com a clàusula d'entrada o bé requerint la inserció d'un àtom d'un literal negatiu seleccionat durant la derivació.

Inserció en una vista: En peticions d'inserció, Decker defineix el concepte d'arbres d'actualització de vistes per a què la funció de selecció usada en la resolució SLDNF no impedeixi l'obtenció d'algunes traduccions. Informalment, la idea bàsica d'aquests arbres consisteix a seleccionar i resoldre cada literal de cada objectiu amb cada clàusula d'entrada candidata. Les

traduccions s'obtenen fent que un objectiu fracassat d'aquest arbre tingui èxit. Això es pot aconseguir afegint una instància de cada literal positiu que apareix en aquest objectiu i requerint l'esborrat de l'àtom de tots els literals negatius que hi apareixen.

Hi ha quatre diferències importants entre el Mètode dels Esdeveniments i el mètode de Decker. Totes elles es descriuen detalladament a les properes seccions.

10.1.1 Solucions invalidades a causa de la negació

El problema principal que presenta el mètode de Decker és que algunes de les solucions obtingudes poden ser invalidades degut a la negació. És a dir, pot obtenir traduccions que no satisfan l'actualització de vista demanada. Per tant, en presència de negació aquest mètode obté traduccions *possibles*. Una traducció possible serà *vàlida* sempre i quan la base de dades actualitzada satisfaci la petició. Això s'ha de validar executant la petició un cop modificada la base de dades, la qual cosa implica un decrement considerable en l'eficiència d'aquest mètode.

L'exemple següent serveix per il·lustrar aquest fet.

Exemple 10.1: Considerem una base de dades deductiva D que conté els següents fets i regles deductives, on Q , R i S són predicats bàsics.

F.1	$Q(a)$
F.2	$R(a)$
DR.1	$P(X) \leftarrow Q(X) \wedge R(X) \wedge \neg S(X)$
DR.2	$S(a) \leftarrow Q(a)$

Suposem que volem inserir $P(a)$ en aquesta base de dades. Observi's que no hi ha cap traducció que satisfaci aquesta actualització de vista. En el mètode de Decker, les traduccions s'obtenen a partir de l'arbre SLDNF de $D \cup \{\leftarrow P(a)\}$. Aquest arbre es mostra en la figura següent:

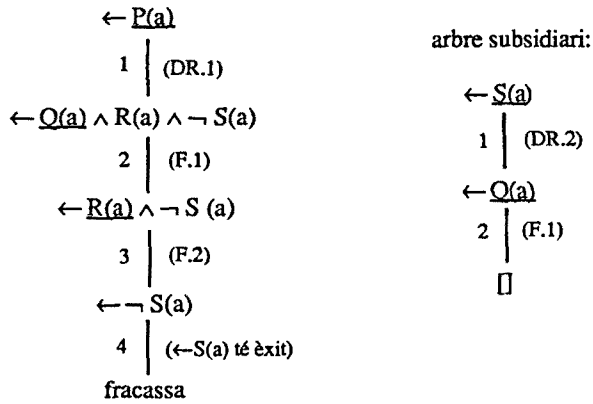


Figura 10.1

La derivació de l'esquerra diu que $P(a)$ fracassa perquè $S(a)$ té èxit. De l'arbre subsidiari es dedueix que $S(a)$ té èxit perquè $Q(a)$ és cert en l'estat actual de la base de dades. Per tant, esborrant $Q(a)$ aconseguirem que l'arbre subsidiari fracassi i així l'arbre principal tindrà èxit. És a dir, en aquest mètode s'obté la traducció $T = \{\text{esborrar } Q(a)\}$.

No obstant, l'esborrat de $Q(a)$ no satisfà la inserció de $P(a)$. És a dir, la solució obtinguda no és correcta. Aquest fet es coneix amb el nom d'invalidació de solucions a causa de la negació. En l'exemple anterior, això s'ha produït perquè la clàusula $Q(a)$ també s'utilitza en la derivació principal. El mètode de Decker no és capaç de detectar aquest fet.

Afortunadament, aquest problema no es presenta en el nostre mètode. Per comprovar-ho aplicarem el Mètode dels Esdeveniments en aquest mateix exemple. Algunes regles de transició i d'esdeveniment corresponents a la base de dades anterior són:

- T.1 $P'_{1,1}(X) \leftarrow Q(X) \wedge \neg \delta Q(X) \wedge R(X) \wedge \neg \delta R(X) \wedge \neg S(X) \wedge \neg \delta S(X)$
- T.2 $P'_{1,2}(X) \leftarrow Q(X) \wedge \neg \delta Q(X) \wedge R(X) \wedge \neg \delta R(X) \wedge \delta S(X)$
- L.1..7 $\delta P(X) \leftarrow P'_{1,j}(X) \quad j = 2 \dots 8$
- D.1 $\delta S(a) \leftarrow \delta Q(a)$

En el Mètode dels Esdeveniments, T satisfà l'actualització de vista si hi ha una derivació constructiva des de $(\{\leftarrow P(a)\} \emptyset \emptyset)$ fins a $(\square T C)$. Aquesta derivació es mostra a la figura 10.2, on les etiquetes encerclades són referències a les regles del mètode definides a la secció 5.3.

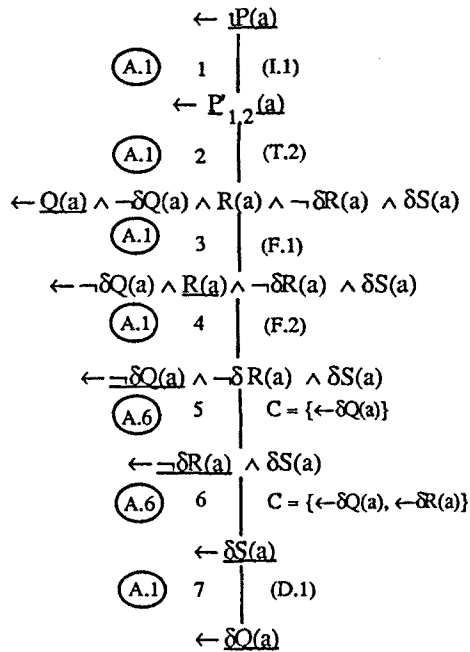


Figura 10.2

Els passos 1 fins el 4 i el 7 són passos de resolució SLDNF on les clàusules d'A(D) actuen com a clàusules d'entrada. En els passos 5 i 6, s'afegeixen condicions a C per a poder verificar que noves inclusions en el conjunt de traducció T garanteixen l'existència de la derivació constructiva. En aquest cas, no existeix cap derivació constructiva perquè l'esdeveniment bàsic seleccionat en el pas 8 viola la consistència d'alguna de les condicions de C.

Selecciónt les altres regles d'esdeveniment d'inserció en el pas 1 de la figura 10.2, tampoc s'obté cap derivació constructiva. Per tant, el nostre mètode no obté cap traducció, la qual cosa vol dir que l'actualització de vista demanada no es pot satisfer canviant únicament la base de dades extensional. Observi's que aquesta és l'única resposta correcta.

10.1.2 Necessitat d'iteració

En alguns casos, el mètode de Decker s'ha d'iterar un cert nombre de vegades per obtenir traduccions vàlides. El problema és que, en general, no se sap quantes iteracions s'han de realitzar. Per aquest motiu, Decker suggereix que només es realitzi una única passada del mètode, amb el que és possible que no s'obtinguin totes les solucions vàlides. L'exemple següent serveix per il·lustrar aquest fet.

Exemple 10.2: Considerem una base de dades deductiva D que conté únicament les següents regles de derivació, on R i S són predicats bàsics.

- DR.1 $P \leftarrow R \wedge \neg Q$
 DR.2 $Q \leftarrow R \wedge \neg S$

En el mètode de Decker, les traduccions que satisfan aquesta actualització s'obtenen a partir de l'arbre d'actualització de vista de $D \cup \{\leftarrow P\}$ que es mostra a la figura 10.3.



Figura 10.3

La derivació principal fracassa perquè l'objectiu $\leftarrow R$ fracassa. Així, afegint R a la base de dades es pot aconseguir que aquesta derivació tingui èxit. És a dir, s'obté la traducció {inserir R }. Malauradament, aquesta solució és invalidada a causa de la negació ja que la inserció de R provoca que l'arbre subsidiari tingui èxit i, per tant, que la derivació principal fracassi.

Es podria modificar la base de dades, obtenint $D' = D \cup \{R\}$, i iterar la petició d'inserir P sobre la nova base de dades D' . Aleshores, a partir de l'arbre d'actualització de $D' \cup \{\leftarrow P\}$ s'obté la traducció {inserir S } (aquesta derivació no es mostra en l'exemple). Efectuant aquesta modificació sobre D' , el nou estat de la base de dades esdevé $D'' = D' \cup \{S\}$. Observi's que l'actualització de vista demanada si que se satisfà en aquest nou estat. Per tant, una traducció vàlida és {inserir R , inserir S }. Aquesta traducció no es pot obtenir en un únic pas del mètode de Decker.

Aquest problema no es presenta en el nostre mètode. Per a comprovar-ho, l'aplicarem en aquest mateix exemple. Les regles de transició i d'esdeveniment rellevants en aquest cas són:

- T.1 $P'_{1,1} \leftarrow R \wedge \neg \delta R \wedge \neg Q \wedge \neg i Q$
 T.2 $P'_{1,2} \leftarrow R \wedge \neg \delta R \wedge \delta Q$
 T.3 $P'_{1,3} \leftarrow i R \wedge \neg Q \wedge \neg i Q$
 T.4 $P'_{1,4} \leftarrow i R \wedge \delta Q$
 T.5 $Q'_{1,1} \leftarrow R \wedge \neg \delta R \wedge \neg S \wedge \neg i S$
 T.6 $Q'_{1,2} \leftarrow R \wedge \neg \delta R \wedge \delta S$
 T.7 $Q'_{1,3} \leftarrow i R \wedge \neg S \wedge \neg i S$
 T.8 $Q'_{1,4} \leftarrow i R \wedge \delta S$
- I.1...3 $i P \leftarrow P'_{1,j} \quad j = 2 \dots 4$
 I.4...6 $i Q \leftarrow Q'_{1,j} \quad j = 2 \dots 4$

T és una traducció si existeix una derivació constructiva des de $(\{\leftarrow \neg P\} \emptyset \emptyset)$ fins a $(\square \text{ T C})$. Aquesta derivació es mostra a la figura 10.4, on les etiquetes encerclades són referències a les regles del mètode definides a la secció 5.3.

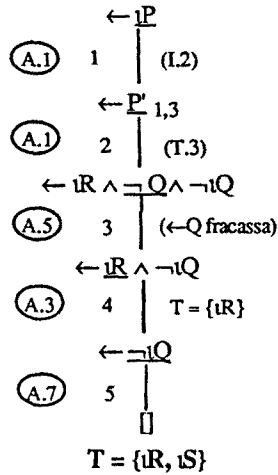


Figura 10.4

Els passos 1 fins el 3 són passos de resolució SLDNF on les clàusules d'A(D) actuen com a clàusules d'entrada. En el pas 4, l'esdeveniment bàsic seleccionat s'inclou en el conjunt de traducció T. En el pas 5, s'assoleix la clàusula buida perquè existeix la derivació de consistència associada a $\{\leftarrow \neg Q\}$. Part d'aquesta derivació es mostra a la figura 10.5.

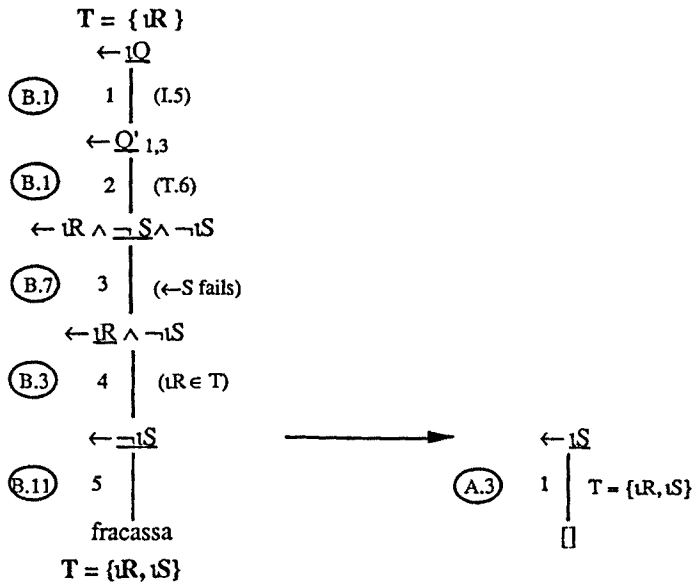


Figura 10.5

Els passos 1 fins el 4 són passos de resolució SLDNF on les clàusules d'A(D) i les de T actuen com a clàusules d'entrada. En el pas 5, existirà la derivació de consistència si hi ha una derivació constructiva associada al literal $\{\leftarrow \iota S\}$. Aquesta derivació es mostra a la part inferior dreta de la figura anterior i provoca la inclusió de l'esdeveniment bàsic ιS en el conjunt de traducció T.

Les derivacions de consistència resultants de seleccionar les clàusules I.4 i I.6 en el pas 1 de la figura 10.5 també existeixen, però no modifiquen el contingut del conjunt de traducció T ni el del conjunt de condició C . Aquestes derivacions no es mostren en l'exemple.

Un cop fracassat l'arbre de la figura 10.5, la derivació constructiva inicial (figura 10.4) assoleix la clàusula buida i, per tant, finalitza el procés de traducció. Aleshores, s'obté la traducció $T = \{\iota R, \iota S\}$. Observi's que aquesta traducció satisfà l'actualització de vista demanada i que s'ha obtingut amb un únic pas del Mètode dels Esdeveniments.

10.1.3 Satisfacció de les restriccions d'integritat

En el mètode de Decker, la satisfacció de les restriccions d'integritat s'ha de realitzar com un pas addicional, independent de l'actualització de vistes. En canvi, en el Mètode dels Esdeveniments, es pot incorporar en el procés de traducció de la manera descrita en el capítol 7. Els avantatges d'aquest darrer enfocament ja han estat mencionats en el mateix capítol.

10.1.4 Actualitzacions de vista múltiples i mixtes

Finalment, en el mètode de Decker es processen de manera diferent les peticions d'inserció i les d'esborrat. Això impedeix que aquest mètode pugui tractar actualitzacions de vista múltiples i mixtes, com ara inserir P i esborrar Q. En canvi, el processament uniforme de qualsevol petició permet que aquests tipus d'actualitzacions es puguin traduir amb Mètode dels Esdeveniments.

10.2 Mètode de Guessoum i Lloyd [GL90a, GL90b]

Com ja hem esmentat en el capítol 1, el procediment emprat pel mètode de Guessoum i Lloyd és molt semblant al que segueix el mètode de Decker. Per aquest motiu, els problemes que presenten un i altre mètode són gairebé els mateixos. Malgrat aquest fet, en aquesta secció comparem el Mètode dels Esdeveniments amb el mètode de Guessoum i Lloyd fent referència als inconvenients del mètode de Decker descrits a la secció 10.1.

A diferència del mètode de Decker, el mètode de Guessoum i Lloyd efectua una comprovació final per a obtenir únicament les traduccions vàlides. Una traducció serà vàlida si, un cop actualitzada la base de dades, l'arbre SLDNF corresponent fracassa finitament (esborrat d'una vista) o bé si hi ha una refutació SLDNF associada a l'actualització demanada (inserció en una vista). Gràcies a aquesta comprovació, aquest mètode no presenta el problema de *solucions invalidades a causa de la negació*. Convé remarcar que el cost d'aquesta verificació serà molt elevat ja que, en general, es realitzaran tants accessos a la base de dades com accessos s'hagin efectuat durant el procés de traducció. En canvi, el Mètode dels Esdeveniments no presenta aquest problema sense necessitat d'efectuar cap comprovació final.

La segona diferència és la més important. En peticions d'inserció, Decker defineix els arbres d'actualització de vista per poder obtenir totes les traduccions impedides per la funció de selecció utilitzada per la resolució SLDNF. En canvi, Guessoum i Lloyd no proposen cap alternativa a aquest problema. Per aquest motiu, en general aquest mètode no obté totes les traduccions que satisfan una inserció en una vista. L'exemple 1.3 del capítol 1 serveix per il·lustrar aquest fet.

Finalment, de la mateixa manera que el mètode de Decker, el mètode de Guessoum i Lloyd no admet actualitzacions de vista múltiples i mixtes, no permet que la satisfacció de les restriccions d'integritat s'incorpori en el procés de traducció i en alguns casos ha de ser iterat un cert nombre de vegades per obtenir traduccions correctes. En la secció anterior es pot trobar una descripció detallada d'aquests problemes.

10.3 Mètode de Kakas i Mancarella [KM90a, KM90b]

El mètode de Kakas i Mancarella emmarca el problema de l'actualització de vistes en un enfocament abductiu. Es distingeixen clarament dues etapes. En la primera, l'actualització de vista es tradueix en diferents conjunts Δ_i . Cada Δ_i és una especificació d'un conjunt de requeriments que la base de dades extensional (EDB) hauria de satisfer. En la segona etapa, s'obté una transacció que permet actualitzar Δ_i sobre l'EDB.

Aquest mètode presenta bàsicament dos inconvenients. El primer és que pot realitzar treball innecessari en la primera etapa ja que no té en compte el contingut de la base de dades extensional. El segon és que, malgrat integrar la satisfacció de les restriccions d'integritat en el procés de traducció, en alguns casos pot no detectar la violació d'alguna restricció. Aquests problemes es descriuen detalladament a les seccions 10.3.1 i 10.3.2, respectivament.

10.3.1 Treball innecessari

El primer problema d'aquest mètode és que pot realitzar algun treball innecessari en la primera etapa ja que no té en compte el contingut de la base de dades extensional per a obtenir els diferents conjunts de requeriments Δ_i . Aleshores, es poden obtenir Δ_i que expressen requeriments que la base de dades ja satisfà, amb la qual cosa el nombre de Δ_i obtinguts pot esdevenir molt gran. Aquest fet pot incrementar considerablement la quantitat de treball innecessari. L'exemple següent serveix per il·lustrar aquest problema.

Exemple 10.3: Considerem la següent base de dades, on Q, R, A, B, C i D són predicats bàsics:

- F.1 Q
- F.2 R
- DR.1 $P \leftarrow Q \wedge R$
- DR.2 $P \leftarrow S \wedge T$
- DR.3 $S \leftarrow A \wedge B$
- DR.4 $T \leftarrow C \wedge D$

i suposem que l'actualització de vista demanada és l'esborrat de P.

L'espai de cerca abductiu de la primera etapa d'aquest mètode es mostra a la figura 10.6.

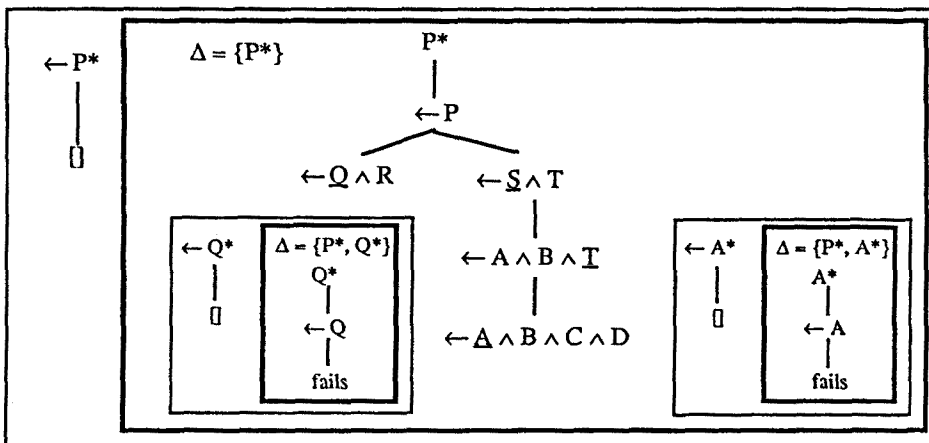


Figura 10.6

Els subconjunts de l'espai de cerca enquadrats en negreta mostren la comprovació de la consistència de les noves hipòtesis que s'afegeixen a Δ . Aquestes derivacions no depenen d'una regla de selecció concreta ja que, en general, cal explorar totes les possibilitats de fer fracassar una conjunció $\leftarrow L_1 \wedge \dots \wedge L_k$.

En la derivació de consistència associada a P, s'assoleixen els objectius $\{\leftarrow Q \wedge R\}$ i $\{\leftarrow A \wedge B \wedge C \wedge D\}$. Aleshores, cal tenir en compte totes les possibilitats per fer fracassar aquests objectius. A la figura 10.6 només s'ha expressat una de les diverses alternatives.

A la primera etapa del mètode de Kakas i Mancarella s'obtenen vuit conjunts Δ_i diferents si es consideren totes les possibilitats:

$$\begin{array}{ll} \Delta_1 = \{P^*, Q^*, A^*\} & \Delta_5 = \{P^*, R^*, A^*\} \\ \Delta_2 = \{P^*, Q^*, B^*\} & \Delta_6 = \{P^*, R^*, B^*\} \\ \Delta_3 = \{P^*, Q^*, C^*\} & \Delta_7 = \{P^*, R^*, C^*\} \\ \Delta_4 = \{P^*, Q^*, D^*\} & \Delta_8 = \{P^*, R^*, D^*\} \end{array}$$

on S^* denota que el fet S no ha de ser cert en el nou estat de la base de dades. No obstant, només hi ha dues solucions que satisfan l'actualització de vista demanada: {esborrar Q} i {esborrar R}. Per tant, el treball innecessari relitzat en aquest cas ha estat considerable.

El Mètode dels Esdeveniments té en compte el contingut de la base de dades a l'hora d'obtenir les traduccions. Per aquest motiu, només s'obtenen aquelles solucions que són coherents amb l'estat actual de la base de dades, no realitzant-se gens de treball innecessari. Apliquem el nostre mètode en aquest mateix exemple. Les regles de transició i d'esdeveniment rellevants en aquest cas són:

- D.1 $\delta P \leftarrow Q \wedge \delta R$
- D.2 $\delta P \leftarrow \delta Q \wedge R$
- D.3 $\delta P \leftarrow S \wedge \delta T$
- D.4 $\delta P \leftarrow \delta S \wedge T$

Les traduccions que satisfan l'esborrat de P s'obtenen a partir de la derivació constructiva associada a l'objectiu $\{\leftarrow \delta P\}$. Aquesta derivació es mostra a la figura 10.7, on les etiquetes encerclades són referències a les regles del mètode definides a la secció 5.3.

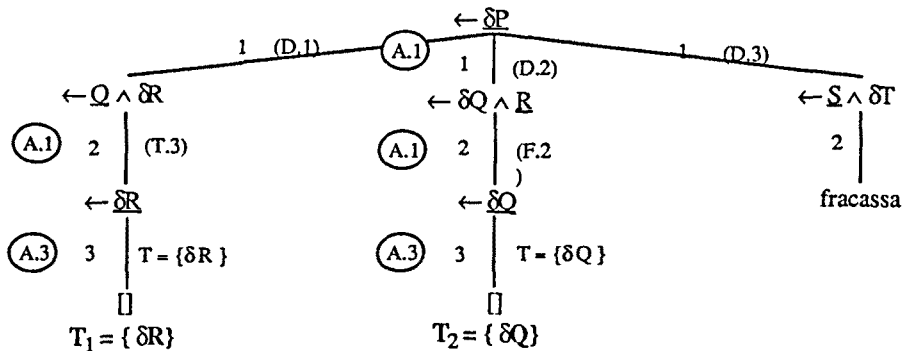


Figura 10.7

En la branca dreta, no hi ha derivació constructiva perquè el literal seleccionat en el pas 2 no és cert en la base de dades actual. Seleccionant la regla D.4 en el pas 1 de la figura anterior s'obté una derivació que fracassa pel mateix motiu (aquesta derivació no es mostra en l'exemple).

A partir de les branques de l'esquerra i central s'obtenen dues traduccions: $T_1 = \{\delta Q\}$ i $T_2 = \{\delta R\}$. És a dir, l'actualització de vista demanada es pot satisfer esborrant el fet Q i esborrant el fet R. Observi's que les traduccions obtingudes es corresponen a les úniques solucions vàlides.

10.3.2 Satisfacció de les restriccions d'integritat

L'altra diferència important la trobem en la satisfacció de les restriccions d'integritat. Tant el Mètode dels Esdeveniments com el mètode de Kakas i Mancarella incorporen aquesta verificació durant el procés de traducció, amb els avantatges que això comporta. El problema és que aquest segon mètode no detecta totes les violacions de restriccions d'integritat. L'exemple següent serveix per il·lustrar aquesta fet.

Exemple 10.4: Considerem la següent base de dades deductiva, on Q i R són predicats bàsics:

- DR.1 $S(X) \leftarrow Q(X)$
- DR.2 $P(X) \leftarrow Q(X)$
- IC.1 $\text{Ic1}(X) \leftarrow P(X) \wedge \neg R(X)$

i suposem que es demana la inserció de S(a). En la figura 10.8 es mostra la derivació abductiva que es realitzaria en la primera etapa del mètode de Kakas i Mancarella.

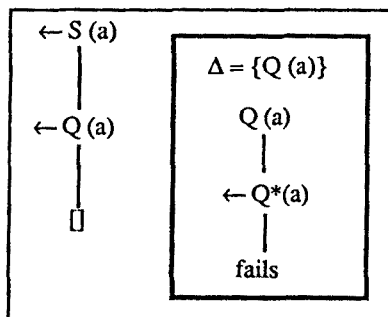


Figura 9.8

En la primera etapa s'obté el conjunt $\Delta = \{Q(a)\}$. Aquest requeriment es pot satisfer en la segona etapa inserint el fet Q(a) en la base de dades extensional. Per tant, s'obté la traducció {inserir Q(a)}. Observi's que aquesta solució satisfà l'actualització de vista demanada però viola la restricció d'integritat.

Afortunadament, aquest problema no es presenta en el Mètode dels Esdeveniments. Per comprovar-ho aplicarem el nostre mètode en aquest mateix exemple. Les regles de transició i d'esdeveniment rellevants en aquest cas són:

- T.1 $S'_{1,1}(X) \leftarrow Q(X) \wedge \neg\delta Q(X)$
- T.2 $S'_{1,2}(X) \leftarrow \imath Q(X)$
- T.3 $P'_{1,1}(X) \leftarrow Q(X) \wedge \neg\delta Q(X)$
- T.4 $P'_{1,2}(X) \leftarrow \imath Q(X)$
- T.5 $Ic1'_{1,1}(X) \leftarrow P(X) \wedge \neg\delta P(X) \wedge \neg R(X) \wedge \neg\imath R(X)$
- T.6 $Ic1'_{1,2}(X) \leftarrow P(X) \wedge \neg\delta P(X) \wedge \delta R(X)$
- T.7 $Ic1'_{1,3}(X) \leftarrow \imath P(X) \wedge \neg R(X) \wedge \neg\imath R(X)$
- T.8 $Ic1'_{1,4}(X) \leftarrow \imath P(X) \wedge \delta R(X)$
- I.1 $\imath S(X) \leftarrow S'_{1,2}(X)$
- I.2 $\imath P(X) \leftarrow P'_{1,2}(X)$
- I.3..5 $\imath Ic1(X) \leftarrow Ic1'_{1,j}(X) \quad j = 2 \dots 4$
- I.6 $\imath Ic \leftarrow \imath Ic1(X)$

T és una traducció que satisfà l'actualització de vista demanada si hi ha una derivació constructiva des de $\{\leftarrow \imath S(a) \wedge \neg \imath Ic\} \emptyset \emptyset$ fins a $(\square \text{ T C})$. Aquesta derivació es mostra a la figura 10.9, on les etiquetes encerclades són referències a les regles del mètode definides a la secció 5.3.

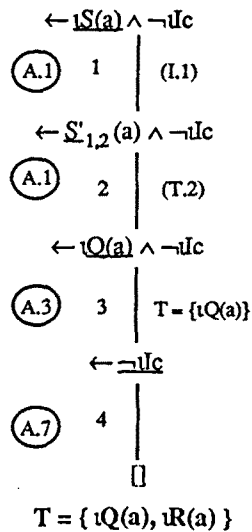


Figura 10.9

Els passos 1 i 2 són passos de resolució SLDNF on $A(D)$ actua com a conjunt d'entrada. En el pas 3, l'esdeveniment bàsic seleccionat s'inclou en el conjunt de traducció T . En el pas 4 se selecciona el literal $\neg \text{Ic}$. S'assolirà el següent pas de la derivació constructiva si existeix la derivació de consistència associada al literal $\{\leftarrow \text{Ic}\}$. Part d'aquesta derivació es mostra a la figura següent:

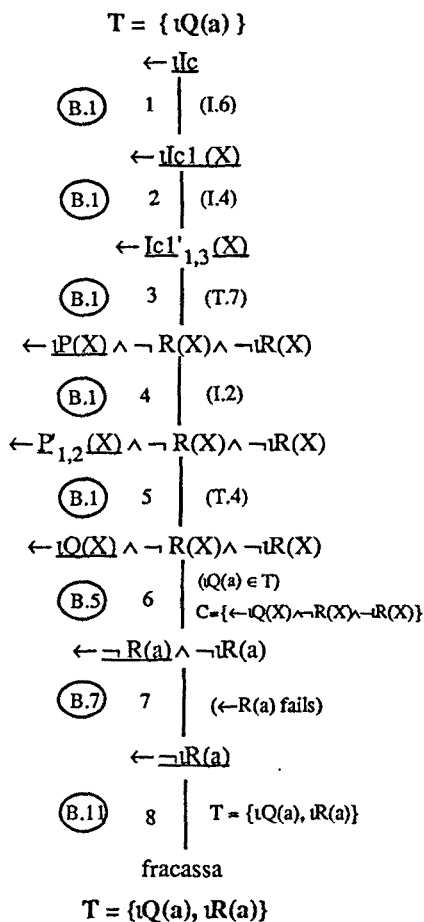


Figura 10.10

Els passos 1 fins el 5 són passos de resolució SLDNF. En el pas 6 es resol amb el conjunt de traducció T i s'inclou l'objectiu actual en el conjunt de condició C . En el pas 8 se selecciona el literal $\neg \text{I}R(a)$. Per a què acabi la derivació de consistència, $\text{I}R(a)$ s'ha d'incloure en el conjunt de traducció T . Això s'aconsegueix en la derivació constructiva associada a $\{\leftarrow \text{I}R(a)\}$ (aquesta derivació no es mostra en l'exemple).

Un cop finalitzada la derivació constructiva inicial de la figura 10.9, s'obté la traducció $T=\{1Q(a), 1R(a)\}$. Observi's que aquesta traducció satisfà tant l'actualització de vista demanada com la restricció d'integritat.

Finalment, el mètode de Kakas i Mancarella no permet gestionar restriccions d'integritat de transició, mentre que el Mètode dels Esdeveniments si que ho permet.

11. Lligam amb els antecedents en bases de dades relacionals

El gran nombre de propostes realitzades i la diferència existent entre les bases de dades relacionals i les deductives dificulta que el Mètode dels Esdeveniments pugui ser comparat exhaustivament amb els mètodes proposats per l'actualització de vistes en el camp relacional. Malgrat aquest fet, sembla interessant establir algun tipus de relació entre el nostre mètode i els antecedents en aquest camp.

Com ja hem esmentat en el capítol 1, els mètodes proposats en bases de dades relacionals es poden classificar bàsicament en tres grans enfocaments: *complement d'una vista*, *traducció segons el tipus de vista* i *hipòtesi de la relació universal*. Cadascun d'ells es correspon a una filosofia diferent d'entendre l'actualització de vistes. Per tant, és interessant conèixer quines característiques d'aquests enfocaments es poden aplicar en el Mètode dels Esdeveniments i de quina manera s'ha de portar a terme aquesta aplicació. Aquest és, concretament, l'objectiu d'aquest capítol: establir un lligam entre el nostre mètode i els enfocaments proposats en bases de dades relacionals.

Aquest capítol s'estructura de la manera següent. A la secció 10.1 es tracta l'enfocament del complement d'una vista, a la secció 10.2 el de traducció segons el tipus de vista i a la secció 10.3 el de la hipòtesi de la relació universal. Tots tres enfocaments han estat descrits en el capítol 1 d'aquesta tesi.

11.1 Complement d'una vista

L'enfocament del complement d'una vista [BS81] consisteix en associar a cada vista un *complement* que descriu la informació no visible en la vista. Així doncs, el complement és la informació que cal donar, conjuntament amb la vista, per a calcular tota la base de dades. En aquest context, una actualització de vista es tradueix de tal manera que només es canvia la vista, mentre que el seu complement es manté invariant.

Malgrat presentar alguns inconvenients importants (com conèixer el complement d'una vista?, com saber si un complement és mínim?, etc.), gairebé tothom coincideix a destacar la rellevància d'aquest enfocament. Per tant, fóra interessant que aquesta idea es pogués integrar d'alguna manera en el Mètode dels Esdeveniments.

L'interrogant que se'ns planteja en aquest cas és veure si això serà possible i, en cas que ho sigui, conèixer com es pot efectuar la integració. De fet, mantenir el complement invariant es pot entendre com la prevenció de qualsevol efecte secundari sobre la vista escollida com a complement. Per tant, adaptant el marc descrit en el capítol 8 per la prevenció d'efectes secundaris es pot aconseguir que el Mètode dels Esdeveniments segueixi l'enfocament del complement d'una vista.

Sigui $P(X)$ una vista i $Q(Y)$ el seu complement, on X i Y representen vectors de variables. El primer que cal fer és definir la vista complement de $P(X)$. Això es pot aconseguir mitjançant la declaració de la regla deductiva: $CP \leftarrow Q(Y)$, on CP és el nom que es dona a aquest complement. Cal tenir en compte que, en general, caldrà definir tants complements com vistes de la base de dades puguin ser actualitzades.

A més, s'ha de modificar l'objectiu arrel per a integrar la verificació d'invariància del complement en el procés de traducció. L'objectiu arrel es defineix com $\{\leftarrow u \wedge \neg iCP \wedge \neg \delta CP\}$, on u és l'actualització de vista demanada i $\neg iCP$ i $\neg \delta CP$ indiquen que no volem que es produeixin insercions ni esborrats sobre la vista complement (és a dir, volem que el complement es mantingui invariant).

Aleshores, les traduccions s'obtenen trobant una derivació constructiva que té com a objectiu inicial $\{\leftarrow u \wedge \neg iCP \wedge \neg \delta CP\}$. Durant aquesta derivació, els literals $\neg iCP$ i $\neg \delta CP$ seran seleccionats. Com que es tracta d'esdeveniments derivats negatius (cas A7), s'assolirà el següent pas de la derivació constructiva si existeix una derivació de consistència des de $\{\leftarrow iCP\}$ o $\{\leftarrow \delta CP\}$ fins a $\{\}$ i C , on T i C són, respectivament, els conjunts de traducció i de condició ja determinats quan aquests literals són seleccionats. L'existència de les derivacions de consistència assegura que el complement es manté invariant. El següent exemple serveix per il·lustrar aquestes idees.

Exemple 11.1: Sigui D una base de dades deductiva que conté dos predicats bàsics (ED i DM) i un predicat derivat (EDM). Un fet bàsic $ED(e,d)$ indica que e és un empleat del departament d i un fet bàsic $DM(d,m)$ ens diu que m és el director del departament d . El predicat vista EDM ens diu donat un empleat, qui és el seu director per cada departament. Suposem que l'estat actual de la base de dades és el següent:

- F.1 $ED(\text{pol}, d1)$
- F.2 $DM(d1, \text{anna})$

$$\text{DR.1 } \text{EDM}(E,D,M) \leftarrow \text{ED}(E,D) \wedge \text{DM}(D,M)$$

Els dos predicats bàsics ED i DM poden ser complements de la vista EDM. Suposem que, per algun motiu determinat, el complement escollit en aquest cas és DM. El primer que s'ha de fer és definir la vista complement. Això es pot fer de la manera següent:

$$\text{DR.2 } \text{Cedm} \leftarrow \text{DM}(D,M)$$

on Cedm és el nom que volem donar a la vista complement. Algunes regles de transició i d'esdeveniment associades a aquesta base de dades són:

$$\text{T.1 } \text{Cedm}'_{1,1} \leftarrow \text{DM}(D,M) \wedge \neg \delta \text{DM}(D,M)$$

$$\text{T.2 } \text{Cedm}'_{1,2} \leftarrow \neg \text{DM}(D,M)$$

$$\text{I.1 } \neg \text{Cedm} \leftarrow \text{Cedm}'_{1,2}$$

$$\text{D.1 } \delta \text{EDM}(E,D,M) \leftarrow \text{DM}(D,M) \wedge \delta \text{ED}(E,D)$$

$$\text{D.2 } \delta \text{EDM}(E,D,M) \leftarrow \text{ED}(E,D) \wedge \delta \text{DM}(D,M)$$

$$\text{D.3 } \delta \text{Cedm} \leftarrow \delta \text{DM}(D,M) \wedge \neg \text{Cedm}'_{1,1} \wedge \neg \text{Cedm}'_{1,2}$$

Suposem que volem esborrar el fet vista EDM(pol,d1,anna) seguint l'enfocament de mantenir invariant el complement. Com hem esmentat anteriorment, les traduccions que satisfan aquesta petició s'obtenen a partir de la derivació constructiva que té com a objectiu arrel $\{\leftarrow \delta \text{EDM}(\text{pol},d1,\text{anna}) \wedge \neg \text{Cedm} \wedge \neg \delta \text{Cedm}\}$. Part d'aquesta derivació es mostra a la figura 11.1, on les etiquetes encerclades són referències a les regles del mètode definides a la secció 5.3.

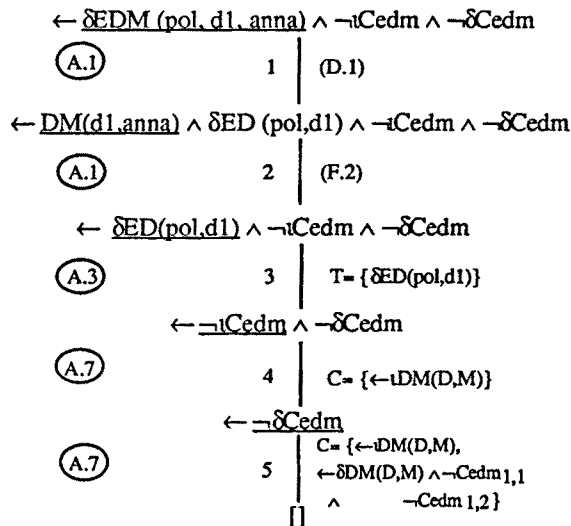


Figura 11.1

Els passos 1 i 2 d'aquesta derivació són passos de resolució SLDNF on les clàusules d'A(D) actuen com a clàusules d'entrada. En el pas 3, l'esdeveniment bàsic seleccionat s'inclou en el conjunt de traducció T. En els passos 4 i 5 existeixen les derivacions de consistència associades a la negació dels literals seleccionats. En ambdós casos aquestes derivacions provoquen la inclusió de noves condicions en C (aquestes derivacions no es mostren a la figura anterior).

Un cop finalitzada la derivació constructiva, s'obté la traducció $T = \{\delta ED(pol, d1)\}$. És a dir, l'actualització de vista es pot satisfer esborrant el fet que en Pol treballa al departament d1. Observi's que aquesta traducció satisfà l'actualització de vista i manté el complement invariant.

En canvi, si seleccionem la clàusula D.2 en el pas 1 de la figura anterior no s'obté cap derivació constructiva ja que no es pot obtenir cap traducció que mantingui el complement invariant. Això es mostra a la figura 11.2, on les etiquetes encerclades són referències a les regles del mètode definides a la secció 5.3.

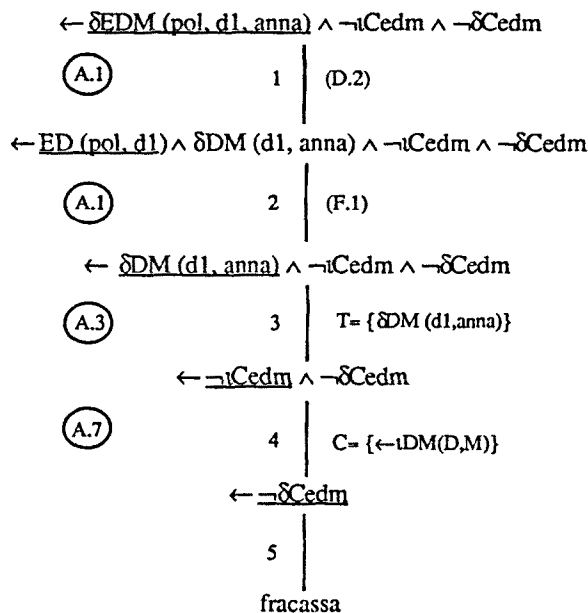


Figura 11.2

Els passos 1 i 2 són passos de resolució SLDNF on A(D) actua com a conjunt d'entrada. En el pas 3, l'esdeveniment bàsic seleccionat s'inclou en el conjunt de traducció T. En el pas 4, s'assoleix el següent objectiu de la derivació constructiva perquè existeix la derivació de consistència associada al literal $\neg \iota Cedm$. Aquesta derivació no es mostra en l'exemple però provoca la inclusió de la condició $\{\neg \iota DM(D, M)\}$ en C.

En el pas 5 se selecciona el literal $\neg\delta\text{Cedm}$. Com que no existeix la derivació de consistència associada a la negació d'aquest literal, la derivació constructiva de la figura 11.2 no pot acabar. Per tant, a partir d'aquesta branca no s'obté cap traducció. Observi's que la no existència d'aquesta derivació es correspon a una violació de la condició de mantenir invariant el complement. Més concretament, indica que s'ha produït algun esborrat d'algun fet de la vista complement. A la figura 11.3 es pot veure que no existeix la derivació de consistència associada a δCedm .

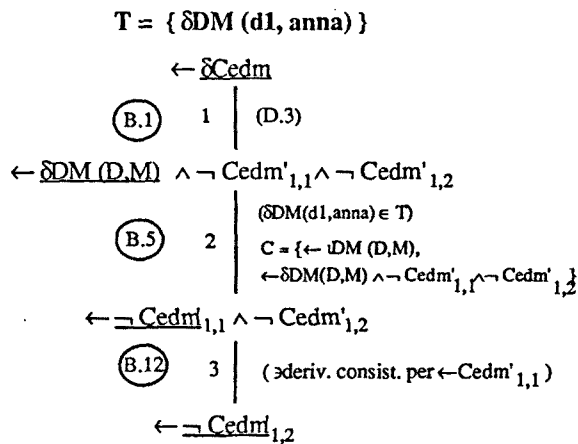


Figura 11.3

El pas 1 és un pas de resolució SLDNF on $A(D)$ actua com a conjunt d'entrada. En el pas 2 es resol amb el conjunt de traducció T i s'afegeix l'objectiu al conjunt de condició C . En els pas 3, hi ha una derivació de consistència associada a la negació del literal seleccionat. En el pas 4 el literal seleccionat és un predicat nou. Com que no hi ha cap derivació constructiva associada a aquest literal, la derivació de consistència del literal $\leftarrow\delta\text{Cedm}$ no existeix.

Per tant, en aquest exemple s'ha pogut comprovar que el Mètode dels Esdeveniments proporciona la possibilitat de seguir l'enfocament del complement d'una vista. Donada una vista, l'únic que cal fer és definir el seu complement i augmentar l'objectiu arrel. Aleshores, el Mètode dels Esdeveniments s'aplica de la manera habitual descrita en el capítol 5.

11.2 Traducció segons el tipus de vista

Una segona alternativa per resoldre el problema de l'actualització de vistes en bases de dades relacionals consisteix en definir per cada operador de definició de vista (projecció, unió, join, etc.) i tipus d'actualització (inserció, esborrat, modificació) quina és la seva traducció expressada en modificacions de les relacions bàsiques corresponents.

Qualsevol vista definida mitjançant els operadors relacionals clàssics es pot expressar amb regles deductives. No és difícil demostrar que, quan s'aplica el Mètode dels Esdeveniments a una d'aquestes vistes, les traduccions obtingudes pel nostre mètode es corresponen exactament a les traduccions proposades en el camp relacional. L'exemple següent, on s'actualitza una vista que es pot definir mitjançant l'operador diferència, serveix per il·lustrar aquest fet.

Exemple 10.2: Considerem la següent base de dades deductiva, on R i S són predicats bàsics.

- F.1 R(a)
- DR.1 $P(X) \leftarrow R(X) \wedge \neg S(X)$

Les regles d'esdeveniment rellevants en aquest cas són:

- D.1 $\delta P(X) \leftarrow R(X) \wedge \iota S(X)$
- D.2 $\delta P(X) \leftarrow \delta R(X) \wedge \neg S(X)$

Les traduccions que satisfan l'esborrat de P(a) s'obtenen a partir de la derivació constructiva que té com a objectiu arrel $\leftarrow \delta P(a)$. Aquesta derivació es mostra a la figura 11.4, on les etiquetes encerclades són referències a les regles del mètode definides a la secció 5.3.

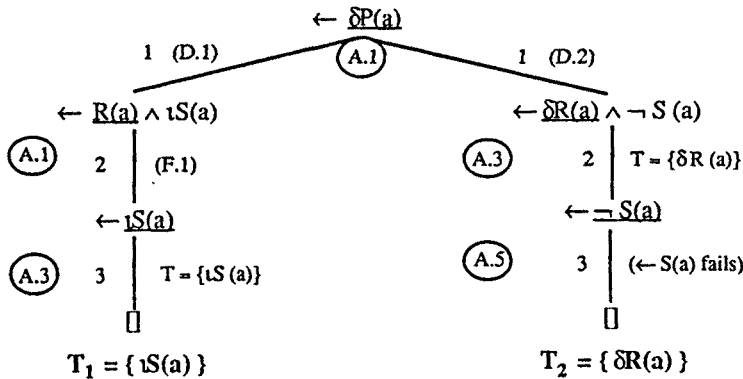


Figura 11.4

Els passos 1 i 2 de la branca esquerra són passos de resolució SLDNF on les clàusules d'A(D) actuen com a clàusules d'entrada. En el pas 3, el literal seleccionat $\iota S(a)$ s'inclou en el conjunt de traducció T.

En branca dreta, els passos 1 i 3 són passos de resolució SLDNF on A(D) actua com a conjunt d'entrada. En el pas 2, l'esdeveniment bàsic seleccionat $\delta R(a)$ s'inclou en el conjunt de traducció T.

Un cop finalitzada la derivació constructiva de la figura 11.4, s'han obtingut les traduccions $T_1 = \{\iota S(a)\}$ i $T_2 = \{\delta R(a)\}$. Observi's que aquesta solució es correspon a la solució expressada en

bases de dades relacionals per a traduir esborrats de vistes definides mitjançant l'operador diferència.

L'avantatge principal del Mètode dels Esdeveniments respecte els mètodes que segueixen aquest enfocament és que, a causa de la negació i la recursivitat, les regles deductives permeten expressar vistes que no es poden definir mitjançant els operadors relacionals clàssics. Per aquest motiu, el nostre mètode permet traduir vistes que no han estat mai considerades pels mètodes que segueixen l'enfocament de traducció segons el tipus de vista.

11.3 Hipòtesi de la relació universal

Segons la hipòtesi de la relació universal, l'usuari veu la base de dades com si fos una única relació definida sobre l'univers d'atributs. Per tant, quan l'usuari actualitza aquesta relació ho fa només en termes dels atributs, sense tenir coneixement de quin atribut apareix en quina relació.

A [Sag83] es proposa la *instància representativa* com una representació correcta de les dades que estan emmagatzemades en la base de dades. Quan l'usuari expressa una actualització de vista, ho fa especificant un subconjunt dels atributs d'aquesta instància. Aleshores, l'actualització demanada s'ha de traduir a modificacions de les relacions bàsiques.

El concepte d'*extension join* s'utilitza per calcular projeccions de la instància representativa. Per tant, coneixent quin 'extension join' s'usa per calcular determinada projecció d'aquesta instància es pot aplicar el Mètode dels Esdeveniments per efectuar la traducció.

Per seguir l'enfocament de la hipòtesi de la relació universal, s'ha de definir una vista per cada projecció de la instància representativa que pugui ser actualitzada. El cos de les regles deductives que defineixen aquestes vistes estarà format per l'extension join que defineix la projecció. Aleshores, el Mètode dels Esdeveniments s'aplicarà de la manera habitual descrita en el capítol 5. L'exemple següent serveix per il·lustrar aquesta idea.

Exemple 10.3: Considerem una base de dades que conté tres relacions bàsiques: $Q(X)$, $R(X, Y)$ i $S(Z)$, on els atributs subratllats es corresponen a les claus de cada relació, i suposem que el seu contingut actual és el següent:

F.1	Q (<u>a</u>)
F.2	R (<u>a</u> ,b)
F.3	S (<u>c</u>)

La instància representativa d'aquesta base de dades és:

X	Y	Z
a	b	δ_1
δ_2	δ_3	c

Sigui P el nom donat a la projecció d'aquesta instància representativa sobre els atributs X i Y. Aquesta projecció es pot calcular mitjançant l'extension join $Q[X = X]R$. Per tant, la vista $P(X, Y)$ corresponent a aquesta projecció es defineix com:

$$DR.1 \quad P(X, Y) \leftarrow Q(X) \wedge R(X, Y)$$

Les regles d'esdeveniment rellevants en aquest cas són:

$$D.1 \quad \delta P(X, Y) \leftarrow Q(X) \wedge \delta R(X, Y)$$

$$D.2 \quad \delta P(X, Y) \leftarrow \delta Q(X) \wedge R(X, Y)$$

Suposem que es vol esborrar el fet vista $P(a, b)$. T és una traducció que satisfà l'actualització de vista demanada si existeix una derivació constructiva des de $(\{\leftarrow \delta P(a, b)\} \emptyset \emptyset)$ fins a $(\square \mid T \ C)$. Aquesta derivació es mostra a la figura següent, on les etiquetes encerclades són referències a les regles del mètode definides a la secció 5.3.

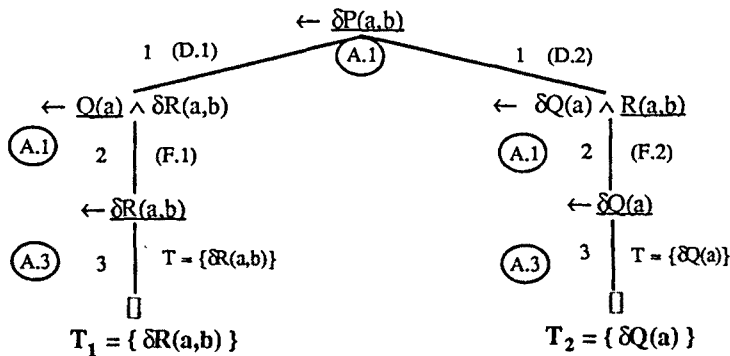


Figura 11.5

Els passos aplicats a ambdues branques són els mateixos. Els passos 1 i 2 són passos SLDNF on les clàusules d'A(D) actuen com a clàusules d'entrada. En el pas 3, l'esdeveniment bàsic seleccionat s'inclou en el conjunt de traducció T.

Un cop finalitzada la derivació constructiva, s'obtenen les traduccions $T_1 = \{\delta R(a, b)\}$ i $T_2 = \{\delta Q(a)\}$. O sigui, l'actualització de vista especificada com a projecció de la instància representativa es pot satisfer esborrant el fet $R(a, b)$ o bé esborrant el fet $Q(a)$. Per tant, en aquest exemple s'ha pogut comprovar que el Mètode dels Esdeveniments també és aplicable en el context de la hipòtesi de la relació universal. Donada una instància representativa, l'únic que cal fer és definir els extension join que calculen projeccions d'aquesta instància. Aleshores, el nostre mètode s'aplica de la manera habitual descrita en el capítol 5.

12. Conclusions i recerca futura

12.1 Conclusions

Les vistes permeten delimitar els continguts de la base de dades rellevants per a cada grup d'usuaris. Aquest fet presenta una sèrie d'avantatges com ara afavorir la independència lògica de les dades, simplificar la interfície amb l'usuari i proporcionar una mesura de protecció.

Les operacions que es realitzen sobre una vista s'han de traduir a operacions equivalents expressades sobre l'estructura de la base de dades. En un entorn ideal, l'usuari no hauria de distingir entre els fets bàsics emmagatzemats físicament i les vistes. Per tant, els llenguatges de consulta i els de manipulació s'haurien d'aplicar de la mateixa manera en ambdós casos.

El processament de les actualitzacions de vista planteja alguns problemes importants. Entre d'altres, es poden destacar l'existència de múltiples traduccions que satisfan una actualització de vista, els efectes secundaris i la satisfacció de les restriccions d'integritat. Per aquest motiu, els sistemes de gestió de bases de dades actuals no permeten assolir completament els avantatges que hem esmentat anteriorment.

En aquesta tesi s'ha presentat un nou mètode, anomenat Mètode dels Esdeveniments, per a l'actualització de vistes en bases de dades deductives. Aquest mètode augmenta la base de dades amb un conjunt de regles, anomenades regles de transició i d'esdeveniment, que defineixen explícitament les insercions i els esborrats induïts per una modificació de la base de dades. Mitjançant aquestes regles, s'utilitza la resolució SLDNF per a obtenir totes les traduccions mínimes que satisfan una actualització de vista.

Un dels resultats que cal destacar especialment d'aquesta tesi és la demostració del fet que el Mètode dels Esdeveniments és *complet* en bases de dades estratificades. És a dir, en aquest tipus de bases de dades el mètode obté totes les traduccions mínimes que satisfan una actualització de vista.

En el nostre context, una traducció és mínima si no està subsumida per cap altra traducció. Convé destacar que, fins el moment actual, no s'havia demostrat la completesa de cap mètode proposat per a l'actualització de vistes.

Una conseqüència directa d'aquest resultat és que, quan no s'obté cap traducció, l'actualització demanada no es pot satisfer canviant únicament la base de dades extensional. Aleshores, s'haurien de considerar modificacions de la base de dades intensional (insercions, supressions i modificacions de regles deductives, insercions de fets derivats, etc.).

A més de la demostració de completesa, hi ha altres aspectes del Mètode dels Esdeveniments que cal destacar. Probablement, l'avantatge principal del nostre mètode és la possibilitat d'*incorporar la satisfacció de les restriccions d'integritat* en el procés de traducció. D'aquesta manera s'obtenen traduccions que satisfan tant l'actualització de vista demanada com les restriccions d'integritat de la base de dades. A causa d'aquesta integració, en alguns casos s'obtenen solucions que no serien generades si es considerés la satisfacció de les restriccions d'integritat com una etapa addicional, independent de l'actualització de vistes. Cal tenir en compte que cap dels mètodes proposats fins el moment actual realitza correctament aquesta integració.

Hi ha altres avantatges importants. Per exemple, el tractament uniforme de les insercions i dels esborrats permet que el nostre mètode pugui gestionar *actualitzacions de vista múltiples* (més d'una actualització alhora) i *mixtes* (diferents tipus de petició). A més, disposa d'un mecanisme per a la *prevenció d'efectes secundaris*, amb el que es pot evitar que es produeixin actualitzacions no desitjades d'altres vistes. Finalment, el Mètode dels Esdeveniments admet *peticions de modificació* de vistes i pot incorporar una tècnica, anomenada *anotació de regles*, per reduir el nombre de traduccions obtingudes.

El darrer resultat remarcable d'aquesta tesi és la constatació del fet que el Mètode dels Esdeveniments es pot aplicar segons els tres grans enfocaments proposats per a l'actualització de vistes en bases de dades relacionals: complement d'una vista, traducció segons el tipus de vista i hipòtesi de la relació universal. Aquest fet és especialment important perquè aquests tres enfocaments es corresponen a tres filosofies diferents d'entendre l'actualització de vistes.

Per tots aquest motius, es pot concloure que el Mètode dels Esdeveniments té tot el poder dels mètodes proposats fins el moment actual per a l'actualització de vistes en bases de dades deductives, però no presenta algunes de les seves limitacions.

12.2 Recerca futura

Afortunadament, l'àmbit d'aplicació del Mètode dels Esdeveniments no es restringeix al problema de l'actualització de vistes. En bases de dades i en altres àrees d'investigació es poden trobar diversos problemes que tenen alguna similitud amb el tema tractat en aquesta tesi. Per tant, en un moment en el que sembla que diferents àrees de recerca tendeixen a convergir, seria interessant adaptar el nostre mètode per a resoldre aquests problemes.

En aquest sentit, en el camp de la intel·ligència artificial trobem el tema de la *revisió de les creences*. És a dir, l'estudi de quina manera s'ha de modificar una base de coneixement per incorporar les noves informacions que s'hagin pogut assimilar. Les analogies amb l'actualització de vistes són evidents i, per tant, el nostre mètode podria aportar quelcom en aquesta àrea.

Una altra aplicació del Mètode dels Esdeveniments podria ser per a la *reparació de bases de dades inconsistentes*. O sigui, es té una base de dades que viola alguna restricció d'integritat i es volen impulsar accions correctores per reparar-ne la seva consistència. Aquestes accions correctores es corresponen a modificacions de la base de dades. Gràcies a la riquesa semàntica del concepte d'esdeveniment, es podria convertir una petició de reparació d'una base de dades en una petició d'actualització de vista. Aleshores, s'ha d'estudiar de quina manera cal estendre el nostre mètode per resoldre aquest problema.

El Mètode dels Esdeveniments també es podria adaptar per gestionar el que en bases de dades s'anomena *control de consistència reactiu*. Aquest problema consisteix a aplicar mesures correctores quan una transacció viola alguna restricció d'integritat. Per tant, sembla que l'enfocament seguit per a la imposició de les restriccions d'integritat es podria utilitzar en aquest cas.

A part d'adaptar el Mètode dels Esdeveniments per resoldre tots aquests problemes, s'hauria d'investigar la completesa del nostre mètode en altres tipus de bases de dades, com per exemple les acícliques [Cav91]. L'avantatge d'aquest tipus de bases de dades és que garanteixen l'acabament del procés de traducció. Malauradament, hi ha un ampli ventall de bases de dades estratificades (per exemple la definició de la relació familiar d'avantpassat) que no són acícliques i, per aquest motiu, en aquesta tesi ens hem referit únicament a les estratificades. Demostrant la completesa del nostre mètode en altres tipus de bases de dades més generals, serà possible traduir correctament una gamma més àmplia d'actualitzacions de vista.

Finalment, cal esmentar que una tercera línia possible de recerca consisteix a resoldre el *problema d'ambigüitat semàntica*. És a dir, es necessiten criteris clars per escollir quina de les

múltiples solucions obtingudes és la més adient. Un cop es trobi una resposta a aquesta qüestió, es podrà considerar que l'actualització de vistes és un problema completament resolt.

Referències

- [ABW88] Apt, K.R.; Blair, H.A.; Walker, A. "Towards a Theory of Declarative Knowledge", in *Foundations of Deductive Databases and Logic Programming* (J. Minker Ed.), Morgan-Kaufman, 1988, pp. 89-148.
- [Ban79] Bancilhon, F. "Supporting View Updates in Relational Data Bases". In *Data Base Architecture* (Bracci and Nijssen Eds.), North Holland, Amsterdam, 1979.
- [BR86] Bancilhon, F; Ramakrishnan, R. "An Amateur's Introduction to Recursive Query Processing Strategies". Proc. ACM SIGMOD Int. Conf. on Management of Data, Washington D.C., 1986, pp. 16-152.
- [BS81] Bancilhon, F; Spyrtos, N. "Update Semantics of Relational Views", ACM Transactions on Database Systems, vol. 6, núm. 4, 1981, pp. 557-575.
- [BV88] Brodda, V.; Vossen, G. "Update and Retrieval in a Relational Database through a Universal Schema Interface", ACM Transactions on Database Systems, vol. 13, núm. 4, 1988, pp. 449-485.
- [CA79] Carlson, C.R.; Arora, A.K. "The Updatability of Relational Views Based on Functional Dependencies", Proc. 3rd Int. Computer Software and Applications Conf., IEEE Computer Society, Chicago, 1979, pp. 415-420.
- [CADF90] Committee for Advanced DBMS Function. "Third Generation Database Systems Manifesto", Proc. of the 1990 ACM SIGMOD Int. Conf. on Management of Data, Atlantic City 1990, p. 396.
- [Cav91] Cavedon, L. "Acyclic logic programs and the completeness of SLDNF resolution", Theoretical Computer Science, núm. 86, 1991, pp. 81-92.
- [CL87] Cavedon, L.; Lloyd, J. "A Completeness Theorem for SLDNF-Resolution". TR 87/9, Dept. of Computer Science, Univ. of Melbourne, Australia, 1987.
- [Cla78] Clark, K.L. "Negation as Failure", in *Logic and Data Bases* (Gallaire, H.; Minker, J. Eds.), Plenum Press, New York, 1978, pp. 293-322.

- [CO92] Costal, D.; Olivé, A. "A method for reasoning about deductive conceptual models of information systems", Proc. CAiSE 92, Manchester, 1992, pp. 612-631.
- [CP84] Cosmadakis, S.; Papadimitriou, C. "Updates of Relational Views", Journal of the Association for Computing Machinery, vol. 31, núm. 4, 1984, pp. 742-760.
- [Dat86] Date, C.J. "Updating views", in *Relational Databases: Selected Writings*, Addison - Wesley, 1986, pp.367-395.
- [DC89] Decker, H.; Cavedon, L. "Generalizing Syntactic Properties Which Ensure Completeness of SLDNF Resolution", ECRC IR-KB-52, 1989.
- [Dec89] Decker, H. "The Range Form of databases or: How to avoid Floundering", Proc. 5th ÖGAI, Springer-Verlag, 1989.
- [Dec90a] Decker, H. "Drawing Updates from derivations" (versió estesa), ECRC IR-KB-65, 1990.
- [Dec90b] Decker, H. "Drawing Updates from derivations", Proc. of the 3rd Int. Conf. on Database Theory (ICDT), Paris, 1990, pp. 437-451.
- [Del91] Delhez, M. A. "Bases de donnees deductives: La methode des événements internes", Internal Report, Facultes Universitaires N. D. de la Paix, Namur, 1991.
- [EK89] Eshghi, K.; Kowalski, R. "Abduction Compared with Negation as Failure", Proc. 6th ICLP, 1989, pp. 234-254.
- [FC85] Furtado, A.L.; Casanova, M.A. "Updating Relational Views", in *Query Processing in Database Systems* (W.Kim et el. Eds.), Springer-Verlag, 1985, pp. 127-142.
- [FSS79] Furtado, A.L.; Sevcik, K.C, dos Santos, C.S. "Permitting Updates through views of Databases", Information Systems, 1979.
- [GL90a] Guessoum, A.; Lloyd, J.W. "Updating Knowledge Bases", New Generation Computing, vol. 8, núm. 1, 1990, pp. 71-89.
- [GL90b] Guessoum, A.; Lloyd, J.W. "Updating Knowledge Bases II", draft, Univ. Bristol Computer Science, 1990.
- [GM92] Grant, J.; Minker, J. " The Impact of Logic Programming on Databases", Communications of the ACM, vol. 35, núm. 3, 1992, pp. 66-81.
- [GV89] Gardarin, G.; Valduriez, P. "Relational Databases and Knowledge Bases", Addison-Wesley, 1989.

- Kak90]** Kakas, A. "Belief Revision in Deductive Databases", International Workshop on the Deductive Approach to Information Systems and Databases, Report LSI-90-30, S'Agaró, 1990, pp. 191-218.
- KM90a]** Kakas, A.; Mancarella, P. "Database Updates through Abduction", Proc. of the 16th Int. Conf. on Very Large Data Bases (VLDB), Brisbane, 1990.
- KM90b]** Kakas, A.; Mancarella, P. "An Abductive Procedure for Database View Updates", Imperial College Report, London 1990.
- Kel85]** Keller, A.M. "Algorithms for Translating View Updates to Database Updates for Views Involving Selection, Projections and Joins". Proc. 4th. ACM SIGACT-SIGMOD Symp. on Principle of Database Systems, 1985, pp. 154-163.
- Kel86a]** Keller, A.M. "The Role of Semantics in Translating View Updates", IEEE Computer, vol. 19, núm. 1, 1986, pp. 63-73.
- Kel86b]** Keller, A.M. "Choosing Translator at View Definition Time". Proc. 12th Int. Conf. on Very Large Data Bases (VLDB), Kyoto, 1986, pp. 467-474.
- Kel87]** Keller, A.M. "Comments on Bancilhon and Spyrtos: 'Update Semantics of Relational Views'", Technical Note, ACM Transactions on Database Systems, vol. 12, núm. 3, 1987, pp. 521-523.
- Kow78]** Kowalski, R. "Logic for Data Description". In *Logic and Data Bases* (Gallaire, H.; Minker, J. Eds), Plenum Press, New York, 1978, pp. 77-103.
- KU84]** Keller, A.; Ullman, J.D. "On complementary and independent mappings" Proc. ACM SIGMOD Int. Conf. on Management of Data, 1984, vol. 14, núm. 1, pp. 143-149.
- Kun89]** Kunen, K. "Signed Data Dependencies in Logic Programs", Journal of Logic Programming, 7(3), 1989, pp. 231-245.
- Lan90]** Langerak, R. "View Updates in Relational Databases with an Independent Schema Interface", ACM Transactions on Database Systems, vol. 15, núm. 1, 1990, pp. 40-66.
- Llo87]** Lloyd, J.W. "Foundations on Logic Programming", 2nd edition, Springer, 1987.
- LS91]** Larson, J; Sheth, A. "Updating Relational Views Using Knowledge at View Definition and View Update Time", Information Systems, vol. 16, núm. 2, 1991, pp. 145-168.
- LT84]** Lloyd, J.W.; Topor, R.W. "Making Prolog More Expressive". Journal of Logic Programming, 1984, núm. 3, pp. 225-240.

- [Mas84] Masunaga, Y. "A Relational Database View Update Translation Mechanism". Proc. 10th Int. Conf. on Very Large Data Bases (VLDB), Singapore, 1984, pp. 309-320.
- [Min88] Minker, J. "Perspectives in Deductive Databases", *Journal of Logic Programming*, vol. 5, 1988, pp. 33-60.
- [MW88] Manchanda, S.; Warren, D.S. "A logic-based language for database updates". In *Foundations of Deductive Databases and Logic Programming* (J. Minker Ed.), Morgan-Kaufmann Pub., 1988, pp. 363-394.
- [Oli89] Olivé, A. "On the Design and Implementation of Information Systems from Deductive Conceptual Models", Proc. 15th Int. Conf. on Very Large Data Bases (VLDB), Amsterdam, 1989, pp. 3-11.
- [Oli91] Olivé, A. "Integrity Checking in Deductive Databases", Proc. 17th Int. Conf. on Very Large Data Bases (VLDB), Barcelona, Catalonia, 1991, pp. 513-523.
- [Rei84] Reiter, R. "Towards a Logical Reconstruction of Relational Database Theory". In *On Conceptual Modelling* (Gallaire and Minker Eds.), Springer-Verlag, 1984, pp. 191 - 233.
- [Rob65] Robinson, J.A. "A machine-oriented logic based on the resolution principle", *Journal of the ACM*, vol. 12, 1965, pp. 23-41.
- [Sag83] Sagiv, Y. "A Characterization of Globally Consistent Databases and Their Correct Access Paths". *ACM Transactions on Database Systems*, vol. 8, núm. 2, 1983, pp. 266-286.
- [San92] Sancho, M. R. "The Odissea Approach to the Design of Information Systems from Deductive Conceptual Models", per aparèixer a Proc. IFIP Congress'92, Madrid, 1992.
- [SLW88] Sheth, A.P.; Larson, J.P.; Watkins, E. "TAILOR, a tool for Updating Views". Proc. Int. Conf. on Extending Data Base Technology (EDBT'88), Venice, 1988.
- [SM89] Subieta, K.; Missala, M. "View Updating Through Predefined Procedures", *Information Systems*, vol. 14, núm. 4, 1989, pp. 291-305.
- [Spy80] Spyratos, N. "Translation Structures of Relational Views", Proc. 6th Int. Conf. on Very Large Data Bases (VLDB), Montreal, 1980, pp. 411-416.
- [SS86] Sterling, L.; Shapiro, E. "The Art of Prolog", MIT Press Series in Logic Programming, Massachusetts, 1986.

- [Sto75] Stonebraker, M. "Implementation of Integrity Constraints and Views by Query Modification". Proc. ACM SIGMOD Int. Conf. On Management of Data, San José, 1975.
- [Ten90] Teniente, E. "The Internal Events Method for View Updating in Deductive Databases", International Workshop on the Deductive Approach to Information Systems and Databases, Report LSI-90-30, S'Agaró, 1990, pp. 173-190.
- [Ten91] Teniente, E. "Comprovació de Restriccions d'Integritat i Modificació de Vistes en Bases de Dades Deductives", Report LSI-91-3, Barcelona, 1991.
- [Ten92] Teniente, E. "Actualització de Vistes en Bases de Dades Relacionals: l'Estat de la Qüestió", Report LSI-92-13, Barcelona, 1992.
- [TFC83] Tucherman, L.; Furtado, A.L.; Casanova, M.A. "A Pragmatic Approach to Structured Database Design", Proc. 9th Int. Conf. on Very Large Data Bases (VLDB), Florence, 1983, pp. 219-231.
- [TO92] Teniente, E.; Olivé, A. "The Events Method for View Updating in Deductive Databases", Int. Conf. on Extending Database Technology (EDBT'92), Vienna, 1992, pp. 245-260.
- [Tom88] Tomic, A. "View Update Annotation in Definite Deductive Databases", Proc. Int. Conf. on Database Theory (ICDT '88), Springer-Verlag, 1988, pp. 338-352.
- [Ull88] Ullman, J.D. "Principles of Database and Knowledge-Base Systems", Computer Science Press, New York, 1988.
- [UO92] Urpí, T.; Olivé, A. "A Method for Change Computation in Deductive Databases", per aparèixer als Proc. 16th Int. Conf. on Very Large Data Bases (VLDB), Vancouver, 1992.
- [vEK76] van Emden, M. H; Kowalski, R. "The semantics of predicate logic as a programming language", Journal of the ACM, núm. 23, 1976, pp. 733-742.

Annex: codi de la implementació

```
translate(Goal) :- solve(Goal,[],Lsoluc,[],Lcond),
    write('Trad = '), write(Lsoluc), nl,
    write('Cond = '), write(Lcond), nl,
    check_conds (Lsoluc,Lcond,L_int_ev),
    write('T = '), write(L_int_ev), nl, nl,
    assert(ok),
    findall(X,clause(solucio(X),true),Llista),
    study_subsumption(L_int_ev,Llista),
    fail.

translate(Goal) :- retractall(dem_param(ins(A))), fail.

translate(Goal) :- ok, retractall(ok), nl, nl,
    findall(X,solucio(X),Llista),
    escriu_llista(Llista),
    retractall(solucio(X)).

solve(true,X,X,Y,Y) :- !.

solve((A,B),Ls1,Ls,Lc1,Lc) :-
    !, solve(A,Ls1,Ls2,Lc1,Lc2), solve(B,Ls2,Ls,Lc2,Lc).

solve(not(ins(A)),X,X,Y,[[ins(A)]|Y]) :-
    not(A),
    not(member(not(ins(A)),X)), !.

solve(not(ins(A)),X,X,Y,Y) :-
    !.

solve(not(del(A)),X,X,Y,[[del(A)]|Y]) :-
    A, not(member(not(del(A)),X)), !.

solve(not(del(A)),X,X,Y,Y) :-
    !.

solve(not(A),X,Ls,Y,Lc) :-
    do_fail([A],X,Ls,Y,Lc).
```

```

solve(ins(A),X,X,Y,Y) :-
    ins(A), !.

solve(ins(A),X,[ins(A)|X],Y,Y) :-
    A=..[T1|L], hi_ha_termes_no_instanciats(L), /* A és un predicat que té
    dem_param(ins(A)), !,                algun dels seus membres
    asserta(ins(A)),                    sense instanciar, caldrà
    on_backtracking(retract(ins(A))).    doncs demanar-li els valors
                                        a l'usuari */

solve(ins(A),X,[ins(Res)|X],Y,Y) :-
    A=..[T1|L], hi_ha_termes_no_instanciats(L), !,
    nl, write('Assigna valors a les variables de '),
    write(A), write(' per inserci  '),
    llegir(L), nl,
    Res =.. [T1|L], not(Res),
    asserta(ins(Res)), asserta(dem_param(ins(Res))),
    on_backtracking(retract(ins(Res))).

solve(ins(A),X,[ins(A)|X],Y,Y) :-
    !, not(A), asserta(ins(A)), /* Ja tenim tots els valors
    on_backtracking(retract(ins(A))). a inserir, no cal que li
                                        demanem res a l'usuari */

solve(del(A),X,X,Y,Y) :-
    del(A), !.

solve(del(A),X,[del(A)|X],Y,Y) :-
    !, A, asserta(del(A)), /* El fet de preguntar per A ens
    on_backtracking(retract(del(A))). instància les seves variables */

solve(A,X,Ls,Y,Lc) :-
    clause(A,B),
    solve(B,X,Ls,Y,Lc).

/* La condició de permissibilitat garanteix que tots els literals negatius seran completament
instanciats abans de ser avaluats per la regla de negació per fracàs finit */

do_fail([not(ins(A))],X,X,Y,Y):-
    ins(A), !.

do_fail([not(ins(A))],X,[ins(A)|X],Y,Y):-
    not(A), !,
    asserta(ins(A)),
    on_backtracking(retract(ins(A))).

do_fail([not(ins(A))|B],X,X,Y,Y) :-
    clause(ins(A),true), !.

do_fail([not(ins(A))|B],X,[ins(A)|X],Y,Y) :-
    not(A), asserta(ins(A)),
    on_backtracking(retract(ins(A))).

```

```

do_fail([not(ins(A))|B],X,Ls,Y,Lc) :-
    !, do_fail(B,X,Ls,Y,Lc).

do_fail([not(del(A))],X,X,Y,Y):-
    A, del(A), !.

do_fail([not(del(A))],X,[del(A)|X],Y,Y):-
    A, !,
    asserta(del(A)),
    on_backtracking(retract(del(A))).

do_fail([not(del(A))|B],X,X,Y,Y):-
    A, del(A), !.

do_fail([not(del(A))|B],X,[del(A)|X],Y,Y):-
    A, asserta(del(A)),
    on_backtracking(retract(del(A))).

do_fail([not(del(A))|B],X,Ls,Y,Lc) :-
    !, do_fail(B,X,Ls,Y,Lc).

do_fail([ins(A)],X,X,Y,[ins(A)|Y]) :-
    A=..[T1|Z], hi_ha_termes_no_instanciats(Z), !.

do_fail([ins(A)],X,X,Y,[ins(A)|Y]) :-
    not(A), not(ins(A)),
    not(member(not(ins(A)),X)), !.

do_fail([ins(A)],X,X,Y,Y) :- !, not(ins(A)).

/* Quan en un do_fail trobem una goal en la que el literal seleccionat és un esdeveniment bàsic
que té termes no instanciats, cal afegir totes les solucions possibles a Ls i afegir la goal a Lcond
*/

do_fail([ins(A)|B],X,Ls,Y,[ins(A)|B]|Y) :-
    A=..[T1|Z], hi_ha_termes_no_instanciats(Z),
    findall(B,clause(ins(A),true),Llista), !,
    fer_fallar_llista(Llista,X,Ls,_,_).

do_fail([ins(A)|B],X,X,Y,[ins(A)|B]|Y) :-
    A=..[T1|Z], hi_ha_termes_no_instanciats(Z), !.

do_fail([ins(A)|B],X,Ls,Y,Lc) :-
    not(A),
    do_fail(B,X,Ls,Y,Lc).

do_fail([ins(A)|B],X,X,Y,[ins(A)|Y]) :-
    not(A), not(ins(A)),
    not(member(not(ins(A)),X)), !.

do_fail([ins(A)|B],X,X,Y,Y) :- !, not(ins(A)).

```

```

do_fail([del(A)],X,X,Y,[[del(A)]|Y]) :-
    A, not(del(A)),
    not(member(not(del(A)),X)), !.

do_fail([del(A)],X,X,Y,Y) :- !, not(del(A)).

do_fail([del(A)|B],X,Ls,Y,[[del(A)|B]|Y]) :-
    A=..[T1|Z], hi_ha_termes_no_instanciats(Z),
    findall(B,clause(del(A),true),Llista), !,
    fer_fallar_llista(Llista,X,Ls,_,_)).

do_fail([del(A)|B],X,Ls,Y,[[del(A)|B]|Y]) :-
    A=..[T1|Z], hi_ha_termes_no_instanciats(Z), !.

do_fail([del(A)|B],X,Ls,Y,Lc) :-
    A,
    do_fail(B,X,Ls,Y,Lc).

do_fail([del(A)|B],X,X,Y,[[del(A)]|Y]) :-
    A, not(del(A)),
    not(member(not(del(A)),X)), !.
do_fail([del(A)|B],X,X,Y,Y) :- !, not(del(A)).

do_fail([not(A)],X,Ls,Y,Lc) :-
    !, solve(A,X,Ls,Y,Lc).

do_fail([not(A)|B],X,X,Y,Y) :-                               /* A és bàsic i és a la BD */
    clause(A,true), !.

do_fail([not(A)|B],X,Ls,Y,Lc) :-                             /* A és derivat */
    findall(C,clause(A,C),List),
    List = [],
    solve(A,X,Ls,Y,Lc).

do_fail([not(A)|B],X1,Ls,Y1,Lc) :-                            /* A és derivat */
    clause(A,C),                                             /* Si no hi cap manera de fer
    !,                                                         fracassar A, no cal que fem
    do_fail(B,X1,Ls,Y1,Lc).                                  fracassar B */

do_fail([not(A)|B],X,Ls,Y,Lc) :-                             /* A és bàsic i NO és a la BD */
    !, do_fail(B,X,Ls,Y,Lc).

do_fail([A],X,X,Y,Y) :-                                       /* A és bàsic i és a la BD */
    clause(A,true),!, fail.

/* A és derivat. Cal buscar totes les clàusules C, del tipus A:-C,
i fer fallar C per cadascuna d'elles */

do_fail([A],X1,Ls,Y1,Lc):-
    findall(L,(clause(A,C),convert(C,L)),Llista),
    !, fer_fallar_llista(Llista,X1,Ls,Y1,Lc).

```

```
do_fail([A|B],X1,Ls,Y1,Lc) :-                               /* A és fet bàsic i és a la BD */
    findall(B,clause(A,true),Llista),
    Llista \= [],
    !, fer_fallar_llista(Llista,X1,Ls,Y1,Lc).
```

```
do_fail([A|B],X1,Ls,Y1,Lc) :-                               /* A és derivat */
    findall(L,(clause(A,C),
    convert(C,Lint),append(Lint,B,L)),Llista),
    !, fer_fallar_llista(Llista,X1,Ls,Y1,Lc).
```

```
do_fail([A|B],X,X,Y,Y) :- !.                               /* A és bàsic i NO és a la BD */
```

/* Necessitem tres predicats auxiliars per a garantir que el conjunt de traducció C no viola cap condició de C */

check_conds (Lsolut, Lcond, L_int_ev) <= Donades dues llistes, una amb la solució parcial i una altra de condicions, retorna el conjunt de traducció final T que satisfà les condicions.

make_do_not_violate (Elt, Lcond, New_int_ev) <= Donat un element (esdeveniment bàsic) i una llista de condicions, retorna els nous esdeveniments necessaris per a assegurar que Elt no viola cap condició.

treat_cond (Elt, Cond, New_int_ev) <= Donat un element (esdeveniment bàsic) i una condició, retorna els nous esdeveniments necessaris per a assegurar que Elt no viola cap condició. */

```
check_conds ([],L,[]).
```

```
check_conds ([Elt|Ls],Lcond,[Elt|Total_solut]) :-
    make_do_not_violate (Elt,Lcond,New_int_ev),
    check_conds (Ls,Lcond,X),
    check_conds (New_int_ev,Lcond,Y),
    append(X,Y,Total_solut).
```

```
make_do_not_violate (Elt,[],[]).
```

```
make_do_not_violate (Elt,[Cond|Lc],New_int_ev) :-
    treat_cond (Elt,Cond,X),
    make_do_not_violate (Elt,Lc,Y),
    append(X,Y,New_int_ev).
```

```
treat_cond (ins(A),[ins(A)|B],L) :- !, do_fail(B,[],L,[],_).
treat_cond (ins(A),[ins(A)],L) :- !, fail.
```

```
treat_cond (del(A),[del(A)|B],L) :- !, do_fail(B,[],L,[],_).
treat_cond (del(A),[del(A)],L) :- !, fail.
```

```
treat_cond (_,_,[],_).
```

/ study_subsumption (Soluc,List) estudia si la solució Soluc subsumeix algun element de la llista List (List conté les solucions que han estat obtingudes previàment). Recordi's que el nostre mètode obté només solucions mínimes */*

```
study_subsumption(Soluc,[]) :- assert(solucio(Soluc)), !.
```

```
study_subsumption(Soluc,[H|T]) :-
    countlist(Soluc,N1),
    countlist(H,N2),
    N2 =< N1,
    subsumes(H,Soluc), !.
```

```
study_subsumption(Soluc,[H|T]) :-
    countlist(Soluc,N1),
    countlist(H,N2),
    N1 < N2,
    subsumes(Soluc,H), !,
    retract(solucio(H)),
    study_subsumption(Soluc,T).
```

```
study_subsumption(Soluc,[H|T]) :-
    study_subsumption(Soluc,T).
```

/ subsumes(X,Y)=cert si la llista X subsumeix la llista Y */*

```
subsumes([],_) :- !.
subsumes([H|T],Y) :-
    member(H,Y),
    subsumes(T,Y).
```

/ A partir d'aquí es troben els predicats intermedis necessaris pel mètode */*

```
fer_fallar_llista([],Ls,Ls,Lc,Lc).
fer_fallar_llista([H|T],X,Ls,Y,Lc) :-
    !, do_fail(H,X,Ls_int,Y,Lc_int),
    fer_fallar_llista(T,Ls_int,Ls,Lc_int,Lc).
```

```
convert((A,B),Llista) :-
    convert(B,L1),
    append([A],L1,Llista), !.
```

```
convert(A,[A]) :- !.
```

```
append([],L,L) :- !.
append([X|L1],L2,[X|L3]) :- append(L1,L2,L3).
```

```
findall(X,G,_) :- asserta(found(mark)), call(G), asserta(found(X)), fail.
```

```
findall(_,L) :- collect_found([],M), !, L=M.
```

```
collect_found(S,L) :- getnext(X), !, collect_found([X|S],L).
```

```
collect_found(L,L).
```

```
getnext(X) :- retract(found(X)), !, X \== mark.
```

/ El predicat "llegir" s'usa per llegir (instanciar) els termes d'un àtom que no estan instanciats quan s'ha d'inserir o esborrar aquest àtom. O sigui cal posar l'àtom instanciat a la llista de solucions T */*

```
llegir([Term]) :- var(Term), read(Term), !.  
llegir([Term]).  
llegir([T1|T2]) :- llegir([T1]), llegir(T2), !.
```

/ El predicat hi_ha_termes_no_instanciats(L) comprova si algun element de la llista L és una variable */*

```
hi_ha_termes_no_instanciats([Term|_]) :- var(Term).  
hi_ha_termes_no_instanciats([_L1]) :- hi_ha_termes_no_instanciats(L1).
```

/ El predicat retract_int_ev retracta els esdeveniments bàsics que pertanyen al conjunt de traducció. Això permet que el programa pugui obtenir una nova traducció T */*

```
retract_int_ev :- retract(ins(A)), fail.  
retract_int_ev :- retract(del(A)), fail.  
retract_int_ev.
```

```
llista_int_ev :- ins(A), write(ins(A)), nl, fail.  
llista_int_ev :- del(A), write(del(A)), nl, fail.  
llista_int_ev.
```

/ el predicat on_backtracking(X) s'usa per a executar X durant el backtrack */*

```
on_backtracking(X).  
on_backtracking(X) :-  
    X, !, fail.
```

/ member(Element,List) :- Element és un element de la llista List */*

```
member(X,[X|_]) :- !.  
member(X,[_|T]) :- member(X,T).
```

/ countlist(List,N) :- N és el nombre d'elements de la llista List */*

```
countlist([],0).  
countlist([X|T],M) :- countlist(T,N), M is N+1.
```

```
escriu_llista([]).  
escriu_llista([X|T]) :-  
    write(X), nl,  
    escriu_llista(T).
```


Agraïments

Vull donar les gràcies a totes les persones que, directament o indirecta, m'han ajudat durant la realització d'aquesta tesi. Els seus comentaris, consells i encoratjament han estat de vital importància per a l'execució d'aquest treball.

Vull agrair especialment al Dr. Antoni Olivé i Ramon el seu excel·lent mestratge en la direcció d'aquesta tesi. El seu rigor intel·lectual, la seva exigència, i també la seva disponibilitat i ajut, m'han permès comprendre perfectament el significat de la paraula investigació. Realment, se'm fa molt difícil expressar en paraules la meva gratitud envers la seva persona.

També vull donar gràcies a la Dolors Costal, a en Hendrik Decker, a en Robert Demolombe, a l'Andonakis Kakas, a l'Enric Mayol, a en Joan Antoni Pastor, a la Carme Quer, a la Maria Ribera Sancho, a en Jaume Sistac i a en Toni Urfí pels seus comentaris i suggeriments a les versions preliminars d'aquesta tesi. La seva inestimable i altruista col·laboració no l'oblidaré mai.

Finalment, vull agrair a la Marta Mariñosa la seva paciència a l'hora de corregir les meves faltes d'ortografia. Els seus savis consells m'han ajudat a conèixer millor la riquesa de la nostra llengua.

Aquest treball ha rebut un ajut parcial del programa CICYT PRONTIC, projecte TIC 680.