

UNIVERSITAT POLITÈCNICA DE CATALUNYA
DEPARTAMENT DE LENGUATGES I SISTEMES INFORMÀTICS

**UN MÈTODE PER A LA COMPROVACIÓ DE
RESTRICCIONS D'INTEGRITAT EN BASES DE DADES
DEDUCTIVES BITEMPORALS**

TESI DOCTORAL

DIRIGIDA PER LA DRA. NÚRIA CASTELL ARIÑO

BARCELONA

2005

Memòria presentada per Carme Martín Escofet per tal d'aconseguir el grau de
Doctora en Informàtica per la Universitat Politècnica de Catalunya amb la
menció de Doctora Europea

*A l'Anna i la Rosa,
les trobo molt a faltar.*

*A en Jordi,
per tot, perquè
sense ell no hagués
estat possible aquesta tesi.*

Agraïments

Hi ha moltíssima gent a la qual he d'agrair que aquesta tesi hagi estat possible, però sens dubte he de començar per la persona que em va proposar aquest interessantíssim tema: Antoni Olivé. A ell especialment, li dec haver sabut despertar en mi l'interès per treballar en aquest tema, les primeres idees i les primeres empentes per continuar endavant malgrat les dificultats que anàvem trobant. També en els inicis d'aquest treball he d'agrair als components del grup ODISSEA: Dolors Costal, Enric Mayol, Antoni Olivé, Joan Antoni Pastor, Carme Quer, Maria Ribera Sancho, Jaume Sistac, Ernest Teniente i Toni Urpí, tot el que vaig aprendre al seu costat i que més tard m'ha servit, d'una manera o altra, per aplicar-ho en la meva investigació.

He tingut la immensa sort de tenir dos grans directors per a aquesta tesi. Jaume Sistac, que em va acompanyar en els inicis, va saber enfocar sàviament el tema i em va assessorar i ajudar en les nostres primeres publicacions, que han donat finalment el seu fruit. Aquest fruit que hem obtingut, ha estat possible gràcies a l'ajuda inestimable de Núria Castell. Amb ella hem aconseguit donar forma a aquest treball, superar les darreres barreres pendents i obtenir els resultats que aquí es presenten.

Voldria fer una menció especial a tots els meus companys i excompanys de despatx, que han viscut tot el procés d'elaboració d'aquest treball, aconsellant-me i donant-me el seu suport en tot moment. A Àngels Hernández, Lluís Padró, Jordi Daudé, Mario Martín, Javier Larrosa, Juan Luis Esteban, Jordi Delgado, Christian Estay i sobretot al meu gran company i amic Pablo Costa, moltes gràcies per la vostra paciència.

A més dels meus companys i excompanys de despatx, hi ha molta gent del meu departament que m'ha ajudat en aquesta feina, ni que sigui escoltant-me quan ho he necessitat. A Victoria Arranz, Fèlix Saltor, Miquel Barceló, Anna Roselló, Lluís Vila, Jordi Conesa, Xavier de Palol, Villie Morocho, M. Elena Rodríguez, Carles Farré, Cristina Gómez, Juan Ramón López, Francesc Sistach, Rafael Camps, Francesc Costa, Jordi Viñallonga, Xavier Franch, Jordi Marco, Xavier Burgués, Ton Sales, M. Teresa Abad, Enrique Romero, María J. Casañ, Marc Alier, Pere Botella, Luis Felipe Fernández, M. Amelia Pacheco i, moltes gràcies.

M'agradaria també donar les gràcies als meus companys de doctorat amb els quals tant he après sobre investigació i amb els quals vaig escriure el meu primer article de recerca: Marta Oliva, Feliciano Sesé i Olga Slavkova.

El tema del meu treball m'ha permès contactar i conèixer molta gent d'arreu del món, que tenen en comú la mateixa passió per aquesta àrea d'investigació. Mario Nacimiento, Andreas Steiner, Robert Kowalski, Jorge Lobo, Jack Minker, Michael Gelfond, Jan Chomicki, David Toman, Carlo Zaniolo, Barbara Pernici, Babis Theodoulidis, Christophe Claramunt, Jan Van den Bussche, Bertram Ludascher, George Gottlob, Arno Siebes, Hadassa Jakobovits, Estrella Pulido, Ale Provetti, Nieves Brisaboa, Amilcar Sernadas i Guido Moerkotte, m'han ajudat donant-me inapreciables consells i suggeriments sobre com desenvolupar la meva feina. I m'agradaria destacar l'inestimable suport, ajut i col·laboració que sempre he rebut de Michael Gertz i Hendrik Decker.

No tinc paraules per agrair l'ajuda desinteressada que he rebut de Michael H. Böhlen, amb qui he estat treballant durant més de tres mesos a la Facultat d'Informàtica de la Universitat Lliure de Bozen-Bolzano. L'atenció que he rebut durant la meva estada de recerca, la qualitat de la feina desenvolupada i el meravellós ambient de treball de què he gaudit són del tot impagables. També voldria mencionar els meus companys del Centre de Bases de Dades i Sistemes d'Informació (DIS): Nikolaus Augsten, Johann Gamper, Peter Sune Jorgensen i Arturas Mazeika per la seva companyonia i ajuda en tot moment, cosa que ha fet que la meva estada hagi estat inoblidable. Aquesta estada ha estat possible gràcies a la concessió d'un ajut a la mobilitat externa del PDI de la UPC.

Amb Alberto Abelló he tingut el gran plaer d'endinsar-me en el món dels magatzems de dades. Compartir els grans coneixements que té en aquesta àrea per relacionar-los amb les bases de dades temporals ha estat un gran privilegi.

A les estudiants Olga Montero i Cristina Surroca voldria agrair-los l'entusiasme i la motivació amb què em van ajudar a implementar les estructures d'emmagatzemament bitemporal per a magatzems de dades. I als estudiants Carlos López i Cristina Rubio els vull agrair la seva dedicació i col·laboració en la implementació dels termes dependents temporalment.

La qualitat del català d'aquesta tesi és el resultat de l'acurada revisió que han fet des del Servei de Llengües i Terminologia de la UPC mitjançant un ajut a la correcció de tesi doctoral redactada en català de la UPC.

He deixat per al final la gent que més ha patit aquest treball, en el sentit que no han pogut estar amb mi tant de temps com haguessin desitjat. Als meus pares: Jesús Martín i Carmen Escofet i a la meua parella: Jordi Grau, moltes, moltes gràcies per haver-me entès en tot moment. I també gràcies a tots els nostres amics, que quan lleixin això ja podran celebrar que he acabat la tesi.

ÍNDIX

1. Introducció	1
1.1. Objectius de la tesi	5
1.2. Organització del document	6
2. Bases de dades temporals	7
2.1. Tipus de temps	7
2.2. Conceptes bàsics de bases de dades temporals	13
2.3. Models de dades temporals	18
2.3.1. La base de dades temporal abstracta	18
2.3.2. El model bitemporal conceptual	20
2.3.3. Model de representació	22
3. Bases de dades deductives	25
3.1. Regles de deducció	26
3.2. Restriccions d'integritat	27
3.3. El mètode dels esdeveniments interns	28
4. Bases de dades deductives bitemporals	33
4.1. Insercions	35
4.2. Esborrats	40
4.3. Modificacions	44
4.4. Fusió	46
4.5. Transaccions coherents	51

5. Restriccions d'integritat temporals	55
5.1. Restriccions d'integritat temporals	56
5.2. Taxonomia de Michael Böhlen	57
5.3. Taxonomia d' Anne Doucet et al.	59
5.4. Taxonomia de restriccions d'integritat temporals en bases de dades deductives bitemporals	61
5.4.1. Exemple	63
5.4.2. Referències a temps de transacció passats	66
5.4.3. Anàlisi de les condicions de les restriccions d'integritat temporals	67
5.4.4. Primer grup: restriccions d'integritat actuals	69
5.4.5. Segon grup: restriccions d'integritat en temps de transacció	71
5.4.6. Tercer grup: restriccions d'integritat en temps vàlid	73
5.4.7. Quart grup: restriccions d'integritat bitemporals	75
5.4.8. Avantatges de la nova classificació	76
6. Descripció del mètode de comprovació de restriccions d'integritat en bases de dades deductives bitemporals	79
6.1. Esdeveniments temporals	79
6.2. Regles de transició temporals	84
6.3. Regles d'esdeveniments temporals	87
6.4. La base de dades augmentada	89
6.5. Comprovació de restriccions d'integritat en bases de dades deductives bitemporals	89
6.6. Restriccions d'integritat actuals	90
6.7. Restriccions d'integritat en temps de transacció	94

6.8. Restriccions d'integritat en temps vàlid	99
6.9. Restriccions d'integritat bitemporals	104
6.10. Simplificacions de les regles de transició temporals	109
6.11. Regles de transició després d'aplicar el mètode de descomposició de restriccions d'integritat	110
6.12. Restriccions d'integritat amb predicats derivats	114
6.13. Comparació amb altres mètodes	117
7. Termes dependents temporalment	119
7.1. Comparació amb el treball previ relacionat	122
7.2. La base de dades completada	126
7.3. Operacions sobre la base de dades completada	135
7.4. Consultes dependents temporalment	140
7.5. Restriccions d'integritat dependents temporalment	141
8. Magatzems de dades bitemporals	145
8.1. Bases de dades bitemporals i magatzems de dades	146
8.2. Una estructura de dades bitemporal com a magatzem de dades	149
9. Conclusions i recerca futura	153
9.1. Conclusions	153
9.2. Recerca futura	157
Referències	159
Annex	173

1. Introducció

Les bases de dades temporals han estat, en els darrers anys, el centre d'interès de molts investigadors, tant de l'àmbit universitari com de l'industrial, a causa de la seva important aplicació en molts sectors. Són especialment destacables els sectors mèdic i bancari, on el tractament temporal ha estat aplicat específicament per resoldre problemes concrets d'aquests entorns.

Seguint la cronologia presentada a [Sno90] emmarcarem, anecdòticament, les primeres bases de dades temporals fa milers d'anys, quan ja es va sentir la necessitat d'emmagatzemar informació. I aquesta informació es donava en un temps i evolucionava amb el temps. El primer intent de treballar amb bases de dades temporals podríem trobar-lo en el període paleolític, si considerem les pintures rupestres com una primera aproximació del que més endavant s'anomenarien bases de dades en temps vàlid.¹ Les pintures rupestres, que ens donen informació del que va passar fa 27.000 anys, fins i tot van permetre modificacions d'informació, com es pot notar a la figura 1.1 en l'encavalcament existent entre uns dibuixos i uns altres. El més semblant a les primeres bases de dades en temps de transacció² ho trobem, ja més tard, en les inscripcions de la civilització sumèria, o els jeroglífics de la civilització egípcia, uns 3.000 anys aC, on la informació pot ser modificada en el moment en què es grava, però no abans

¹ S'anomenen bases de dades en temps vàlid aquelles que ens ofereixen la gestió del temps en el qual realment es coneix la informació.

² S'anomenen bases de dades en temps de transacció aquelles que ens ofereixen la gestió del temps en el qual s'emmagatzema la informació.

ni després. Podem veure un exemple de l'emmagatzemament sumeri a la figura 1.2. Evidentment, tots aquests primers intents d'emmagatzemar informació tenen una dimensió temporal implícita.



Figura 1.1. Figures policromades del sostre de la sala d'entrada de les coves d'Altamira

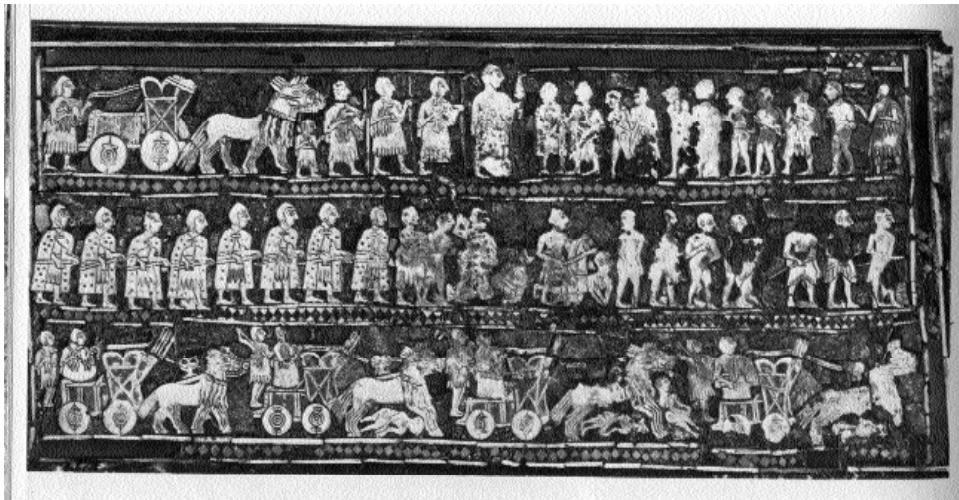


Figura 1.2. L'estendard d'Ur. Primera dinastia sumèria. Museu Britànic

Deixant de banda els inicis anecdòtics de les bases de dades temporals i centrant-nos ja en els segles XX i XXI, s'han escrit diverses bibliografies amb articles ben rellevants al voltant de la incorporació de temps en bases de dades. Concretament, el 1982 trobem [BAD+82], el 1986 [McK86] i [Sno86], el 1988 [SS88], el 1990 [Sno90], el 1991 [Soo91], el 1993 [Kli93], el 1996 [TK96] i finalment el 1998 [WJW98]. Endinsem-nos, doncs, en el nostre temps, amb l'objectiu d'emmarcar l'aportació d'aquesta tesi.

Al principi dels anys setanta, es va començar a veure la necessitat d'incorporar el temps en l'àrea de sistemes d'informació, concretament, en estudis realitzats sobre models conceptuals de dades. Una altra àrea pionera a destacar en la incorporació de temps va ser la intel·ligència artificial, que va començar a aplicar la lògica temporal en els seus treballs.

El primer treball, pròpiament de l'àrea de bases de dades, apareix el 1975 [WFW75]. Aquest article és el primer que es va presentar en un congrés d'informàtica. I, a més, els seus autors implementen el primer sistema de gestió de bases de dades (SGBD) en temps vàlid a partir dels requeriments d'un sistema d'informació mèdic.

El 1977 localitzem el primer article [Sch77] que utilitza temps de transacció. No gaire més tard, al final dels anys setanta, es realitza la primera implementació d'un SGBD en temps de transacció [Kim78]. I és aquest també el moment en què apareix el primer llenguatge algebraic de consultes [JMS79] que ofereix combinacions, seleccions i agregats temporals.

El 1983 es defineix la primera semàntica formal per al model de dades en temps vàlid [CW83]. Aquest és el primer article que introdueix el concepte de bases de dades orientades a temps en publicacions importants de bases de dades.

Entre el 1984 i el 1985 s'implementa el primer prototipus d'SGBD bitemporal amb suport de temps vàlid i temps de transacció [DLW84], [LDE+84] i [LDE+85]. També el 1985, es demostra que el temps vàlid i el temps de transacció són ortogonals, la qual cosa permet poder cercar cadascun independentment [AS86].

La TAIS (Temporal Aspects in Information Systems), feta el 1987, és la primera conferència dedicada a les bases de dades temporals. Encara que molt

orientada a sistemes d'informació, s'hi poden trobar articles que consideren molts altres aspectes del temps.

El 1988 es fan les primeres investigacions en bases de dades deductives temporals [CI88], ja que la gran quantitat d'informació, en molts casos infinita, que requereixen emmagatzemar les bases de dades temporals, es veu clarament fonamentada per les regles de deducció de les bases de dades deductives.

Al principi dels anys noranta comença un creixement espectacular del nombre de treballs sobre bases de dades temporals. Els principals temes en què se centra la recerca són el disseny de bases de dades temporals, els llenguatges de consultes temporals, aspectes d'implementació i bases de dades deductives temporals. A més, les bases de dades temporals han esdevingut un tema més d'interès en els principals congressos de bases de dades.

Són especialment destacables les publicacions següents: el primer glossari de conceptes de bases de dades temporals [JCG+92], en el qual, entre molts altres termes importants, es defineixen formalment, per primer cop, els temps vàlid i de transacció i la seva correspondència amb altres noms rebuts fins aleshores. També, cal destacar el primer llibre dedicat exclusivament a bases de dades temporals [TCG+93] i el primer llibre sobre el llenguatge temporal TSQL2 [Sno95]. Quant a congressos, el 1993 es fa a Arlington la primera reunió de treball sobre infraestructura de bases de dades temporals [Sno93] i el 1995 a Zuric una reunió de treball internacional sobre bases de dades temporals [CT95a] i un tutorial de bases de dades temporals [JS95]. Tots els resultats obtinguts en aquests congressos queden clarament refermats el 1998 en el seminari de Dagstuhl sobre bases de dades temporals [EJS98]. Els darrers llibres apareguts fins ara sobre bases de dades temporals són [Sno00], on ja es troba la nova i clara tendència dins l'àrea d'anar cap a l'aplicació pràctica dels conceptes teòrics de bases de dades temporals, que també podem veure reflectida en el llibre [DDL03], i [BJW00], on es tracta un tema molt important en bases de dades temporals com és la gestió de la granularitat del temps.

També durant els anys noranta han anat apareixent diferents URL destinades a assessorar sobre l'evolució de la recerca en bases de dades temporals. La majoria d'aquestes URL estan associades a algun grup d'investigació universitari. La més important és, potser, el Time Center de la universitat d'Aalborg (<http://www.cs.auc.dk/TimeCenter/>).

Al final dels anys noranta es va desenvolupar el projecte CHOROCHRONOS [KSF+03]. El principal objectiu d'aquest projecte era permetre que diferents investigadors europeus poguessin treballar en bases de dades temporals i espacials compartint informació per integrar amb facilitat resultats i metodologies.

Tenim, doncs, que durant els anys setanta s'implementen els primers SGBD en temps vàlid i en temps de transacció. Durant els anys vuitanta s'implementa el primer prototipus d'SGBD bitemporal i comencen els primers treballs en bases de dades deductives temporals. En els anys noranta es treballa en estàndards de disseny, conceptes i llenguatges, i continuen els treballs cap a una clara cooperació entre bases de dades deductives i bases de dades temporals.

1.1. Objectius de la tesi

Com hem comentat abans, durant els anys noranta la cooperació entre bases de dades deductives i bases de dades temporals experimenta un considerable interès, evidenciat per un increment en el nombre de publicacions en aquest àmbit. És el cas d'aquesta tesi, que treballa amb bases de dades deductives bitemporals, *bd-dbts* a partir d'ara, i presenta el primer mètode per a la comprovació de restriccions d'integritat en aquest tipus de bases de dades. Tenint en compte la tendència cap a l'obtenció d'estàndards, no és el nostre objectiu lligar-nos a cap llenguatge concret, sinó que en capítols posteriors utilitzarem un llenguatge de lògica de predicats de primer ordre per manipular *bd-dbts*.

Per assolir l'objectiu d'aquesta tesi –un mètode per a la comprovació de la integritat en *bd-dbts*– caldrà conèixer en profunditat les àrees de bases de dades temporals i bases de dades deductives. També caldrà estudiar el comportament de les transaccions en *bd-dbts* i totes les possibles restriccions d'integritat temporals que cal considerar en *bd-dbts*.

L'objectiu principal d'aquesta tesi, que a més li dona títol, ha estat descrit en els paràgrafs anteriors. Malgrat això, la nostra recerca en els darrers anys ha anat obtenint nous resultats, sempre sota la temàtica comuna de les bases de dades temporals. Com a mostra, hem considerat oportú incorporar dos capítols on es dibuixen les aportacions obtingudes de la nostra recerca més recent, com és l'estudi en la dimensió de temps vàlid de termes de predicats d'una base de dades

dependents del temps i la combinació de bases de dades temporals i magatzems de dades. La idea de considerar restriccions d'integritat més potents ens va portar a l'estudi dels termes dependents temporalment. Les dues dimensions de temps en què vam desenvolupar el nostre mètode de comprovació de restriccions d'integritat ens va donar la clau de la possible combinació de bases de dades temporals amb magatzems de dades. Amb aquests dos nous temes, tot just iniciats i que per descomptat formaran part també de la nostra recerca futura, donem per finalitzat el material que s'inclou en la tesi.

1.2. Organització del document

El capítol següent introdueix conceptes sobre bases de dades temporals, així com el capítol 3 ho fa de bases de dades deductives. El capítol 4 presenta l'actualització de bd-dbts i la seva correspondència amb el model de representació final. El capítol 5 estudia els diferents tipus de restriccions d'integritat temporals que podem trobar. El capítol 6 defineix el nostre mètode de comprovació de restriccions d'integritat. Els capítols 7 i 8 presenten els darrers temes en què ens hem endinsat. En el capítol 7 s'identifica un nou tipus de terme en els predicats d'una base de dades dependent del temps. Aquest nou tipus de terme, anomenat *terme dependent temporalment*, és definit formalment i és estudiat en profunditat per analitzar-ne les característiques i mostrar la seva aplicació a consultes i restriccions d'integritat. Aquesta tesi s'emmarca en un entorn bitemporal, excepte el capítol 7, que, com que es tracta d'un treball amb continuïtat futura, només es planteja per a la dimensió temporal de temps vàlid. En el capítol 8 les bases de dades temporals es combinen amb una altra àrea emergent, com són els magatzems de dades. En aquest capítol s'expliquen els punts que hem trobat que tenen en comú aquestes dues àrees de recerca i es proposa una estructura de dades bitemporal com a magatzem de dades corporatiu. I abans d'arribar a la bibliografia emprada, finalitzem amb el capítol 9, on oferim les principals aportacions d'aquesta tesi i les futures direccions de continuïtat de la nostra recerca.

2. Bases de dades temporals

Moltes aplicacions requereixen tractament temporal i, sense la incorporació de la gestió del temps en els sistemes de bases de dades, l'única solució és fer un tractament *ad hoc* de cada cas amb la ineficiència que això comporta. En aquest capítol introduïrem alguns conceptes bàsics de bases de dades temporals. A més, veurem que hi ha moltes decisions a prendre a l'hora de definir el temps amb què volem treballar. Analitzarem totes les possibilitats existents per establir una bona base per al nostre treball, ja que volem garantir que dins del ventall de solucions que han aparegut fins ara, escollim les més utilitzades.

2.1. Tipus de temps

Hem de distingir entre tres tipus de temps, definits per primer cop a [SA86], i estandarditzats a [JDB+98]:

Temps definit per l'usuari

És un atribut definit sobre un domini temporal sense cap tipus de suport d'un SGBD temporal. La *data de naixement* o la *data de contractació* són exemples de temps definits per l'usuari.

Temps vàlid

És el temps en què és certa la informació de la realitat que s'està modelant.

Temps de transacció

És el temps en què la informació s'emmagatzema a la base de dades.

Les relacions sense cap mena de suport temporal s'anomenen *relacions fotogràfiques* i poden treballar amb temps definit per l'usuari com treballen amb qualsevol altre tipus d'atribut. Però sense aquest tipus de suport, la gestió de la informació temporal s'ha de fer trobant la solució de cada cas específic que es vulgui resoldre.

Tenir suport temporal pot resoldre aplicacions que necessiten realitzar, sobretot, consultes històriques, representar canvis retroactius o predictius i emmagatzemar i processar dades dependents del temps. Segons sobre quin tipus de temps es faci aquest suport, obtindrem diferents tipus de relacions temporals.

Relacions en temps de transacció

Les relacions en temps de transacció donen suport a un temps de transacció. I es poden veure com una seqüència d'estats fotogràfics indexats per aquest temps.

Relacions en temps vàlid

Les relacions en temps vàlid donen suport a un temps vàlid. I representen el que ha anat passant o passarà a la nostra relació.

Relacions bitemporals

Les relacions bitemporals donen suport a un temps vàlid i un temps de transacció. I són una seqüència del que ha anat passant o passarà a la nostra relació, indexat pel temps de transacció.

A la figura 2.1, apareguda per primer cop a [SA86] i actualitzada amb els estàndards de [JDB+98], podem veure una representació gràfica dels quatre tipus de relacions temporals que obtenim segons el tipus de temps que escollim per treballar.

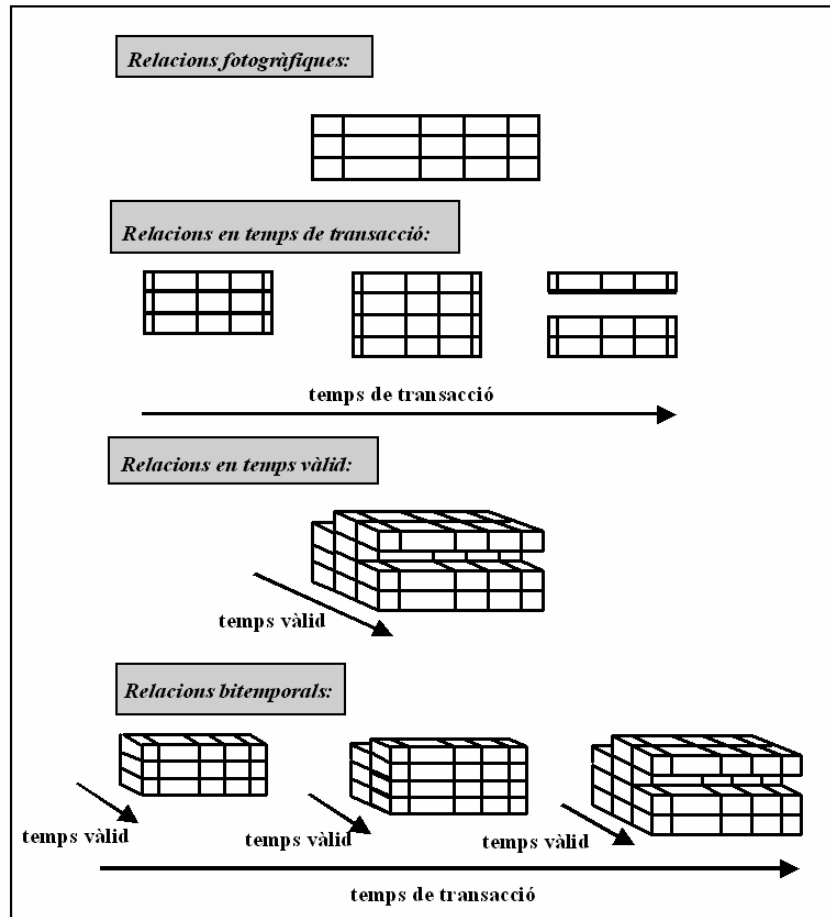


Figura 2.1. Representació gràfica de les relacions fotogràfiques i dels diferents tipus de relacions temporals

Les *bases de dades fotogràfiques* no són més que bases de dades convencionals on totes les relacions no tenen cap tipus de suport temporal.

Si totes les relacions d'una base de dades són en temps vàlid llavors podem parlar de *bases de dades en temps vàlid*. Si totes les relacions són en temps de transacció aleshores parlem de *bases de dades en temps de transacció*. I si totes les relacions són bitemporals, llavors ens trobem davant d'una *base de dades bitemporal*.

Les necessitats reals de les aplicacions fan que en una mateixa base de dades convisquin relacions de diferents tipus. Per això, parlem de *bases de dades temporals* quan les relacions donen suport a qualsevol tipus de temps, exceptuant per descomptat el temps definit per l'usuari.

Per entendre millor l'elecció d'una relació temporal, presentarem un exemple sobre una relació *empleats* amb els atributs *nom* i *sou*, on veurem els avantatges i els desavantatges d'utilitzar els quatre tipus de relacions temporals. Ens centrarem en l'evolució temporal del *sou* de l'empleat *Jordi*, i anirem veient com es comporten les diferents relacions temporals. Començarem donant un sou de 60.000 ptes. a en *Jordi* el dia 1 de gener de 1999.

Relació fotogràfica *Empleats*

<i>nom</i>	<i>sou</i>
Jordi	60.000

Relació en temps de transacció *Empleats*

<i>nom</i>	<i>sou</i>	<i>temps de transacció</i>
Jordi	60.000	1-1-1999

Relació en temps vàlid *Empleats*

<i>temps vàlid</i>	<i>nom</i>	<i>sou</i>
1-1-1999	Jordi	60.000

Relació bitemporal *Empleats*

<i>temps vàlid</i>	<i>nom</i>	<i>sou</i>	<i>temps de transacció</i>
1-1-1999	Jordi	60.000	1-1-1999

Si les necessitats de la relació *empleats* fossin només saber el sou actual que guanyen els empleats de l'empresa, aleshores n'hi hauria prou amb utilitzar una *relació fotogràfica*. En canvi, si volem obtenir informació històrica de l'evolució dels sous dels empleats, haurem de continuar analitzant els altres tres tipus de

relacions. Suposem ara, que el dia 1 de febrer de 1999 augmenten el sou d'en *Jordi* en 20.000 ptes.

A la relació fotogràfica tindriem:

<i>Relació fotogràfica</i>	<i>Empleats</i>
----------------------------	-----------------

<i>nom</i>	<i>sou</i>
Jordi	80.000

En canvi, a les altres tres relacions, comencem a veure l'evolució temporal:

<i>Relació en temps de transacció</i>	<i>Empleats</i>
---------------------------------------	-----------------

<i>nom</i>	<i>sou</i>	<i>temps de transacció</i>
Jordi	60.000	1-1-1999
Jordi	80.000	1-2-1999

<i>Relació en temps vàlid</i>	<i>Empleats</i>
-------------------------------	-----------------

<i>temps vàlid</i>	<i>nom</i>	<i>sou</i>
1-1-1999	Jordi	60.000
1-2-1999	Jordi	80.000

<i>Relació bitemporal</i>	<i>Empleats</i>
---------------------------	-----------------

<i>temps vàlid</i>	<i>nom</i>	<i>sou</i>	<i>temps de transacció</i>
1-1-1999	Jordi	60.000	1-1-1999
1-1-1999	Jordi	60.000	1-2-1999
1-2-1999	Jordi	80.000	1-2-1999

De moment, qualsevol de les tres relacions temporals ens mostra l'evolució dels sous, però veiem què passaria si introduïssim actualitzacions predictives o retroactives com ara l'actualització següent, que és concretament retroactiva: El dia 1 de març de 1999 ens adonem que en *Jordi* hauria d'haver guanyat 100.000 ptes. des de l'1 de febrer de 1999.

Relació en temps de transacció *Empleats*

<i>nom</i>	<i>sou</i>	<i>temps de transacció</i>
Jordi	60.000	1-1-1999
Jordi	80.000	1-2-1999
Jordi	100.000	1-3-1999

Relació en temps vàlid *Empleats*

<i>temps vàlid</i>	<i>nom</i>	<i>sou</i>
1-1-1999	Jordi	60.000
1-2-1999	Jordi	100.000

Relació bitemporal *Empleats*

<i>temps vàlid</i>	<i>nom</i>	<i>sou</i>	<i>temps de transacció</i>
1-1-1999	Jordi	60.000	1-1-1999
1-1-1999	Jordi	60.000	1-2-1999
1-2-1999	Jordi	80.000	1-2-1999
1-1-1999	Jordi	60.000	1-3-1999
1-2-1999	Jordi	100.000	1-3-1999

Després d'aquesta darrera actualització veiem que si escollim una *relació en temps de transacció* només tenim constància del temps d'enregistrament de la informació i per tant perdem la possibilitat de fer actualitzacions predictives i retroactives [JS94]. Si optem per utilitzar una *relació en temps vàlid* aleshores podem expressar actualitzacions predictives i retroactives, però perdem la constància que la relació va tenir informació errònia durant un període de temps. L'única relació que ens permet fer actualitzacions predictives i retroactives, mantenint tota la informació de l'evolució dels sous tal com ha estat i s'ha emmagatzemat realment, és la *relació bitemporal*.

Tenint en compte que les relacions bitemporals són les relacions amb més capacitat expressiva i suposant que hi haurà una clara necessitat de gestió temporal, la nostra tesi se centrarà en bases de dades bitemporals, on totes les

En bases de dades en temps vàlid i en temps de transacció tenim una única dimensió temporal i, per tant, els elements temporals se situen sobre l'únic eix de temps existent. En bases de dades bitemporals, en canvi, els instants, els cronons i els intervals de temps es defineixen amb dues dimensions de temps, i, per tant, hem de parlar d'*elements bitemporals*. Podem veure la representació gràfica d'aquests conceptes en bases de dades bitemporals a la figura 2.3.

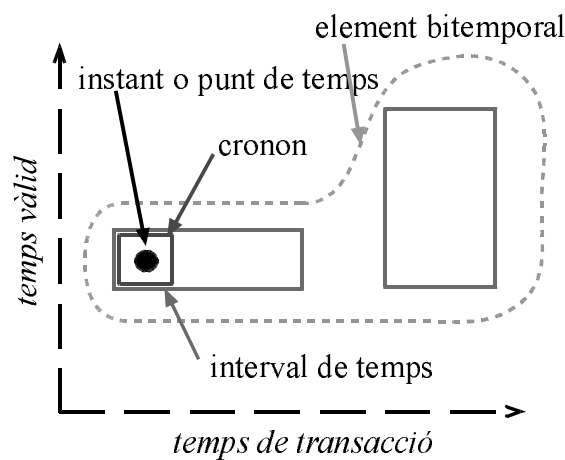


Figura 2.3. Representació gràfica dels conceptes d'instant, cronon, interval de temps i element bitemporal en bases de dades bitemporals

Un *esdeveniment* és un fet instantani que ocorre durant un instant de temps determinat. Quan es dona un esdeveniment l'haurem d'emmagatzemar a la base de dades temporal amb el seu *estampillatge de temps* corresponent, que serà el valor de temps associat a aquest fet.

A l'hora d'estampillar el temps podem escollir entre fer-ho amb instants (o punts de temps) o fer-ho amb intervals de temps. La primera possibilitat, l'estampillatge amb punts de temps, és clarament més simple i fàcil de representar. A més, els punts de temps es poden operar amb els predicats predefinitos estàndard $<$, $>$, \leq , \geq , $=$ i \neq . El gran desavantatge és la ineficiència causada per la gran quantitat d'informació que cal emmagatzemar. La segona possibilitat, l'estampillatge amb intervals de temps, és molt més eficient, però perdem la senzillesa de representació i d'interpretació que ens ofereixen els punts de temps. A més, per operar intervals de temps necessitem operadors específics,

com per exemple els tretze operadors temporals presentats a [All83]: *before*, *after*, *precedes*, *follows*, *during*, *contains*, *equals*, *overlaps*, *starts*, *ends*, *overlapped_by*, *started_by* i *ended_by*. Veiem un exemple d'ambdues possibilitats. Si en *Jordi* tingués un sou de 80.000 ptes. entre l'1 de gener de 1999 i l'1 de febrer de 1999, segons l'estampillatge escollit obtindriem:

Estampillatge amb punts de temps

<i>punts de temps</i>	<i>nom</i>	<i>sou</i>
1-1-1999	Jordi	80.000
2-1-1999	Jordi	80.000
3-1-1999	Jordi	80.000
...
31-1-1999	Jordi	80.000

Estampillatge amb intervals de temps

<i>intervals de temps</i>	<i>nom</i>	<i>sou</i>
[1-1-1999, 31-1-1999) ³	Jordi	80.000

En realitat, si el que volem dir és que en *Jordi* va tenir un sou de 80.000 ptes. tots els dies entre l'1 de gener de 1999 i l'1 de febrer de 1999, només podem fer-ho utilitzant l'estampillatge de punts de temps. Si utilitzem l'estampillatge d'intervals de temps estem dient només que en *Jordi* va guanyar 80.000 ptes. entre l'1 de gener de 1999 i l'1 de febrer de 1999. No podem dir res dels subintervals i molt menys dels punts que componen l'interval [Cho94].

En l'àmbit de la intel·ligència artificial es treballa molt més amb estampillatge amb intervals de temps, mentre que en l'àmbit de les bases de dades s'utilitza molt més l'estampillatge amb punts de temps. Els intervals, doncs, en l'àmbit de les bases de dades amb estampillatge de punts de temps, són només la descripció

³ Hem escollit treballar amb intervals tancats en el punt de temps d'inici i oberts en el punt de temps final seguint [Sri91]. Encara que per donar claredat a aquest capítol, quan no usem la notació d'intervals de temps, hem preferit que el punt de temps de finalització no sigui igual al punt de temps d'inici següent.

d'un conjunt de punts de temps on s'indica el punt de temps d'inici i el punt de temps de finalització. Veiem l'exemple anterior amb punts de temps representats en forma d'interval:

Estampillatge amb punts de temps representats en forma d'interval

<i>punt de temps d'inici (pi)</i>	<i>punt de temps de finalització (pf)</i>	<i>nom</i>	<i>sou</i>
1-1-1999	31-1-1999	Jordi	80.000

En el capítol 7 tractarem més a fons la semàntica de l'estampillatge de punts de temps representada en forma d'interval i veurem que fins i tot en aquest cas no és tan senzill deduir la informació en els subinterval d'un determinat interval de temps.

En bases de dades bitemporals hi ha dos tipus de temps a emmagatzemar: el temps vàlid i el temps de transacció; per tant, una de les possibilitats de fer l'estampillatge amb punts de temps representats en forma d'interval hauria de tenir un punt de temps d'inici i un punt de temps de finalització per a ambdós tipus de temps. Aquest tipus de representació rep el nom de tuples amb quatre estampillatges de temps [JSS94] i és la representació més coneguda i utilitzada.

Tuples amb quatre estampillatges de temps

<i>punt de temps vàlid d'inici (pvi) / punt de temps vàlid de finalització (pvf)</i>	<i>punt de temps de transacció d'inici (pti) / punt de temps de transacció finalització (ptf)</i>	<i>nom</i>	<i>sou</i>
1-1-1999 / 31-1-1999	1-1-1999 / 31-1-1999	Jordi	80.000

A [JSS92a] trobem una altra possibilitat per emmagatzemar el temps vàlid i el temps de transacció amb punts de temps representats en forma d'interval: els tuples amb cinc estampillatges de temps. En aquest cas tindriem el temps vàlid d'inici, el temps vàlid de finalització, el temps de transacció quan el temps vàlid d'inici s'enregistra, el temps de transacció quan el temps vàlid de finalització s'enregistra i el temps de transacció de l'esborrat lògic del tuple.

Tuples amb cinc estampillatges de temps

<i>punt de temps vàlid d'inici (pvi) / punt de temps vàlid de finalització (pvf)</i>	<i>punt de temps de transacció de pvi / punt de temps de transacció de pvf / punt de temps de transacció d'esborrat</i>	<i>nom</i>	<i>sou</i>
1-1-1999 / 31-1-1999	1-1-1999 / 1-1-1999 / 31-1-1999	Jordi	80.000

Fins ara l'estampillatge del temps l'hem associat a tuples, però també es podria associar, molt més eficientment, a atributs si no fos necessari tenir relacions en primera forma normal. Veiem la diferència entre aquests dos tipus d'associacions per a l'evolució del sou d'en *Jordi*, incorporant un nou atribut *departament* a la relació *empleats*, que ens indiqui a quin departament pertany l'empleat.

Estampillatge amb punts de temps representats en forma d'interval associats a tuples

<i>pi / pf</i>	<i>nom</i>	<i>sou</i>	<i>departament</i>
1-1-1999/31-1-1999	Jordi	80.000	LSI
1-2-1999/31-3-1999	Jordi	90.000	LSI
1-4-1999/31-5-1999	Jordi	90.000	AC
1-6-1999/31-12-1999	Jordi	100.000	AC

Estampillatge amb punts de temps representats en forma d'interval associats a atributs

<i>pi / pf</i>	<i>nom</i>	<i>pi / pf</i>	<i>sou</i>	<i>pi / pf</i>	<i>departament</i>
1-1-1999/ 31-12-1999	Jordi	1-1-1999/ 31-1-1999	80.000	1-1-1999/ 31-3-1999	LSI
		1-2-1999/ 31-5-1999	90.000	1-4-1999/ 31-12-1999	AC
		1-6-1999/ 31-12-1999	100.000		

En l'associació a tuples cada vegada que es produeix un canvi en un atribut hem d'inserir un nou tuple, mentre que en l'associació a atributs només hem

d'estampillar el temps en l'atribut afectat sense tocar els altres. L'elecció del tipus d'associació depèn de la necessitat de treballar amb primera forma normal o sense. En l'àrea de bases de dades, malgrat l'eficiència de l'associació a atributs, la majoria de treballs s'han desenvolupat amb associació a tuples.

2.3. Models de dades temporals

El model de dades que seguirem en el nostre treball és un model que unifica la majoria de models de dades temporals existents. A la figura 2.4 podem veure la situació d'aquest model, proposat per [JSS94], tenint en compte les tasques de disseny lògic i físic de la base de dades, els models de representació de dades i els formats de presentació.

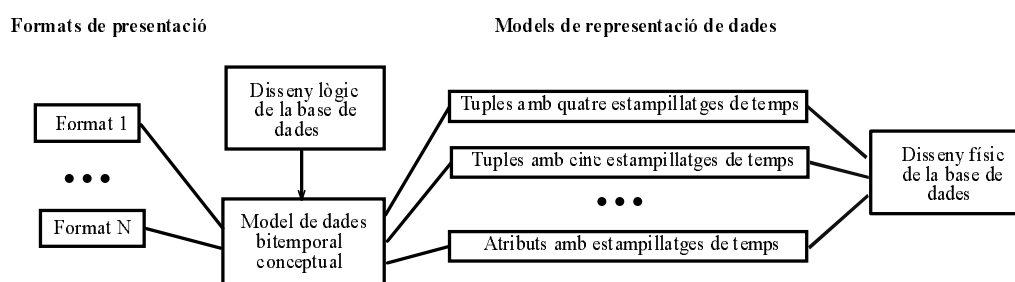


Figura 2.4. Model de dades bitemporal

Pel que fa als formats de presentació, no és el nostre objectiu lligar-nos a cap format concret; en el capítol 4 presentarem la lògica de predicats de primer ordre com a llenguatge de definició i manipulació de bd-dbts amb tractament explícit de punts de temps. No ens endinsarem en el disseny lògic d'una base de dades bitemporal, però sí tractarem, sense aprofundir gaire, el model conceptual i un dels possibles models de representació en els apartats següents.

2.3.1. La base de dades temporal abstracta

Abans de començar a veure el model de dades bitemporal, introduïrem el concepte de *base de dades temporal abstracta*, necessari per a la comprensió del model. Una base de dades temporal abstracta, definida a [Cho94], és capaç de capturar el significat d'una base de dades temporal independentment de la seva

representació. Cada model específic de base de dades temporal, anomenada *base de dades temporal concreta*, no és més que una representació d'una base de dades temporal abstracta. Dues bases de dades temporals concretes són equivalents si representen la mateixa base de dades temporal abstracta.

Una base de dades temporal abstracta es pot veure des de moltes, però equivalents, perspectives. Veiem dues de les principals perspectives de base de dades temporal abstracta, la fotogràfica i la d'estampillatge de temps.

Donats un domini temporal T , per simplificar d'una dimensió, i un domini de dades U , considerem una relació R d'aritat n i una relació temporal abstracta R' d'aritat $n+1$ on:

$$(a_1, \dots, a_n, t) \in R' \text{ si } R(a_1, \dots, a_n) \text{ ocorre en l' instant } t \text{ on } a_1, \dots, a_n \in U$$

Més formalment, una base de dades es pot veure com una estructura finita $D = (U, R_1, \dots, R_k)$ pel llenguatge de primer ordre L_D , que conté símbols de relació per a totes les relacions R_1, \dots, R_k de la base de dades i símbols constants per a tots els elements d' U . La base de dades temporal abstracta corresponent serà una estructura $D' = (U, T, R'_1, \dots, R'_k)$ pel llenguatge de primer ordre L'_D , que conté un nou símbol de relació temporal per a cada relació R'_i de la base de dades i símbols constants per a tots els elements d' U i possiblement per a alguns elements de T .

En la perspectiva fotogràfica, una base de dades temporal abstracta D' és un conjunt de funcions $\{f_{R_1}, \dots, f_{R_k}\}$ on per a cada instant $t \in T$

$$f_{R_i}(t) = \{(a_1, \dots, a_n) : R_i(a_1, \dots, a_n) \text{ ocorre en temps } t\}$$

En la perspectiva d'estampillatge de temps, una base de dades temporal abstracta D' és un conjunt de funcions $\{g_{R_1}, \dots, g_{R_k}\}$ on

$$g_{R_i}(a_1, \dots, a_n) = \{t : R_i(a_1, \dots, a_n) \text{ ocorre en temps } t\}$$

En totes dues perspectives cada relació no està en primera forma normal. En la perspectiva fotogràfica el valor del segon atribut de la relació són un conjunt de tuples i, en la perspectiva d'estampillatge de temps, són un conjunt d'instantes de temps.

Veiem un exemple a la figura 2.5 de totes dues perspectives sobre la relació *empleats*, on per simplificar suposarem que només tenim l'atribut *departament* i l'únic temps que utilitzarem serà el temps vàlid. L'empleat *Jordi* és contractat pel departament d'*LSI* per un període del 5 de juny de 1999 al 15 de juny de 1999. A més sabem que l'empleat *Jordi* canvia al departament d'*AC* per un període del 16 de juny de 1999 al 30 de juny de 1999 i que es contracta la *Maria* al departament d'*LSI* per un període del 16 de juny de 1999 al 30 de juny de 1999.

Perspectiva fotogràfica

t	empleats(nom,departament)
5/6	empleats(Jordi,LSI)
6/6	empleats(Jordi,LSI)
...	
16/6	empleats(Jordi,AC) empleats(Maria,LSI)
...	

Perspectiva d'estampillatge de temps

empleats		
nom	departament	t
Jordi	LSI	5/6...15/6
Jordi	AC	16/6...30/6
Maria	LSI	16/6...30/6

Figura 2.5. Exemple de la perspectiva fotogràfica i la perspectiva d'estampillatge de temps per a la relació *empleats*

La definició d'una base de dades abstracta ha donat lloc a la idea d'equivalència fotogràfica [JSS93]. L'equivalència fotogràfica es basa en el fet que dues instàncies d'una relació són equivalents si totes les seves fotografies, tant en temps vàlid com en temps de transacció, són idèntiques. En el model de dades bitemporal, l'equivalència fotogràfica servirà per demostrar que una instància d'una relació bitemporal conceptual és independent de la instància de l'esquema de representació que s'utilitzi.

2.3.2. El model bitemporal conceptual

El model bitemporal conceptual associa a cada tuple d'una instància d'una relació bitemporal conceptual els valors de dos dominis de temps ortogonals: temps de transacció i temps vàlid. L'esquema d'una relació bitemporal conceptual *R* consisteix en un nombre arbitrari d'atributs, A_1, A_2, \dots, A_n , i un atribut estampillat implícit, *T*. D'aquesta manera, un tuple, $x = (a_1, a_2, \dots, a_n | t_b)$,

en una instància d'una relació bitemporal conceptual, $r(R)$, consisteix en un nombre de valors d'atributs associat amb el valor de temps estampillat. Associat a cada tuple tindrem, doncs, un element bitemporal t_b en l'espai de dues dimensions format per temps de transacció i temps vàlid. Com no es pot donar el cas que dos tuples tinguin els mateixos valors d'atributs explícits en una instància d'una relació bitemporal conceptual, la història completa d'un fet es pot expressar en un únic tuple.

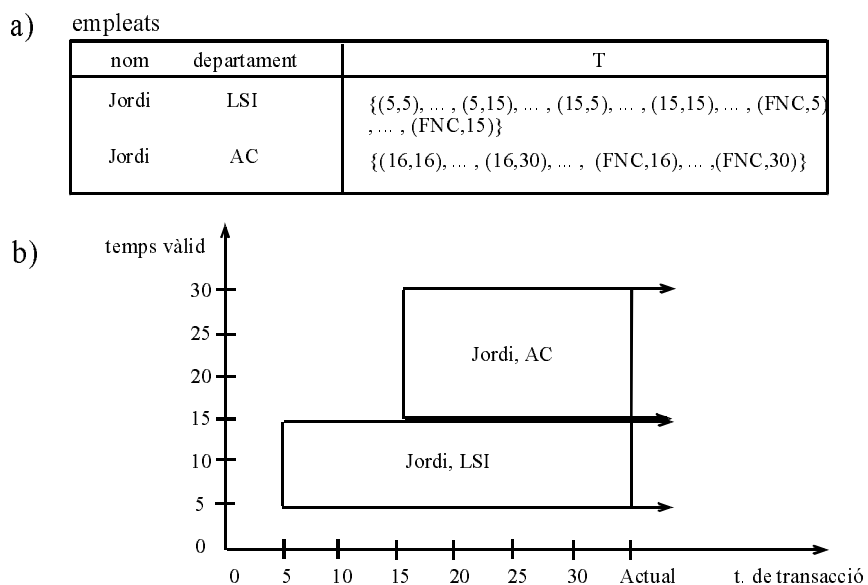


Figura 2.6. a) Model de dades bitemporal conceptual de la relació *empleats* amb instàncies d'aquesta relació. b) Representació gràfica del model de dades bitemporal conceptual de la relació *empleats* i de les instàncies d'exemple

Veiem un exemple del model bitemporal conceptual de la relació *empleats*, on per simplificar suposarem que només tenim els atributs *nom* i *departament*. L'empleat *Jordi* és contractat pel departament d'*LSI* el dia 5 de juny de 1999 per un període del 5 de juny de 1999 al 15 de juny de 1999. El dia 16 de juny de 1999 l'empleat *Jordi* canvia al departament d'*AC* per un període del 16 de juny de 1999 al 30 de juny de 1999. A la figura 2.6 es mostra la relació bitemporal conceptual *empleats* i la seva representació gràfica.

A la relació bitemporal conceptual *empleats* de la figura 2.6.a, l'atribut de temps estampillat T mostra cada cronon de temps de transacció associat a cada cronon de temps vàlid com un conjunt de parells ordenats. El temps de transacció actual rep la notació *Actual* [CDI+97]. Per simbolitzar que, encara que el temps de transacció vagi passant, sempre i en tot moment tindrem el temps de transacció més gran possible, podem utilitzar la variable *Fins Nous Canvis (FNC)* [CDI+97].

En els eixos de coordenades de la representació gràfica de la relació bitemporal conceptual *empleats* de la figura 2.6.b hem escollit com a cronon els dies dels mesos de l'any. Per simplificar però, hem posat només els dies del mes de gener agrupats de cinc en cinc. Les fletxes dels elements bitemporals signifiquen que mentre la informació no s'esborri lògicament, s'anirà actualitzant a mesura que va passant el temps de transacció.

2.3.3. Model de representació

Encara que en el model de dades bitemporal es pot associar el model conceptual bitemporal amb qualsevol model de representació, nosaltres hem escollit com a model de representació els tuples amb quatre estampillatges de temps, per la seva senzillesa per expressar els tuples en primera forma normal.

Donat un esquema de relació bitemporal R que conté atributs A_1, A_2, \dots, A_n, T on T és un atribut definit en el domini d'elements bitemporals. Aquest esquema és representat per l'esquema de relació següent:

$$R = (A_1, \dots, A_n, T_{vi}, T_{vf}, T_{ti}, T_{tf})$$

On els atributs $T_{vi}, T_{vf}, T_{ti}, T_{tf}$ contenen els punts d'inici i finalització del temps vàlid i els punts d'inici i finalització del temps de transacció, respectivament. Aquests quatre valors representen cronons bitemporals en una regió rectangular. La idea és dividir la regió completa, coberta per l'element bitemporal d'un tuple d'una instància de la relació conceptual, en un conjunt de rectangles, i aleshores representar el tuple conceptual com un conjunt de tuples de valor equivalent, un per cada rectangle. Cada funció que cobreixi un element bitemporal x/T d'un tuple bitemporal x ha de satisfer les dues propietats següents:

1. Cada cronon bitemporal a x/T ha d'estar contingut en un rectangle.
2. Cada cronon bitemporal en un rectangle ha d'estar contingut a x/T .

A la figura 2.7 es pot veure la relació en primera forma normal corresponent a la relació conceptual de la figura 2.6.

empleats		T_{vi}	T_{vf}	T_{ti}	T_{tf}
nom	departament				
Jordi	LSI	5/6	15/6	5/6	FNC
Jordi	AC	16/6	30/6	16/6	FNC

Figura 2.7. Model de dades bitemporal de representació de la relació empleats de la figura 2.6

A [JSS94] podem trobar les funcions per convertir una instància d'una relació bitemporal conceptual en una instància de l'esquema de representació i a la inversa. Mitjançant l'equivalència fotogràfica es demostra que partint de la mateixa instància d'una relació bitemporal conceptual, podem passar a instàncies d'esquemes de representació diferents, però semànticament equivalents, i que donades les instàncies d'esquemes de representació diferents podem obtenir la mateixa instància d'una relació bitemporal conceptual.

3. Bases de dades deductives

La necessitat d'utilitzar temps per emmagatzemar fets és molt evident en certes aplicacions, però els desavantatges de l'ocupació d'espai i la dificultat de representar informació temporal infinita que això suposa són un greu problema. Aquest problema es pot millorar considerablement si utilitzem les regles de deducció de les bases de dades deductives al mateix temps que les bases de dades temporals. A continuació presentem el concepte de base de dades deductiva (no temporal) tal com ha estat definit també per [Ten92] i [Urp93].

Una base de dades deductiva D està formada per tres conjunts finits: un conjunt F de fets, un conjunt R de regles de deducció i un conjunt I de restriccions d'integritat. El conjunt de fets s'anomena *base de dades extensional* i el conjunt de regles de deducció s'anomena *base de dades intensional*.

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

Fets, regles i restriccions d'integritat seran formulats amb lògica de primer ordre seguint [GGG+98]. Un terme és una constant o una variable. Si a_1, \dots, a_n són termes i p és un predicat n -ari, aleshores $p(a_1, \dots, a_n)$ és un àtom, diguem-ne A . Tot àtom és una fórmula. A més, si ϕ i ψ són fórmules, aleshores $\phi \wedge \psi$, $\phi \vee \psi$, $\neg\phi$, $\phi \rightarrow \psi$ també són fórmules. Un fet és un àtom *instanciat*, és a dir que no conté cap variable.

3.1. Regles de deducció

Una regla de deducció 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 (és a dir, 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* (o *cap*) i els L_1, \dots, L_n s'anomenen *condicions* (o *cos*). Suposarem que els termes de la conclusió són variables diferents i que els termes de les condicions poden ser variables o constants. Un predicat derivat pot estar definit per més d'una regla de deducció.

Els predicats de les condicions poden ser ordinaris o avaluable. Els primers són predicats bàsics o derivats, mentre que els segons són predicats que poden ser avaluats sense accedir a la base de dades, com per exemple els predicats de comparació ($=, <, \leq, >, \geq$ i \neq).

El fonament teòric de les bases de dades deductives el podem trobar a *Datalog* [Ull85]. *Datalog* és restrictiu perquè no permet l'ús de funcions, negació i predicats de comparació. Un sistema de bases de dades deductiu basat només en *Datalog* pur no ens permetria expressar requeriments d'aplicacions reals i és per això que s'han fet diversos esforços per estendre *Datalog*, per permetre l'ús de funcions, negació i predicats de comparació, tal com s'explica a [Böh94a].

Els predicats de comparació es poden integrar amb facilitat, cosa que garanteix que els dos arguments dels predicats de comparació tinguin un valor instanciat en temps d'execució. Això es pot aconseguir amb l'anomenada condició *segura* [CGT90].

La negació també es pot permetre sempre que no es donin cicles recursius. Les bases de dades *estratificades* [ABW88] poden ser dividides en capes, on un predicat només es pot utilitzar negat si ha estat definit sense negació en una capa estrictament inferior. 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 $i \leq j$.
- (ii) si $Q \in S_i$ és el símbol de predicat d'una condició negativa de P , aleshores $i < j$.

La *permissibilitat* comprova si una fórmula normalitzada [Llo87] pot ser tractada o no. L'objectiu de la normalització és aconseguir una fórmula que pugui ser convenientment tractada en el seu subsegüent processament. Encara que hi ha diferents definicions de *permissibilitat*, a nosaltres ens interessarà la definició de [Llo87], que garanteix que la computació de la resolució *Standard Linear Deduction with Negation as Failure (SLDNF)* mai no arriba a un punt en què l'objectiu conté només literals negats no instanciats, encara que s'hagin d'introduir predicats auxiliars i les regles puguin ser avaluades només descendentment. És a dir, qualsevol variable que ocorre en una regla deductiva té una ocurrència en una condició positiva d'un literal ordinari. Això assegura que totes les condicions negatives es poden instanciar completament abans de ser avaluades per la regla de negació de fracàs finit.

Respecte a les funcions, atesa la indecidibilitat de determinar si la presència de símbols de funció ens porta a un mínim punt fix finit o no, no treballarem explícitament amb funcions, com la majoria d'enfocaments han decidit fer. Excepcionalment, permetrem l'ús de funcions aritmètiques bàsiques, i deixarem la responsabilitat de la seva terminació a la persona que ha decidit fer-les servir.

3.2. 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 treballarem amb restriccions que tenen forma de denegació:

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

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 fer-ho, primer cal aplicar la transformació en *forma de rang* [Dec89] i, a continuació, utilitzar el procediment descrit a [LT84]. La transformació en forma de rang és necessària perquè evita que la segona transformació introdueixi *entrebanca* 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 *ic* (de la mateixa manera que [DW89] i [Kow78]). Així, les restriccions tenen la mateixa forma que les regles de deducció i, llavors, les anomenem regles d'integritat. Per exemple, la denegació anterior es rescriuria com $ic \leftarrow L_1 \wedge \dots \wedge L_n$. Les restriccions d'integritat també han de ser permissibles.

3.3. El mètode dels esdeveniments interns

En aquesta tesi presentem un mètode per a la comprovació de restriccions d'integritat en bd-bdts. Aquest mètode és una extensió del *mètode dels esdeveniments interns* per a la comprovació de restriccions d'integritat en bases de dades deductives [Oli91], que a la vegada 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 són: en l'àrea de les bases de dades deductives, l'actualització de vistes [TO92], el manteniment de restriccions d'integritat [TM99], la monitorització de canvis [UO92], la generació d'especificacions de transaccions [Pas97] i la inclusió de consultes [FTU99]; i en el camp dels models conceptuals deductius, el disseny de transaccions [SO94], el raonament sobre aquestes [CO92] i l'explicitació de les interaccions entre els objectes en especificacions orientades a objectes [QO93].

El *mètode dels esdeveniments interns* augmenta la base de dades ($A(D)$) amb un conjunt de regles, anomenades de transició i d'esdeveniment, que defineixen explícitament els canvis (*TR*, és a dir, les insercions i els esborrats) produïts i/o induïts per una actualització de la base de dades. Mitjançant aquestes regles i la resolució SLDNF, es pot fer la comprovació de les restriccions d'integritat en bases de dades deductives, tal com es pot veure a la figura 3.1.

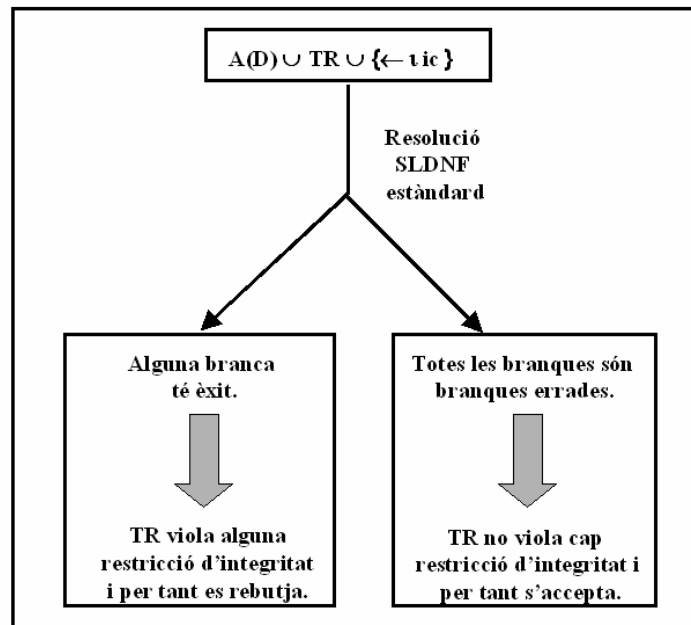


Figura 3.1. El mètode dels esdeveniments interns per a una base de dades deductiva augmentada $A(D)$, una transacció TR i una restricció d'integritat ic

A [Oli91], el mètode dels esdeveniments interns és comparat amb altres mètodes de comprovació de restriccions d'integritat, com per exemple [NY78], [KSS87] i [BDM88]. La seva simplicitat i el tractament uniforme de les restriccions d'integritat estàtiques i de transició, a més del guany de temps a causa de la seva implementació en un sistema Prolog sense requerir l'ús de cap metaintèrpret, fan que sigui el mètode que hàgim escollit per adaptar a bd-dbts. Tanmateix, el mètode té un alt cost d'espai per emmagatzemar les regles de transició i d'esdeveniments. Aquest cost s'ha de tenir especialment en compte quan treballem amb bases de dades amb un gran nombre de regles de deducció i restriccions d'integritat. Precisament, aquest en principi inconvenient, en el cas de bases de dades temporals, no és significat, ja que el nombre de regles de deducció i restriccions d'integritat és petit comparat amb el gran nombre de fets que s'han d'emmagatzemar.

Tenint en compte que el nostre mètode és una extensió del mètode dels esdeveniments interns, a continuació introduïrem els conceptes bàsics d'aquest mètode per fer més clara la nostra aportació en el capítol 6.

Tenim una base de dades D , una transacció TR i la base de dades modificada D' . Tenint en compte que la transacció TR consistirà en un conjunt de fets que caldrà inserir o esborrar, el mètode dels esdeveniments interns defineix que TR indueix una transició des de D fins a D' .

Tenint en compte les regles de deducció, TR també pot induir altres modificacions sobre predicats derivats. Tenim un predicat derivat p a D i p' , que seria el mateix predicat derivat avaluat a D' . Assumint que un fet $p(k)$ és cert a D , on k és un vector de constants, es poden presentar dos casos:

Que $p'(k)$ es continuï donant a D' ($p(k)$ i $p'(k)$ tots dos són certs) (1)

Que $p'(k)$ no es doni a D' ($p(k)$ és cert, però $p'(k)$ és fals) (2)

I assumint que $p'(k)$ és cert a D' , també són possibles dos casos:

Que $p(k)$ es continuï donant a D ($p(k)$ i $p'(k)$ tots dos són certs) (3)

Que $p(k)$ no es doni a D ($p'(k)$ és cert, però $p(k)$ és fals) (4)

En el cas (2), diem que s'ha produït un *esdeveniment intern d'esborrat* durant la transició. L'esdeveniment d'esborrat es denota $\delta p(k)$.

En el cas (4), diem que s'ha produït un *esdeveniment intern d'inserció* durant la transició. L'esdeveniment d'inserció es denota $\wp(k)$.

Formalment, a cada predicat bàsic, derivat o d'inconsistència p se li associa un predicat d'esdeveniment intern d'inserció \wp i un esdeveniment intern d'esborrat δp , definits com a:

$$\forall X (\wp(X) \leftrightarrow p'(X) \wedge \neg p(X)) \quad (5)$$

$$\forall X (\delta p(X) \leftrightarrow p(X) \wedge \neg p'(X)) \quad (6)$$

on X és un vector de variables. De (5) i (6) es pot obtenir:

$$\forall X (p'(X) \leftrightarrow (p(X) \wedge \neg \delta p(X)) \vee \wp(X)) \quad (7)$$

$$\forall X (\neg p'(X) \leftrightarrow (\neg p(X) \wedge \neg \wp(X)) \vee \delta p(X)) \quad (8)$$

Si p és un predicat bàsic, aleshores \wp i $\delta\wp$ representen insercions i esborrats de fets, respectivament. Si p és un predicat derivat o d'inconsistència, a partir de (7) i (8) es generen les seves *regles de transició*. Una regla de transició defineix un predicat p en el nou estat (p') en termes dels predicats que hi ha a l'estat actual i els esdeveniments interns. Per deduir les insercions o esborrats induïts durant una transició utilitzarem les *regles d'esdeveniments interns* d'inserció o esborrat. En el cas dels predicats d'inconsistència només tindrem esdeveniments interns d'inserció. Els esdeveniments interns d'inserció que tinguin lloc durant la transició correspondran a violacions de les restriccions d'integritat.

El *mètode dels esdeveniments*, aplicació del *mètode dels esdeveniments interns* per a l'actualització de vistes en bases de dades deductives, és presentat i se'n demostra la correctesa i consistència a [Ten92]. D'aquesta manera, nosaltres assumim que partim d'un mètode correcte i consistent, abans de fer l'adaptació per a bases de dades bitemporals.

4. Bases de dades deductives bitemporals

El nostre objectiu és analitzar en profunditat el problema de la comprovació de la integritat en bases de dades bitemporals, presentades en el capítol 2, aprofitant tota la potència de les regles de deducció que ens ofereixen les bases de dades deductives que hem vist en el capítol anterior. Les bd-dbts, conceptualment, no són més que bases de dades deductives amb un terme més en els predicats bàsics i derivats, definit sobre un domini bitemporal.

Abans d'entrar en l'estudi de les restriccions d'integritat temporals en el capítol següent, presentarem el llenguatge de lògica de predicats de primer ordre que utilitzarem per actualitzar la base de dades deductiva bitemporal. Només veurem les operacions d'inserció, esborrat i *modificació* (tractada com una inserció després d'un esborrat), ja que les operacions de consulta no actualitzen la base de dades i, per tant, no són rellevants per a la comprovació de la integritat. També compararem les nostres operacions amb [Böh94a], que, tal com s'explica a [Böh95], en aquest moment és l'únic prototipus de bd-dbt que existeix.

Una base de dades deductiva bitemporal D consisteix en tres conjunts finits: un conjunt de fets F , un conjunt de regles de deducció R , i un conjunt de restriccions d'integritat temporals I . Fets, regles i restriccions d'integritat temporals es formulen en el següent llenguatge de lògica de predicats de primer ordre, seguint la notació de [GGG+98].

Les constants són caràcters, números o punts de temps. Els punts de temps pertanyen a un domini temporal T . T és isomorf al conjunt de nombres naturals,

sobre el que es defineix l'ordre lineal $<_T$, on donats dos punts de temps t_i i t_j : $t_i <_T t_j$ vol dir que t_i és anterior a t_j [GL95]. El conjunt T és la base per incorporar les dimensions temporals a la base de dades deductiva fent servir la representació que utilitza un punt de temps vàlid d'inici, un punt de temps vàlid de finalització, un punt de temps de transacció d'inici i un punt de temps de transacció de finalització [JSS93]. Per referir-nos al punt de temps de transacció inicial utilitzarem la notació t_{i0} , i per al punt de temps vàlid inicial, t_{v0} [CT98b]. El punt de temps de transacció que, encara que el temps de transacció vagi passant, simbolitzarà sempre i en tot moment el temps de transacció més gran possible, serà *FNC*, i el punt de temps vàlid que simbolitzarà que el fet modelat continua sent vàlid serà *Ara*. [CDI+97].

Els identificadors són seqüències arbitràries de caràcters que comencen amb una lletra i poden contenir lletres i números. Les variables es denoten com identificadors que comencen amb una lletra majúscula.

Els símbols de predicats són identificadors que comencen amb una lletra minúscula. Alguns símbols de predicats són predefinits, com és el cas dels operadors de comparació que s'utilitzen habitualment: =, <, >, ≤, ≥ i ≠.

Un terme és una constant o una variable. Si X_1, \dots, X_n són termes no temporals, T_{vi} , T_{vf} , T_{ti} i T_{tf} són termes temporals, que denoten els punts de temps d'inici i finalització del temps vàlid i els punts de temps d'inici i finalització del temps de transacció respectivament, i p és un predicat n -ari aleshores $p(X_1, \dots, X_n, T_{vi}, T_{vf}, T_{ti}, T_{tf})$ és un àtom, diguem-li A , que significa que $p(X_1, \dots, X_n)$ és vàlid des de T_{vi} fins a T_{vf} , i es emmagatzemat des de T_{ti} fins a T_{tf} .

Una regla de deducció és una fórmula de la forma:

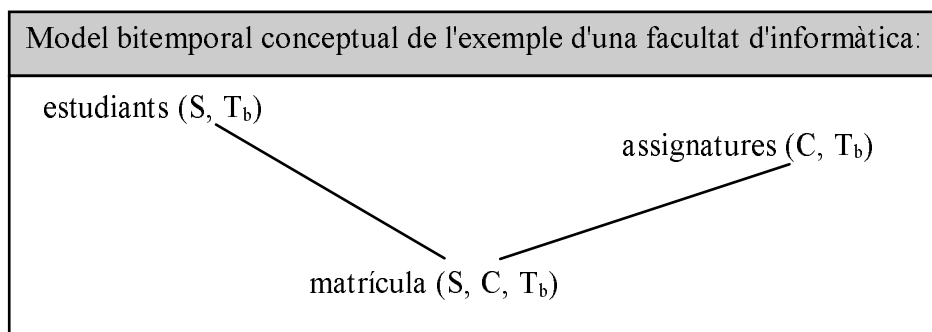
$$A \leftarrow L_1 \wedge \dots \wedge L_n \text{ amb } n \geq 1,$$

on l'àtom A s'anomena conclusió de la regla i L_1, \dots, L_n són literals que representen condicions. Un literal és un àtom o un àtom negat. Totes les variables a A, L_1, \dots, L_n s'assumeixen universalment quantificades davant la fórmula.

Una regla amb una conclusió buida normalment es considera una consulta. Tal com es demostra a [CTB01], suposarem que les consultes són independents del domini temporal.

Una regla instanciada, amb unes condicions buides s'anomena un fet.

Aplicarem el llenguatge sobre l'exemple d'una facultat d'informàtica que ofereix assignatures en què es matriculen els estudiants. Seguint el model de dades bitemporal conceptual del capítol 2 [JSS94], presentarem l'esquema conceptual de l'exemple, en el qual, per simplificar, hem posat dins l'esquema dels predicats bàsics el mínim nombre d'atributs necessaris: S per identificar l'estudiant i C per donar nom a l'assignatura. Recordem que T_b és un element bitemporal on trobarem el temps vàlid i el temps de transacció.



4.1 Insercions

Per denotar que es produeix una inserció utilitzarem el símbol ι . Una inserció sempre es produeix en temps de transacció *Actual* i continuarà sent vàlida fins que no l'esborrem lògicament. Això vol dir que no caldrà donar explícitament el temps de transacció en fer qualsevol tipus d'inserció, sinó que aquest temps es pot afegir de forma automàtica en l'instant en què s'emmagatzema a la base de dades. Com en el nostre cas, a [Böh94a] el temps de transacció també es manipulat implícitament.

Si suposem que en l'exemple de la facultat d'informàtica volem oferir l'assignatura de *bases de dades* des del *15 de setembre de 1999* fins al *15 de desembre de 1999*, introduint aquesta informació el *15 de juliol de 1999*, hauríem de fer la inserció del fet següent:

ι assignatures (*bases de dades, 15-9-1999, 15-12-1999*),

on *Actual* serà el 15 de juliol de 1999. Però simplifiquem la informació temporal utilitzant nombres naturals en lloc de dates reals, posant per exemple:

assignatures (bases de dades, 4, 6), on *Actual* serà 2.

En bd-dbt, quan fem una inserció ens podem trobar amb les situacions que es descriuen a la figura 4.1.

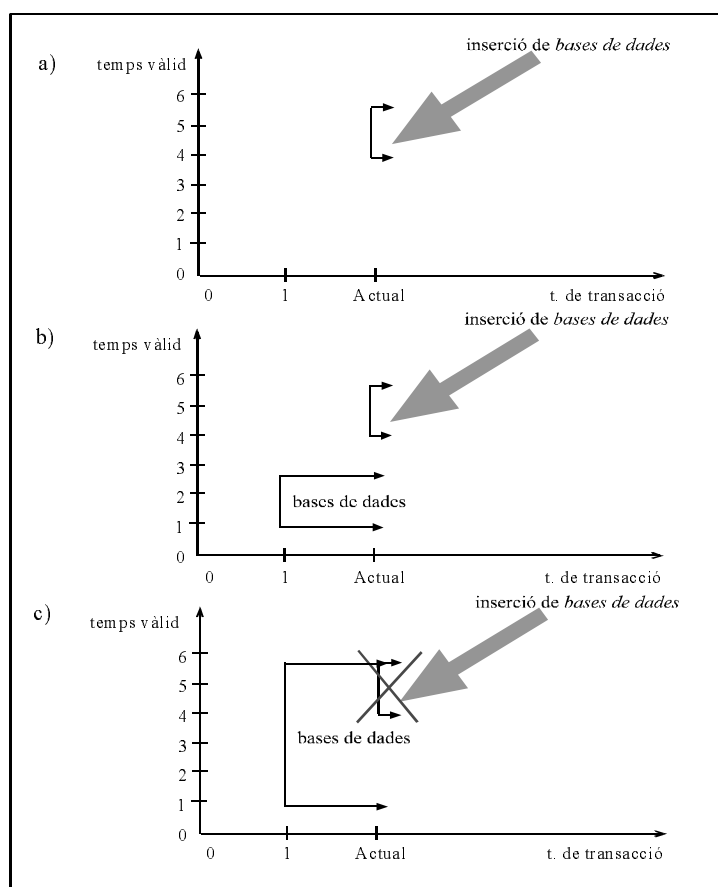


Figura 4.1. Inserció de l'assignatura bases de dades en una bd-dbt

A la figura 4.1.a ens trobem el cas que el curs *bases de dades* no s'havia ofert mai i, per tant, es pot inserir. A la figura 4.1.b veiem que el curs de *bases de dades* ja s'havia ofert en un temps vàlid anterior, però no coincideix amb el temps

vàlid que volem inserir. En aquest cas, per poder fer la inserció primer esborrarem la informació que teníem fins ara sobre el fet a inserir i tornarem a inserir-la però afegint la nova informació que ens ve donada per l' l . En canvi, a la figura 4.1.c ens trobem un cas on no es pot fer la inserció perquè el curs de *bases de dades* ja s'està oferint actualment amb un temps vàlid que coincideix en part amb el que volem inserir, en aquest cas la inserció és rebutjada, ja que en realitat es tracta d'una modificació. Aquests tres casos es presenten a [JSS94] per inserir fets en una relació bitemporal. L'operació d'inserció de [Böh94a] no té en compte aquestes tres opcions a l'hora d'inserir un fet. En canvi, però, preveu la possibilitat de ser definida per una regla de deducció i , per tant, de poder inserir només aquells fets que compleixen unes determinades condicions.

A continuació, definirem formalment els tres casos que podem trobar a l'hora de fer una inserció:

Donada la inserció següent d'un fet f :

$$f(a_1, \dots, a_n, t_v)$$

On t_v és l'element temporal corresponent al temps vàlid amb temps vàlid d'inici t_{vi} i temps vàlid de finalització t_{vf} , t_b és l'element bitemporal amb temps vàlid i temps de transacció i t_{actual} serà el valor que tingui el temps de transacció actual.

Es poden donar els casos següents:

- $f \cup \{(a_1, \dots, a_n \mid \{t_{actual}, FNC\} \times t_v)\}$
 si $\neg \exists t_b (a_1, \dots, a_n \mid t_b) \in f$
- $f - \{(a_1, \dots, a_n \mid t_b)\} \cup \{(a_1, \dots, a_n \mid t_b \cup \{t_{actual}, FNC\} \times t_v)\}$
 si $\exists t_b ((a_1, \dots, a_n \mid t_b) \in f \wedge (\{t_{actual}, FNC\} \times t_v) \cap t_b = \emptyset)$
- f si un altre cas

També es podria donar el cas que volguéssim inserir una informació temporal amb temps vàlid d'inici conegut, però sense temps vàlid de finalització. Per exemple aquest seria el cas que sabéssim que l'assignatura de *bases de dades* començarà el dia *15 de setembre de 1999* però encara no coneguéssim quan finalitzarà. En aquesta situació en lloc del temps vàlid de finalització, hauríem de posar la variable *Ara* [CDI+97]. *Ara* simbolitzarà que el curs de *bases de dades* s'anirà oferint mentre no s'introdueixi un temps vàlid de finalització. La inserció d'aquest nou fet seria:

tassignatures (bases de dades, 15-9-1999, Ara) o, simplement,
tassignatures (bases de dades, 15-9-1999),

però simplifiquem la informació temporal que mostrem a la figura 4.2, suposant que el curs de *bases de dades* no s'ha ofert mai abans, fent:

tassignatures (bases de dades, 4, Ara) o *tassignatures (bases de dades, 4)*.

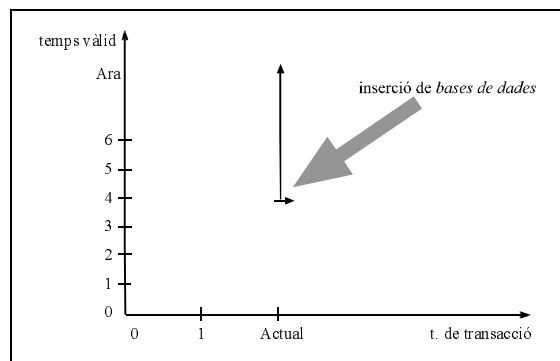


Figura 4.2. Inserció de l'assignatura bases de dades en una bd-dbt sense temps vàlid de finalització

- Representació de la inserció

Veiem ara, tots els casos dels exemples d'inserció de l'assignatura de *bases de dades* del model bitemporal conceptual en el model de representació a la figura 4.3. En els casos 4.3.a i 4.3.d, l'assignatura *bases de dades* no s'ha ofert mai i es pot inserir, obtenint la representació que s'emmagatzema en el predicat *assignatures_r(S, T_{vi}, T_{vf}, T_{ti}, T_{tf})*, on *S* és el nom de l'assignatura, *T_{vi}* és el temps vàlid d'inici, *T_{vf}* és el temps vàlid de finalització, *T_{ti}* és el temps de transacció

d'inici i T_{tf} és el temps de transacció de finalització. En el cas 4.3.b l'assignatura *bases de dades* ja s'oferia en un temps vàlid diferent i ara s'afegeix la informació referent al nou curs de *bases de dades*. I finalment, en el cas 4.3.c no es pot inserir l'assignatura de *bases de dades* perquè s'està oferint amb un temps vàlid que coincideix en part amb el temps vàlid demanat.

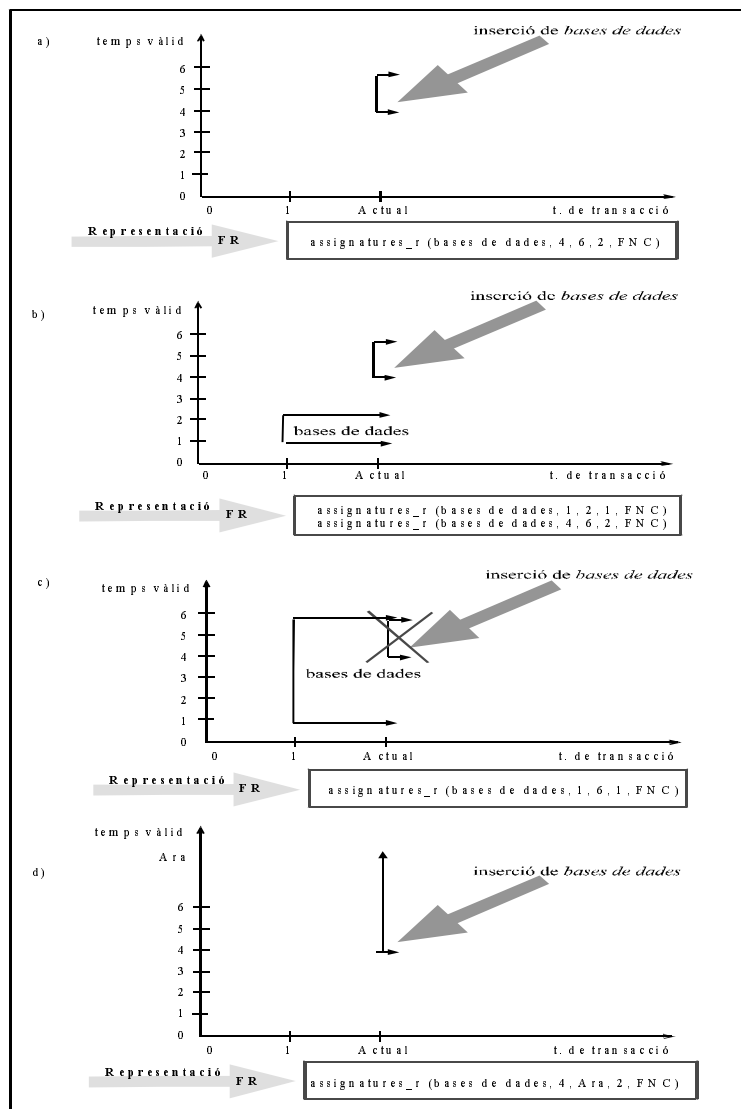


Figura 4.3. Representació de la inserció de l'assignatura bases de dades en una bd-dbt

Definim formalment els tres casos que hem distingit en el model conceptual, en el model de representació, tenint en compte que FR és el conjunt de fets del model de representació, FI és el conjunt de fets a inserir, f_r és la representació d'un fet concret, t_{vi} és el temps vàlid d'inici, t_{vf} és el temps vàlid de finalització, t_{ti} és el temps de transacció d'inici i t_{tf} és el temps de transacció de finalització del rectangle bitemporal del fet f_r i t_{actual} serà el valor que tingui el temps de transacció actual.

- $FR \cup \{f_r(a_1, \dots, a_n, t_{vi}, t_{vf}, t_{actual}, FNC)\}$
 si $\neg \exists f_r(a_1, \dots, a_n, t_{vi}, t_{vf}, t_{ti}, t_{tf}) \in FR$
- $FR \cup \{f_r(a_1, \dots, a_n, t1_{vi}, t1_{vf}, t_{actual}, FNC)\}$
 si $\exists f_r(a_1, \dots, a_n, t_{vi}, t_{vf}, t_{ti}, t_{tf}) \in FR \wedge \neg \exists (f_r(a_1, \dots, a_n, t1_{vi}, t1_{vf}, t1_{ti}, t1_{tf}) \in FI \wedge t1_{vf} \geq t_{vi} \wedge t1_{vi} \leq t_{vf} \wedge t_{tf} = FNC)$
- FR si un altre cas

4.2 Esborrats

Per denotar que es produeix un esborrat utilitzarem el símbol δ . Un esborrat sempre es produeix en temps de transacció *Actual*. Això vol dir que no caldrà donar explícitament el temps de transacció en fer qualsevol tipus d'esborrat, sinó que aquest temps es pot afegir de forma automàtica en l'instant en que s'emmagatzema a la base de dades. A més, els esborrats sempre són esborrats lògics. De manera que la informació no existeix d'ara (temps *Actual*) en endavant, però continua existint en el passat. Un esborrat lògic el que fa és tancar l'element bitemporal especificat pel temps de transacció obert fins *FNC*, de la mateixa manera que es defineix a [JSS94].

Si suposem que en l'exemple de la facultat d'informàtica volem deixar d'oferir l'assignatura de *bases de dades* que s'oferia amb temps vàlid del 4 al 6, hauriem de fer l'esborrat següent:

δ assignatures (bases de dades, 4, 6), on *Actual* seria 3.

En bd-dbts, quan fem un esborrat ens podem trobar amb les situacions que es descriuen a la figura 4.4.

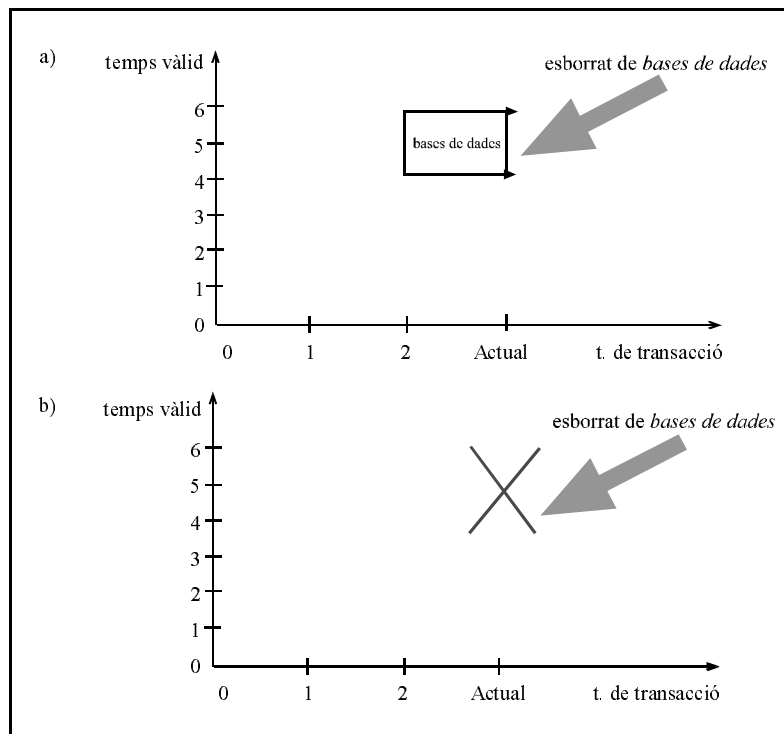


Figura 4.4. Esborrat de l'assignatura bases de dades en una bd-dbt

A la figura 4.4.a ens trobem el cas que el curs *bases de dades* s'estava oferint del 4 al 6 i, per tant, es pot esborrar. En canvi, a la figura 4.4.b veiem un cas en el qual no es pot fer l'esborrat perquè actualment no s'està oferint cap curs de *bases de dades*. També es podria haver donat el cas que haguéssim volgut esborrar el curs *bases de dades* i ja hi fos, però amb uns temps vàlids que no coincidissin amb els temps vàlids del predicat a esborrar. En aquest darrer cas, el curs tampoc no s'hagués pogut esborrar. A diferència de [JSS94], on a l'operació d'esborrar no s'especifica temps vàlid, i de manera similar a [Böh94a], per esborrar nosaltres proposem especificar el temps vàlid del fet que volem esborrar, a fi d'identificar quin fet concretament volem esborrar en cas que n'hi hagi més d'un.

A continuació, definirem formalment els dos casos que podem trobar a l'hora de fer un esborrat:

Donat l'esborrat següent d'un fet f :

$$\mathcal{E}(a_1, \dots, a_n, t_v)$$

On t_v és l'element temporal corresponent al temps vàlid, t_b és l'element bitemporal amb temps vàlid i temps de transacció, t_{ti} és el temps de transacció d'inici de l'element temporal t_t corresponent al temps de transacció i t_{actual} serà el valor que tingui el temps de transacció actual.

Es poden donar els casos següents:

- $f - \{(a_1, \dots, a_n | t_b)\} \cup \{(a_1, \dots, a_n | (\{t_{ti}, t_{actual}\} \times t_v | t_{ti} \in t_t))\}$
 si $\exists t_b ((a_1, \dots, a_n | t_b) \in f \wedge t_b = t_v \times t_t)$
- f si un altre cas
- Representació de l'esborrat

Veiem ara, tots els casos dels exemples d'esborrat de l'assignatura de *bases de dades* del model bitemporal conceptual en el model de representació a la figura 4.5.

En el cas 4.5.a l'assignatura *bases de dades* s'està oferint, es pot esborrar lògicament i s'obté la representació que s'emmagatzema en el predicat *assignatures_r*($S, T_{vi}, T_{vf}, T_{ti}, T_{tf}$), on S és el nom de l'assignatura, T_{vi} és el temps vàlid d'inici, T_{vf} és el temps vàlid de finalització, T_{ti} és el temps de transacció d'inici i T_{tf} és el temps de transacció de finalització. En el cas 4.5.b l'assignatura *bases de dades* no s'ha ofert mai i, com que no es pot esborrar, no s'obté cap nova representació.

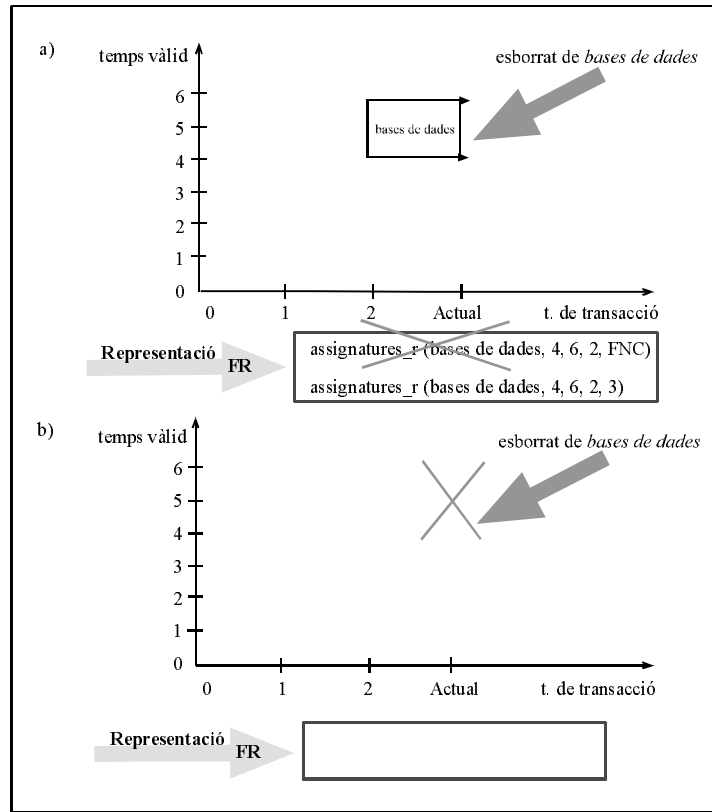


Figura 4.5. Representació de l'esborrat de l'assignatura bases de dades en una bd-dbt

Definim formalment els dos casos que hem distingit en el model conceptual, en el model de representació, tenint en compte que FR és el conjunt de fets del model de representació, f_r és la representació d'un fet concret, t_{vi} és el temps vàlid d'inici, t_{vf} és el temps vàlid de finalització, t_{ti} és el temps de transacció d'inici i t_{tf} és el temps de transacció de finalització del rectangle bitemporal del fet f_r , i t_{actual} serà el valor que tingui el temps de transacció actual.

- $FR - f_r(a_1, \dots, a_n, t_{vi}, t_{vf}, t_{ti}, t_{tf}) \cup f_r(a_1, \dots, a_n, t_{vi}, t_{vf}, t_{ti}, t_{actual})$
si $\exists(f_r(a_1, \dots, a_n, t_{vi}, t_{vf}, t_{ti}, t_{tf}) \in FR \wedge t_{tf} = FNC)$
- FR si un altre cas

4.3 Modificacions

Una modificació sempre es produeix en temps de transacció *Actual*. Això vol dir que no caldrà donar explícitament el temps de transacció en fer qualsevol tipus de modificació, sinó que aquest temps es pot afegir de forma automàtica en l'instant en què s'emmagatzema a la base de dades. De fet, de moment, considerarem que una modificació no és més que un esborrat seguit d'una inserció, tal com també es considera a [JSS94].

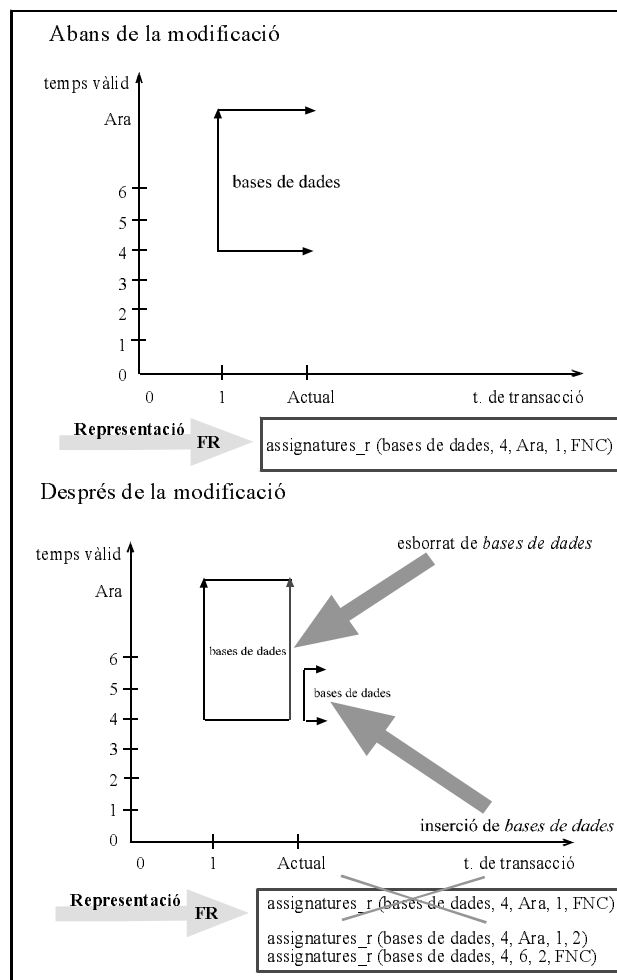


Figura 4.6. Modificació de l'assignatura bases de dades en una bd-dbt

Si suposem que en l'exemple de la facultat d'informàtica volem deixar d'oferir l'assignatura de *bases de dades*, que s'oferia des del 4 fins Ara, i que passi a oferir-se des del 4 al 6, hauríem de fer la modificació següent:

δ assignatures (bases de dades, 4, Ara), τ assignatures (bases de dades, 4, 6).

En bd-dbts, quan fem una modificació ens podem trobar amb totes les situacions que hem descrit per al cas de la inserció i l'esborrat. A la figura 4.6 podem veure com seria la modificació de l'exemple proposat per al curs de bases de dades. A la figura 4.6 també posarem la seva representació.

A [Böh94a] es proposen dues operacions per dur a terme la modificació. L'operació *replace* consisteix en un esborrat seguit d'una inserció, tal com hem explicat nosaltres la modificació. L'operació *update* s'usaria quan es volgués realitzar una modificació i obtenir temps vàlids adjacents. A continuació presentem un exemple per il·lustrar la diferència entre les operacions *replace* i *update*.

Suposem que en el predicat *assignatures*, a més del *nom*, tenim un terme per emmagatzemar el *preu* de les assignatures. En aquests moments sabem que l'assignatura de *bases de dades* s'ofereix amb un preu de 10.000 ptes. des de 4 fins a 6, i volem que a temps de transacció 2, passi a oferir-se des del 8 fins al 10 amb un preu de 20.000 ptes. Aplicant l'operació *replace* obtindríem el següent:

$assignatures_r(bases\ de\ dades, 10.000, 4, 6, 1, 2)$
 $assignatures_r(bases\ de\ dades, 20.000, 8, 10, 2, FNC)$

Aplicant l'operació *update* obtindríem el següent:

$assignatures_r(bases\ de\ dades, 10.000, 4, 6, 1, 2)$
 $assignatures_r(bases\ de\ dades, 10.000, 4, 8, 2, FNC)$
 $assignatures_r(bases\ de\ dades, 20.000, 8, 10, 2, FNC)$

Nosaltres considerem que el més semblant al que nosaltres entenem per una modificació és l'operació *replace*, que és completament coherent amb les definicions de les operacions d'inserció i esborrat. L'operació *update* és una modificació que d'alguna manera dedueix informació per a intervals de temps vàlid on aquesta informació no hi és. El problema de deduir informació inexistent

a partir de la informació que coneixem serà tractat amb més detall al capítol 7 d'aquesta tesi.

4.4 Fusió

Es pot donar el cas que després d'haver fet una inserció d'un fet ens trobem un nou rectangle bitemporal totalment *enganxat* amb el darrer rectangle bitemporal d'aquest fet amb els mateixos valors no temporals. En aquest cas hauríem de fer una fusió [BSS96] d'informació per obtenir un únic rectangle bitemporal. Partint del treball realitzat a [JSS94] i tenint en compte que el temps de transacció no pot ser actualitzat en el passat i que no es pot inserir informació que ja hi és, en el nostre cas a l'hora de fusionar ens podem trobar les situacions que es descriuen a la figura 4.7 i cap altra més.

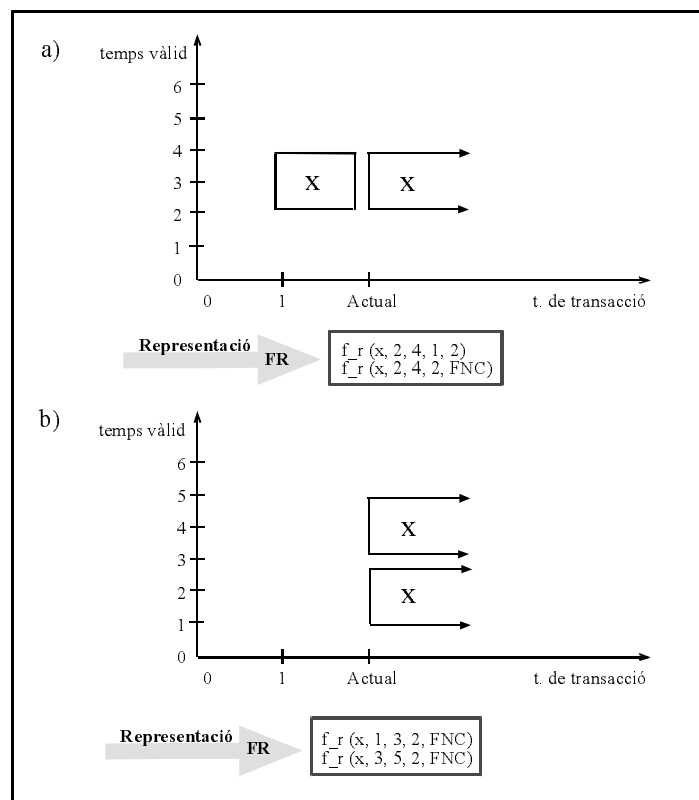


Figura 4.7. Casos en que es pot aplicar fusió

Abans d'aplicar la fusió analitzem els casos presentats a la figura 4.7. A la figura 4.7.a ens trobem que els rectangles a fusionar tenen el mateix temps vàlid i diferents temps de transacció. Aquest podria ser el cas, per exemple, si s'hagués ofert el curs de *bases de dades*, després s'hagués esborrat lògicament i finalment s'hagués arribat a la conclusió que sí que s'havia ofert. Per tant, el cas de la figura 4.7.a és sempre un cas clar a fusionar. En canvi, a la figura 4.7.b ens trobem que els rectangles a fusionar tenen el mateix temps de transacció i diferents temps vàlids. Veiem a la figura 4.8 exemples concrets d'aquest cas.

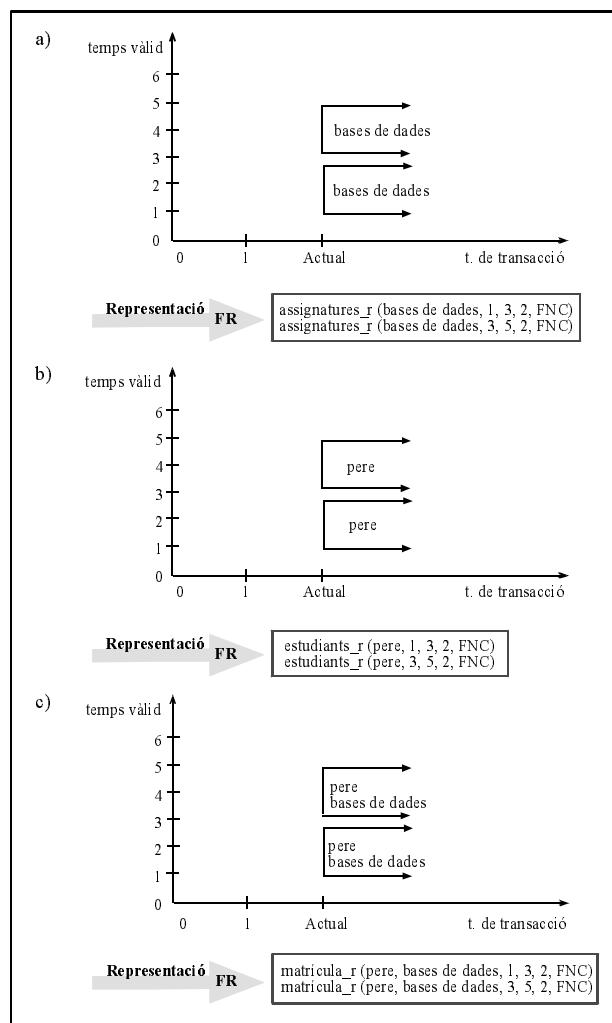


Figura 4.8. Exemples concrets del cas de la figura 4.7.b

En el cas 4.8.a tenim que l'assignatura *bases de dades* s'ha ofert en dos períodes de temps vàlid diferents. Això pot ser per dues raons: l'assignatura ha ampliat el seu període lectiu, o bé són dos cursos diferents. En el primer cas és evident que hauríem de fusionar; en canvi, en el segon cas no hauríem de fusionar perquè, encara que els dos cursos són de *bases de dades*, són cursos diferents.

En el cas 4.8.b tenim que en *pere* és estudiant en dos períodes de temps vàlid diferents. En aquest cas és evident que hauríem de fusionar perquè es tracta del mateix estudiant que, per algun motiu, s'havia pensat que havia deixat de ser estudiant.

En el cas 4.8.c tenim que l'estudiant *pere* s'ha matriculat de l'assignatura *bases de dades* en dos períodes de temps vàlid diferents. Això pot ser per dues raons: s'havia pensat que en *pere* no estava matriculat quan en realitat ho estava, o bé en *pere* repeteix l'assignatura de *bases de dades* perquè no l'ha aprovat en el curs anterior. En el primer cas és evident que hauríem de fusionar; en canvi, en el segon cas no hauríem de fusionar.

En realitat, els tres casos de la figura 4.8 no són un problema de quan fusionar, i quan no, sinó de com identificar els predicats a fusionar. En els casos on s'havia decidit no fusionar, la solució és tan senzilla com introduir un nou atribut que distingeixi entre cursos diferents d'assignatures amb el mateix nom o matrícules diferents en el cas dels estudiants repetidors. Per tant, tal com succeïa en el cas de la figura 4.7.a, també en el cas de la figura 4.7.b es podrà fusionar sempre. El problema que hem plantejat s'ha produït perquè l'exemple amb el que treballem és molt senzill, però complet per a l'objectiu d'aquesta tesi, que és la comprovació de la integritat en bd-dbts i l'estudi de les regles d'integritat temporals. No obstant això, com hem vist, no es tracta d'un model massa real, però tampoc no és necessari per als nostres propòsits.

A més del problema plantejat anteriorment d'identificar els predicats convenientment, a l'hora de treballar amb termes temporals hem de tenir en compte el tema de les dependències funcionals temporals, tractades extensament a [JSS92b] i [Wij99]. A continuació resumirem breument les conseqüències principals de treballar amb termes temporals. Suposem que a $p(K, X_1, \dots, X_n)$, K és la clau primària de p , on K s'utilitza per abreviar el conjunt de termes K_i ,

..., K_m que poden formar la clau i, per tant, K identifica de forma única els fets de p . Quan afegim els termes temporals a p , obtenim: $p(K, X_l, \dots, X_m, T_{vi}, T_{vf}, T_{ti}, T_{tf})$, ja que no es pot considerar que K identifiqui de forma única els fets de p . Aleshores necessitem utilitzar el concepte de clau temporal, que denotarem K^T , definida a [JSS92b], com una superclau temporal mínima. Un conjunt de termes és una superclau temporal si sempre identifica de forma única els fets d'un predicat. Donat el conjunt de termes d'un predicat $\bar{A} = \{Y_l, \dots, Y_m, T_{vi}, T_{vf}, T_{ti}, T_{tf}\}$, $K \subset \bar{A}$, $K^T \subseteq \bar{A}$, i $K \subset K^T$. En el capítol 7 s'estudia amb detall l'aplicació de la fusió a predicats clarament identificats amb una clau temporal.

A la figura 4.9 podem veure el resultat d'aplicar la fusió als casos de la figura 4.7. A les figures 4.7 i 4.9 hem representat gràficament la fusió d'un exemple sobre el model conceptual i el model de representació.

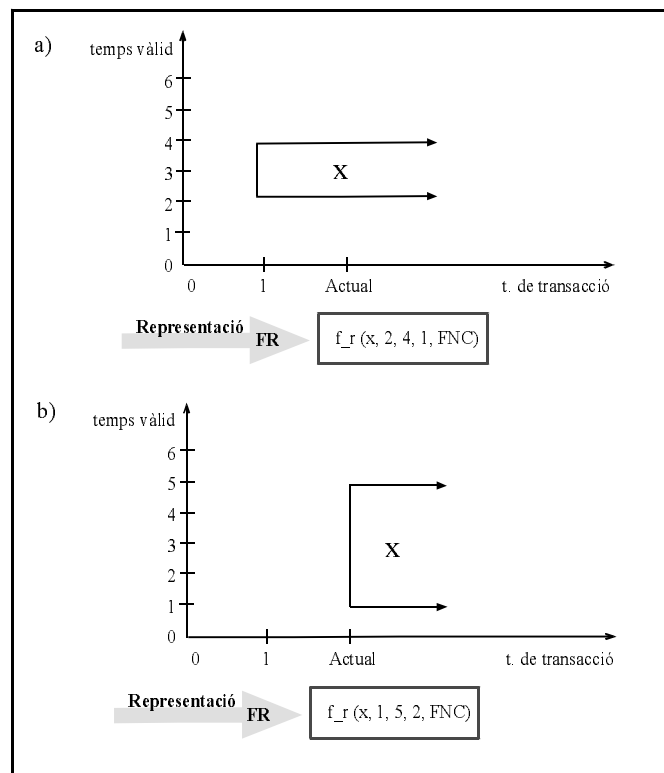


Figura 4.9. Resultat d'aplicar fusió als casos de la figura 4.7

Ara veiem, formalment, com s'aplicaria la fusió directament sobre el model de representació:

El fet f , que s'obté a partir dels dos fets $f1$ i $f2$ amb valors no temporals equivalents, on VT és l'interval de temps vàlid, TT és l'interval de temps de transacció, i A és el conjunt de termes no temporals, és:

- $f(f1[A], f1[VT], f1[T_u], f2[T_{vf}])$ si $value_equivalent(f1,f2) \wedge equals(f1[VT], f2[VT]) \wedge adjacent(f1[TT], f2[TT])$
- $f(f1[A], f1[T_{vi}], f2[T_{vf}], f1[TT])$ si $value_equivalent(f1,f2) \wedge equals(f1[TT], f2[TT]) \wedge adjacent(f1[VT], f2[VT])$

on el predicat *equals* es defineix a [All83] per expressar si dos intervals de temps són iguals o no. Els predicats *adjacent* i *value_equivalent* es defineixen a [BSS96], per expressar si dos intervals de temps són adjacents i si dos predicats tenen termes no temporals equivalents, respectivament.

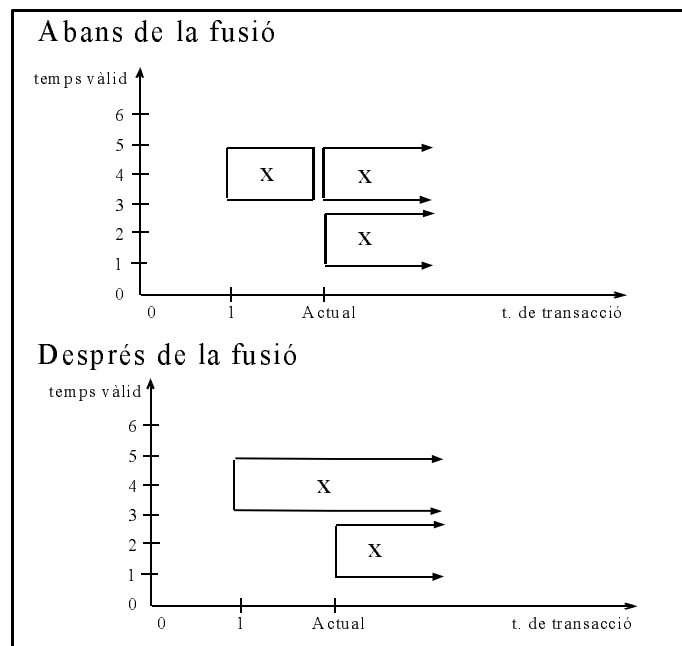


Figura 4.10. Combinació de les dues situacions on podem aplicar fusió i el resultat obtingut després de fusionar

Si trobéssim el cas de la figura 4.10, on es donessin alhora les dues situacions possibles en què podem aplicar fusió, hauríem d'escollir-ne només una, preferentment la situació de la figura 4.7 a, i aplicar-la.

4.5 Transaccions coherents

Una transacció és una unitat lògica de treball d'una base de dades [Dat00]. En el nostre cas una transacció TR consistirà en un conjunt d'insercions i d'esborrats que s'executaran com si fossin una sola operació, ja que tractarem la modificació com una inserció seguida d'un esborrat, tal com hem comentat a la secció 4.3. Les transaccions seran les que podran provocar o induir violacions de les restriccions d'integritat de la base de dades. Abans d'executar una transacció, en aquesta secció proposem una senzilla verificació de coherència que en canvi, ens estalviarà comprovacions de violacions de la integritat innecessàries. Si es detecta que una transacció no és coherent haurem de rescriure-la o rebutjar-la, i després de la seva execució comprovarem si s'ha d'aplicar o no fusió. No considerarem el cas d'execució concurrent de més d'una transacció. L'anàlisi que proposem es realitzarà dins d'una única transacció.

Una transacció diem que és coherent quan les mateixes operacions de la transacció no impedeixen entre si que alguna operació s'arribi a processar. Veiem exemples de transaccions incoherents a la figura 4.11.

Formalment, donada una transacció $TR = \{op f_1, \dots, op f_n\}$, on op són insercions o esborrats, analitzem tots els casos que poden fer que una transacció sigui incoherent i oferim la solució per a cada cas.

$$\forall op f_i, op' f_j \in TR \wedge i < j:$$

- si $op = \iota \wedge op' = \tau \wedge value_equivalent(f_i, f_j) \wedge equals(f_i[VT], f_j[VT])$ aleshores $op' f_j$ pot ser eliminada.
- si $op = \iota \wedge op' = \tau \wedge value_equivalent(f_i, f_j) \wedge intersect(f_i[VT], f_j[VT])$ aleshores $op f_i$ i $op' f_j$ s'hauran d'eliminar perquè alguna de les dues és errònia i no sabem quina és. El predicat *intersect* es defineix a [Sno00] per expressar si dos intervals de temps s'intersequen o no.

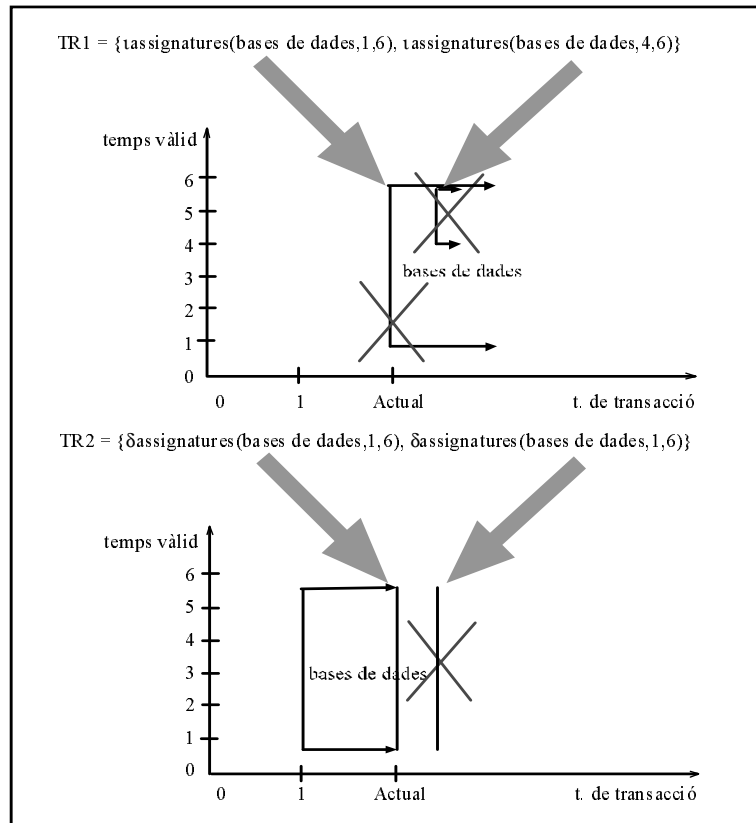


Figura 4.11. Exemples de transaccions incoherents

- si $op = \iota \wedge op' = \delta \wedge value_equivalent(f_i, f_j) \wedge equals(f_i[VT], f_j[VT])$
aleshores $op f_i$ i $op' f_j$ s'hauran d'eliminar.
- si $op = \iota \wedge op' = \delta \wedge value_equivalent(f_i, f_j) \wedge intersect(f_i[VT], f_j[VT])$
aleshores $op' f_j$ s'haurà d'eliminar.
- si $op = \delta \wedge op' = \iota \wedge value_equivalent(f_i, f_j) \wedge equals(f_i[VT], f_j[VT])$
aleshores tenim una modificació nul·la i, per tant, $op f_i$ i $op' f_j$ s'hauran d'eliminar.

- si $op = \delta \wedge op' = \iota \wedge value_equivalent(f_i, f_j) \wedge not\ equals(f_i[VT], f_j[VT])$
aleshores tenim una modificació i, per tant, la transacció és coherent.
- si $op = \delta \wedge op' = \delta \wedge value_equivalent(f_i, f_j) \wedge equals(f_i[VT], f_j[VT])$,
aleshores $op' f_j$ s'haurà d'eliminar.
- si $op = \delta \wedge op' = \delta \wedge value_equivalent(f_i, f_j) \wedge intersect(f_i[VT], f_j[VT])$,
aleshores $op f_i$ i $op' f_j$ s'hauran d'eliminar perquè alguna de les dues és errònia i no sabem quina és.

5. Restriccions d'integritat temporals

Per analitzar en profunditat el problema de la comprovació de la integritat en bases de dades deductives bitemporals, en aquest capítol ens proposem fer un estudi tan complet com sigui possible dels tipus de restriccions d'integritat temporals que es poden arribar a donar. Molts autors en moltes àrees d'investigació, però especialment en bases de dades temporals, han utilitzat el concepte de *restricció d'integritat temporal*, per exemple [GL95], [CT95b] i [CP00]. Però en cap d'aquests treballs ni en la majoria de molts d'altres s'ha utilitzat el concepte de restricció d'integritat temporal amb totes les seves possibilitats, sinó que simplement les restriccions s'han anomenat *temporals* perquè en la seva definició apareixia alguna referència temporal. S'ha de dir també que en aquests treballs l'objectiu no era l'estudi de les restriccions d'integritat temporals, sinó la utilització en algun problema o entorn concret. Fins ara, l'estudi més complet realitzat sobre restriccions d'integritat temporals en l'àrea de bases de dades temporals ha estat la classificació de Michael Böhlen [Böh94b], revisada i actualitzada recentment per Richard Snodgrass [Sno00]. Una altra classificació molt interessant de restriccions d'integritat temporals des d'un punt de vista diferent i dins de l'àrea de versions de bases de dades, molt propera a la de bases de dades temporals, és la d'Anne Doucet i altres autors [DFG+97]. Partint d'aquests dos darrers articles, anirem construint la nostra classificació de restriccions d'integritat temporals per a bd-dbts.

5.1 Restriccions d'integritat temporals

Una restricció d'integritat temporal és una fórmula tancada de primer ordre que qualsevol estat de la bd-dbt ha de satisfer. Nosaltres treballarem amb restriccions que tenen forma de denegació:

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

on els L_i són literals i totes les variables se suposen universalment quantificades davant la fórmula. Però ara, recordem que aquests L_i tenen un element bitemporal que ens permetrà expressar propietats temporals en les restriccions d'integritat.

Una restricció d'integritat temporal estarà formada per predicats bàsics, predicats derivats i predicats predefinits de comparació. Els predicats predefinits de comparació es poden aplicar a termes de predicats bàsics o derivats i a constants, però també ens serviran per expressar condicions temporals. A la figura 5.1 [Böh94a] podem veure com els tretze operadors temporals d'Allen [All83] es poden expressar en termes de punts de temps d'inici i de finalització, tal com nosaltres els utilitzarem.

predicats	gràficament	formalment
A before B		$A.fi < B.inici$
A after B		$A.inici > B.fi$
A precedes B		$A.fi = B.inici$
A follows B		$A.inici = B.fi$
A during B		$A.inici > B.inici \wedge A.fi < B.fi$
A contains B		$A.inici < B.inici \wedge A.fi > B.fi$
A equals B		$A.inici = B.inici \wedge A.fi = B.fi$
A overlaps B		$A.fi > B.inici \wedge A.inici < B.inici \wedge A.fi < B.fi$
A overlapped_by B		$A.inici > B.inici \wedge A.inici < B.fi \wedge A.fi > B.fi$
A starts B		$A.inici = B.inici \wedge A.fi < B.fi$
A started_by B		$A.inici = B.inici \wedge A.fi > B.fi$
A ends B		$A.inici > B.inici \wedge A.fi = B.fi$
A ended_by B		$A.inici < B.inici \wedge A.fi = B.fi$

Figura 5.1. Els tretze predicats temporals d'Allen expressats amb punts de temps d'inici i de finalització

Evidentment, els tretze predicats temporals d'Allen es poden combinar entre si per expressar condicions més complexes. Per exemple, de vegades ens interessarà el predicat *A during B*, però no ens importarà que sigui *A starts B* o *A ends B* o *A equals B*. Si utilitzem els predicats temporals d'Allen hauríem d'utilitzar *during*, *starts*, *ends* i *equals* per poder expressar el que volem. Si seguíssim les indicacions de la figura 5.1, formalment amb punts de temps hauríem de posar: $(A.inici > B.inici \wedge A.fi < B.fi)$ o $(A.inici = B.inici \wedge A.fi < B.fi)$ o $(A.inici > B.inici \wedge A.fi = B.fi)$ o $(A.inici = B.inici \wedge A.fi = B.fi)$. Però en aquest cas, nosaltres posaríem directament $A.inici \geq B.inici \wedge A.fi \leq B.fi$.

5.2 Taxonomia de Michael Böhlen

Les restriccions d'integritat, seguint la taxonomia presentada a [Böh94b], es poden classificar com detallem a continuació:

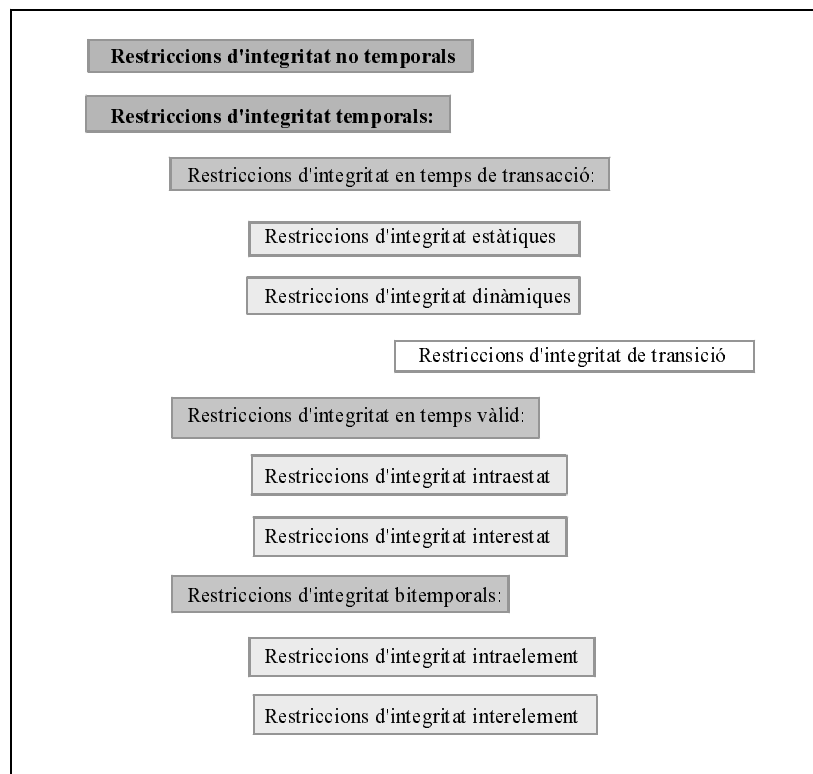


Figura 5.2. Taxonomia de Michael Böhlen

A la primera classificació d'aquesta taxonomia podem dividir les restriccions d'integritat en temporals i no temporals. Les restriccions d'integritat no temporals són restriccions d'integritat convencionals que no treballen ni amb temps vàlid ni amb temps de transacció. Les restriccions d'integritat temporals poden alhora dividir-se en tres grups, segons si les propietats que han de fer complir impliquen temps vàlid, temps de transacció o tots dos temps (bitemporals). Aquestes tres classificacions poden alhora descompondre's en una nova subdivisió segons si fan referència a un únic temps o a diversos.

- Restriccions d'integritat en temps de transacció

Les restriccions d'integritat en temps de transacció estan definides per a bases de dades en temps de transacció. Aquestes es divideixen en estàtiques i dinàmiques. Per obtenir les files en temps vàlid, s'aplicaria l'operador fotogràfic en temps vàlid i per obtenir les files en temps de transacció, l'operador fotogràfic en temps de transacció [JCE+93]. Un estat de la base de dades [JSS92a] es correspon a la fotografia de la base de dades obtinguda de l'aplicació dels operadors fotogràfics. Les restriccions d'integritat estàtiques són restriccions que han de complir-se en cada estat de la base de dades en temps de transacció, mentre que les dinàmiques fan referència i restringeixen diferents estats de la base de dades en temps de transacció. Les restriccions d'integritat de transició són un subconjunt conegut de les restriccions d'integritat dinàmiques, que imposen restriccions entre dos estats consecutius de la base de dades.

- Restriccions d'integritat en temps vàlid

Les restriccions d'integritat en temps vàlid estan definides per a bases de dades en temps vàlid. Aquestes es divideixen en intraestat i interestat. Les restriccions d'integritat intraestat són restriccions que garanteixen la consistència en una fotografia de la base de dades en temps vàlid, mentre que les interestat fan referència i restringeixen diferents fotografies de la base de dades en temps vàlid.

- Restriccions d'integritat bitemporals

Les restriccions d'integritat bitemporals estan definides per a bases de dades bitemporals. Aquestes es divideixen en intraelement i interelement. Les restriccions d'integritat intraelement són restriccions que garanteixen la consistència en tots i cadascun dels estats de la base de dades bitemporal, mentre

que les interelemen fan referència i restringeixen diferents estats de la base de dades bitemporal.

5.2 Taxonomia d'Anne Doucet *et al.*

Per estudiar les possibles motivacions semàntiques de les restriccions d'integritat hem escollit seguir les pautes introduïdes a la classificació d'Anne Doucet *et al.* [DFG+97]:

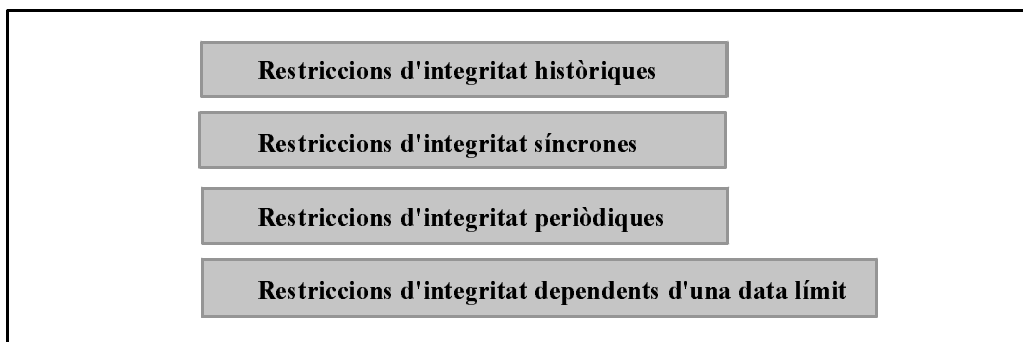


Figura 5.3. Taxonomia d'Anne Doucet *et al.*

- Restriccions d'integritat històriques

Aquest tipus de restriccions d'integritat només poden passar si certes condicions es compleixen. Per exemple: un curs ha de costar més de *10.000* ptes. per poder-se oferir.

Aquestes restriccions nosaltres les anomenarem *restriccions d'integritat per condicions*, perquè la paraula *històric* en bases de dades temporals porta implícites connotacions de temps vàlid. Les condicions poden ser combinacions de variables dels predicats de la restricció d'integritat amb constants mitjançant operadors de comparació, o bé combinacions de variables temporals i/o constants amb també operadors de comparació.

- Restriccions d'integritat síncrones

Les restriccions d'integritat síncrones fan que les diferents condicions de la restricció es compleixin al mateix temps sempre. Per exemple: si un estudiant es matricula d'un curs en un determinat temps vàlid, aquest curs ha d'existir amb el mateix temps vàlid amb el qual l'estudiant es matricula.

Dins de les restriccions d'integritat síncrones nosaltres afegirem una nova classificació: si són *síncrones en temps vàlid* o *en temps de transacció*. Notem que el sincronisme en temps vàlid fa que els temps vàlids siguin iguals i que els temps de transacció siguin diferents; mentre que el sincronisme en temps de transacció fa que siguin iguals els temps de transacció i diferents els temps vàlids. Un sincronisme de temps vàlid i temps de transacció alhora farà que tant els temps vàlids com els temps de transacció siguin iguals en tots els predicats de la restricció d'integritat.

- Restriccions d'integritat periòdiques

Aquest tipus de restriccions d'integritat només poden passar si les condicions a complir són periòdiques. S'han de complir cada setmana, cada mes, cada any, ... Per exemple: el preu de les assignatures s'incrementarà cada any.

Dins de les restriccions d'integritat periòdiques nosaltres afegirem una nova classificació: si són *periòdiques en temps vàlid* o *en temps de transacció*. La periodicitat no és més que un tipus de condició en què es poden utilitzar les variables temporals *Actual*, *Ara* i *FNC* per definir cada quin cert temps s'ha de complir la restricció d'integritat.

- Restriccions d'integritat dependents d'una data límit

Aquestes restriccions d'integritat fan que les condicions a complir tinguin una data límit. Per exemple: un estudiant s'ha de matricular durant els 10 dies següents a l'oferta de l'assignatura; si no, ja no es podrà matricular aquest curs.

Dins de les restriccions d'integritat dependents d'una data límit nosaltres afegirem una nova classificació: si són *dependents d'una data límit en temps*

vàlid o *en temps de transacció*. Una data límit és un tipus de condició en què s'opera una constant amb una variable temporal.

Aquests quatre criteris d'Anne Doucet *et al.* [DFG+97], convertits en set en el nostre cas, combinats amb la taxonomia de Michael Böhlen [Böh94b] ens ajudaran a entendre millor com poden ser les restriccions d'integritat temporals en bd-dbts.

5.4 Taxonomia de restriccions d'integritat temporals en bases de dades deductives bitemporals

Si ens centrem en com poden ser les restriccions d'integritat temporals en una bd-dbt, veiem que:

- Les restriccions d'integritat estan definides en un temps vàlid determinat i en un temps de transacció determinat. En aquest cas tenim restriccions d'integritat actuals (primer grup). L'àmbit d'una restricció d'integritat actual es pot veure a la figura 5.4 notat amb el símbol ■.
- Les restriccions d'integritat estan definides en un temps vàlid determinat i l'àmbit de la restricció ve donat per diferents punts de temps de transacció. En aquest cas tenim restriccions d'integritat en temps de transacció (segon grup). L'àmbit d'una restricció d'integritat en temps de transacció es pot veure a la figura 5.4.a.
- Les restriccions d'integritat estan definides en un temps de transacció determinat i l'àmbit de la restricció ve donat per diferents punts de temps vàlid. En aquest cas tenim restriccions d'integritat en temps vàlid (tercer grup). L'àmbit d'una restricció d'integritat en temps vàlid es pot veure a la figura 5.4.b.
- Les restriccions d'integritat estan definides per diferents punts de temps vàlid i diferents punts de temps de transacció. En aquest cas tenim restriccions d'integritat bitemporals (quart grup). L'àmbit d'una restricció d'integritat bitemporal es pot veure a la figura 5.4.c.

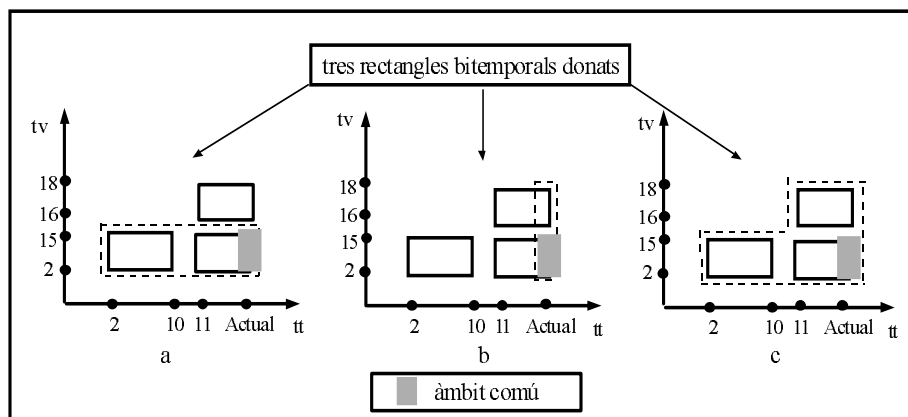


Figura 5.4. Les línies discontinües indiquen l'àmbit de les diferents restriccions d'integritat temporals per a un hipotètic cas concret. El símbol ■ mostra l'àmbit comú de tres tipus de restriccions d'integritat temporals, que concretament serà l'àmbit de les restriccions d'integritat actuals

En bd-dbts podem trobar qualsevol dels tipus de restriccions d'integritat temporals de la taxonomia de Michael Böhlen definides amb la semàntica de la classificació d'Anne Doucet *et al.* Si analitzem la taxonomia de restriccions d'integritat de Michael Böhlen, també utilitzada per Richard Snodgrass, que introdueix una nova nomenclatura [Sno00], podem trobar algunes similituds o redundàncies entre diferents tipus de restriccions, que podríem situar en un únic grup. El cas més clar és el de les restriccions estàtiques, intraestat i intraelement, que podrien formar un únic grup en el nostre cas.

A la figura 5.5, es poden veure els quatre grups de la nostra classificació, resultat de la combinació de les sis categories de la taxonomia de Michael Böhlen amb els nostres set criteris, obtinguts a partir de la taxonomia d'Anne Doucet *et al.* Les línies que connecten el tipus de restriccions d'integritat de Michael Böhlen i els nostres tipus indiquen quin mena de connexió hi ha entre els dos tipus i que explicarem amb detall a les properes seccions. El símbol √ indica quin dels set criteris compleix cada grup. En el cas del primer grup, les restriccions d'integritat poden ser síncrones en temps vàlid i en temps de transacció, però només si ho són al mateix temps. En el darrer grup, les restriccions d'integritat poden ser síncrones en temps vàlid només si no ho són en temps de transacció, i en temps de transacció si no ho són en temps vàlid.

Criteris	Restriccions d'integritat de Michael Böhlen				
	estàtiques	intraestat	intraelement	dinàmiques	
			transició	interestat	
				interelement	
	Restriccions d'integritat	Primer grup actuals	Segon grup en t. de transacció	Terçer grup en temps vàlid	Quart grup bitemporals
Restriccions d'integritat per condicions	√	√	√	√	√
Restriccions d'integritat síncrones en t. vàlid	}		√		(√)
Restriccions d'integritat síncrones en t. transacció		√		√	(√)
Restriccions d'integritat periòdiques en t. vàlid				√	√
Restriccions d'integritat periòdiques en t. trans			√		√
Restriccions d'integritat d.d.l. en t. vàlid				√	√
Restriccions d'integritat d.d.l. en t. transacció.			√		√

Figura 5.5. Classificació de restriccions d'integritat temporals en bd-dbts

5.4.1 Exemple

A continuació presentem els predicats bàsics, les regles de deducció i les restriccions d'integritat temporals de l'exemple de la facultat d'informàtica, que ens ajudarà a il·lustrar el comportament de les diferents restriccions d'integritat temporals.

- Predicats bàsics

$assignatures(C, P, T_{vi}, T_{vf}, T_{ti}, T_{tf})$ expressa que "C és un curs amb un preu P, ofert per la universitat des de temps vàlid T_{vi} fins a temps vàlid T_{vf} i des de temps de transacció T_{ti} fins a temps de transacció T_{tf} ".

$estudiants(S, T_{vi}, T_{vf}, T_{ti}, T_{tf})$ expressa que "S és un estudiant de la universitat des de temps vàlid T_{vi} fins a temps vàlid T_{vf} i des de temps de transacció T_{ti} fins a temps de transacció T_{tf} ".

$matrícula(S, C, T_{vi}, T_{vf}, T_{ti}, T_{tf})$ expressa que "un estudiant S està matriculat en el curs C des de temps vàlid T_{vi} fins a temps vàlid T_{vf} i des de temps de transacció T_{ti} fins a temps de transacció T_{tf} ".

- Regles de deducció

$$\text{curs}(C, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \leftarrow \text{assignatures}(C, P, T_{vi}, T_{vf}, T_{ti}, T_{tf})$$

El predicat derivat $\text{curs}(C, T_{vi}, T_{vf}, T_{ti}, T_{tf})$ expressa que "un curs C s'estarà oferint des de temps vàlid T_{vi} fins a temps vàlid T_{vf} i des de temps de transacció T_{ti} fins a temps de transacció T_{tf} , si entre els mateixos períodes de temps vàlid i de temps de transacció s'ofereix l'assignatura del curs C ".

$$\begin{aligned} \text{més_d'un_estudiant}(C, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \leftarrow & \text{matricula}(S1, C, T_{vi}, T_{vf}, T1_{ti}, T1_{tf}) \wedge \\ & \text{matricula}(S2, C, T_{vi}, T_{vf}, T2_{ti}, T2_{tf}) \wedge S1 \neq S2 \wedge \\ & \text{intersec}(T_{ti}, T_{tf}, T1_{ti}, T1_{tf}, T2_{ti}, T2_{tf}) \end{aligned}$$

El predicat derivat $\text{més_d'un_estudiant}(C, T_{vi}, T_{vf}, T_{ti}, T_{tf})$ expressa que "en un curs C hi haurà més d'un estudiant si, com a mínim, hi ha matriculats dos estudiants diferents des de temps vàlid T_{vi} fins a temps vàlid T_{vf} i des de temps de transacció T_{ti} fins a temps de transacció T_{tf} , on el període de temps de transacció serà la intersecció entre el període de temps de transacció de la matrícula de l'estudiant $S1$ i el de l'estudiant $S2$ ".

En aquesta regla de deducció utilitzem la intersecció entre dos intervals de temps. Podem definir la intersecció amb les regles de deducció següents:

$$\text{intersec}(T1_{ti}, T1_{tf}, T1_{ti}, T1_{tf}, T2_{ti}, T2_{tf}) \leftarrow T1_{ti} \geq T2_{ti} \wedge T1_{tf} \leq T2_{tf}$$

$$\text{intersec}(T2_{ti}, T2_{tf}, T1_{ti}, T1_{tf}, T2_{ti}, T2_{tf}) \leftarrow T1_{ti} < T2_{ti} \wedge T1_{tf} > T2_{tf}$$

$$\text{intersec}(T2_{ti}, T1_{tf}, T1_{ti}, T1_{tf}, T2_{ti}, T2_{tf}) \leftarrow T1_{ti} < T2_{ti} \wedge T1_{tf} < T2_{tf} \wedge T1_{ti} \geq T2_{ti}$$

$$\text{intersec}(T1_{ti}, T2_{tf}, T1_{ti}, T1_{tf}, T2_{ti}, T2_{tf}) \leftarrow T1_{ti} > T2_{ti} \wedge T1_{tf} > T2_{tf} \wedge T1_{ti} \leq T2_{tf}$$

A la figura 5.6 podem veure la representació gràfica corresponent a les quatre regles que defineixen la intersecció.

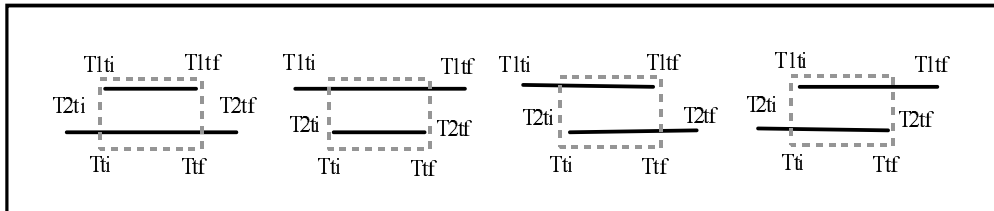


Figura 5.6. Representació gràfica dels casos en que es pot donar intersecció entre intervals de punts de temps

- Restriccions d'integritat temporals

$$ic1 \leftarrow assignatures(C,P,T_{vi},T_{vf},T_{ti},T_{tf}) \wedge P < 10000$$

Ic1 fa complir la propietat següent: "perquè un curs C s'ofereixi ha de costar, com a mínim, 10.000 pessetes".

$$ic2 \leftarrow estudiants(S,T_{vi},T_{vf},T_{ti},T_{tf}) \wedge matricula(S,C,T_{vi},T_{vf},T_{ti},T_{tf}) \wedge T_{ti} \leq T_{ti} \wedge T_{tf} < T_{tf}$$

Ic2 fa complir la propietat que: "un estudiant S només es pot matricular en un curs C si aquest estudiant S consta com a estudiant durant el curs C ".

$$ic3 \leftarrow matricula(S,'bases de dades',T_{vi},T_{vf},T_{ti},T_{tf}) \wedge matricula(S,'enginyeria del software',T_{vi},T_{vf},T_{ti},T_{tf}) \wedge T_{vi} \leq T_{vi}$$

Ic3 fa complir la propietat que: "quan un estudiant S es matricula en el curs *bases de dades*, si aquest estudiant S ja fa també, en el mateix temps de transacció, la matrícula del curs *enginyeria del software*, ha de fer aquesta segona matrícula en un temps vàlid anterior a l'inici de la matrícula del curs *bases de dades*"

$$ic4 \leftarrow assignatures(C,P,T_{vi},T_{vf},T_{ti},T_{tf}) \wedge assignatures(C,P1,T_{vi},T_{vf},T_{ti},T_{tf}) \wedge P1 < P \wedge T_{vf} < T_{vi} \wedge T_{ti} < T_{ti} - 10$$

Ic4 fa complir la propietat que: "si un curs C s'ha ofert amb un preu determinat, només durant els propers 10 dies aquest curs es podrà oferir amb un temps vàlid posterior a l'anterior i amb un preu menor que el que

haviem donat inicialment. Després d'aquests 10 dies, aquest curs no es podrà oferir amb un preu menor que el que originàriament havíem donat".

$$ic5 \leftarrow matricula(S, C, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg curs(C, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge T_{ti} \leq T_{tf} \wedge T_{ti} < T_{tf}$$

Ic5 fa complir la propietat que: "un estudiant S només es pot matricular en un curs C si aquest curs C s'ofereix".

$$ic6 \leftarrow més_d'un_estudiant(C, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg més_d'un_estudiant(C, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge T_{ti} < T_{tf}$$

Ic6 fa complir la propietat que: "si un curs C té més d'un estudiant en algun moment no pot ser que en el futur aquest curs C no tingui més d'un estudiant".

5.4.2. Referències a temps de transacció passats

Si una restricció d'integritat es defineix fent referència a un temps de transacció futur, partint d'un temps de transacció passat:

$$ic \leftarrow p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge q(Y, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge T_{tf} \geq T_{ti} \quad o$$

$$ic \leftarrow p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge q(Y, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge T_{tf} > T_{ti}$$

sempre podem veure-la en forma de referència a temps de transacció passat, si plantegem la mateixa restricció d'integritat en forma de passat:

$$ic \leftarrow p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge q(Y, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge T_{tf} \leq T_{ti} \quad o$$

$$ic \leftarrow p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge q(Y, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge T_{tf} < T_{ti}, \text{ respectivament.}$$

El propòsit d'aquesta transformació és assegurar que totes les referències de temps que trobarem a la restricció d'integritat estaran en forma de temps passats. Podem veure gràficament la transformació de futur en passat a la figura 5.7.

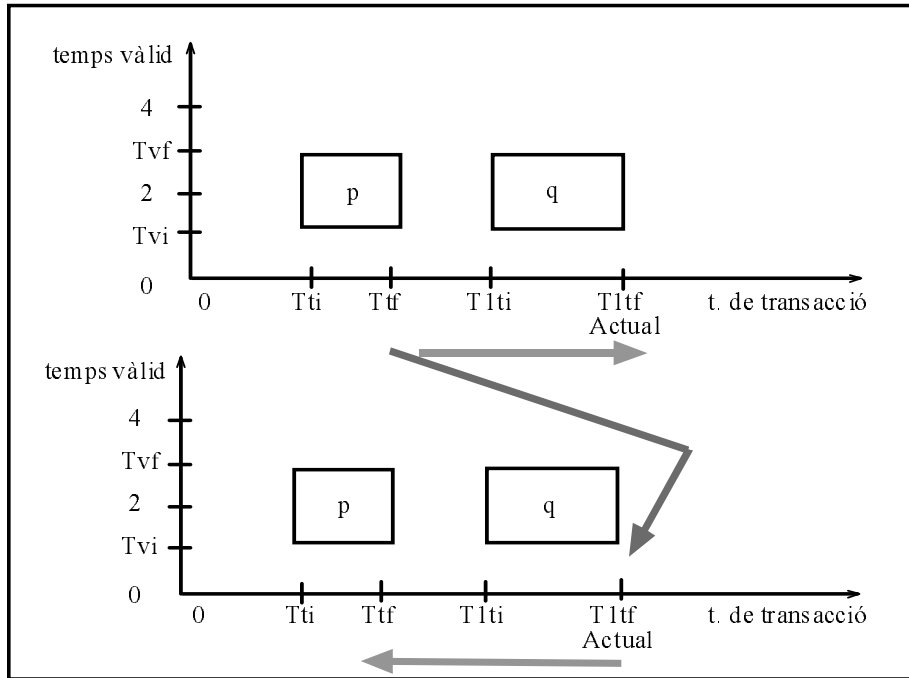


Figura 5.7. Transformació de restriccions d'integritat futures en passades

5.4.3. Anàlisi de les condicions de les restriccions d'integritat temporals

Si hem fet la transformació de restriccions futures en passades anterior i, per tant, trobem una restricció d'integritat que conté una referència menor o igual a un temps de transacció passat sense periodicitats, ni dates límit, com per exemple *ica* per al cas de les restriccions en temps de transacció (i veiem aquí també el cas d'*icb* per a les restriccions d'integritat bitemporals, on les condicions poden fer referència a temp vàlid):

$$ica \leftarrow p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge q(Y, T_{vi}, T_{vf}, T_{li}, T_{lfi}) \wedge T_{ti} \leq T_{li} \wedge T_{lfi} \leq T_{tf} \wedge cond$$

$$icb \leftarrow p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge q(Y, T_{li}, T_{lfi}, T_{ti}, T_{lfi}) \wedge T_{ti} \leq T_{li} \wedge T_{lfi} \leq T_{tf} \wedge cond$$

sempre podrem rescriure-les, i obtindrem:

$ica \leftarrow ica_i \quad i = 1, 2, 3, 4$

$ica_1 \leftarrow p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge q(Y, T_{vi}, T_{vf}, TI_{ti}, TI_{tf}) \wedge T_{ti} < TI_{ti} \wedge TI_{tf} < T_{tf} \wedge cond$

$ica_2 \leftarrow p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge q(Y, T_{vi}, T_{vf}, TI_{ti}, TI_{tf}) \wedge T_{ti} < TI_{ti} \wedge TI_{tf} = T_{tf} \wedge cond$

$ica_3 \leftarrow p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge q(Y, T_{vi}, T_{vf}, TI_{ti}, TI_{tf}) \wedge T_{ti} = TI_{ti} \wedge TI_{tf} < T_{tf} \wedge cond$

$ica_4 \leftarrow p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge q(Y, T_{vi}, T_{vf}, TI_{ti}, TI_{tf}) \wedge T_{ti} = TI_{ti} \wedge TI_{tf} = T_{tf} \wedge cond$

$icb \leftarrow icb_i \quad i = 1, 2, 3, 4$

$icb_1 \leftarrow p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge q(Y, TI_{vi}, TI_{vf}, TI_{ti}, TI_{tf}) \wedge T_{ti} < TI_{ti} \wedge TI_{tf} < T_{tf} \wedge cond$

$icb_2 \leftarrow p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge q(Y, TI_{vi}, TI_{vf}, TI_{ti}, TI_{tf}) \wedge T_{ti} < TI_{ti} \wedge TI_{tf} = T_{tf} \wedge cond$

$icb_3 \leftarrow p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge q(Y, TI_{vi}, TI_{vf}, TI_{ti}, TI_{tf}) \wedge T_{ti} = TI_{ti} \wedge TI_{tf} < T_{tf} \wedge cond$

$icb_4 \leftarrow p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge q(Y, TI_{vi}, TI_{vf}, TI_{ti}, TI_{tf}) \wedge T_{ti} = TI_{ti} \wedge TI_{tf} = T_{tf} \wedge cond$

D'aquesta manera les restriccions d'integritat en temps de transacció com ica_1 , ica_2 i ica_3 seran les que pertanyeran al segon grup.

Les restriccions d'integritat en temps de transacció com ica_4 passaran a formar part del primer grup. I només s'hauran de comprovar en temps de transacció $Actual = T_{ti} = TI_{ti}$.

Les restriccions d'integritat bitemporals com icb_1 , icb_2 i icb_3 passaran a formar part del quart grup.

I finalment, les restriccions d'integritat bitemporals com icb_4 pertanyeran al tercer grup, ja que hem eliminat les referències temporals al passat respecte al temps de transacció i les hem deixat a icb_1 , icb_2 i icb_3 . Ara només s'hauran de comprovar en temps de transacció $Actual = T_{ti} = TI_{ti}$.

En general, per a cada restricció d'integritat amb k condicions amb temps de transacció menor o igual, classificarem $2^k - 1$ restriccions d'integritat en un grup i la número 2^k en el grup anterior.

La idea de la descomposició és que, de les restriccions d'integritat que ens generen més regles de transició, les de temps de transacció i les bitemporals, sempre que sigui possible, generem la seva part actual o en temps vàlid, que només produeix una única regla, de manera que sigui la primera de comprovar-

se, ja que no requereix cap accés a la base de dades.

Per exemple, *ic5* és una restricció d'integritat en temps de transacció:

$$ic5 \leftarrow matrícula(S,C,T_{vi},T_{vf},T_{ti},T_{tf}) \wedge \neg curs(C,T_{vi},T_{vf},TI_{ti},TI_{tf}) \wedge TI_{tf} \leq T_{tf} \wedge T_{ti} < TI_{tf}$$

Tindrà una part en el primer grup, ja que, com que no demana res sobre el temps de transacció d'inici, podria donar-se el cas que $T_{ti} = TI_{ti}$.

$$ic5 \leftarrow ic5_i \quad i = 1, 2, 3$$

$$ic5_3 \leftarrow matrícula(S,C,T_{vi},T_{vf},T_{ti},T_{tf}) \wedge \neg curs(C,T_{vi},T_{vf},TI_{ti},TI_{tf}) \wedge T_{ti} = TI_{ti} \wedge T_{tf} = TI_{tf}$$

i la resta, que en aquest cas només implica dues noves regles en lloc de tres, en el segon grup:

$$ic5_1 \leftarrow matrícula(S,C,T_{vi},T_{vf},T_{ti},T_{tf}) \wedge \neg curs(C,T_{vi},T_{vf},TI_{ti},TI_{tf}) \wedge TI_{tf} < T_{tf} \wedge T_{ti} < TI_{tf}$$

$$ic5_2 \leftarrow matrícula(S,C,T_{vi},T_{vf},T_{ti},T_{tf}) \wedge \neg curs(C,T_{vi},T_{vf},TI_{ti},TI_{tf}) \wedge TI_{tf} = T_{tf} \wedge T_{ti} < TI_{tf}$$

La restricció *ic5* seria més real si s'hagués plantejat com una restricció d'integritat bitemporal, ja que no sembla gaire lògic exigir que els temps vàlids de *matrícula* i $\neg curs$ siguin iguals. Aquesta simplificació s'ha produït per mostrar un exemple més curt, ja que l'única diferència que hi hauria en aquest nou cas seria que hauríem d'afegir més condicions.

5.4.4. Primer grup: restriccions d'integritat actuals

Les restriccions d'integritat estàtiques, intraestat, intraelement i les dinàmiques o de transició, excloses del segon grup, en la notació de la taxonomia de Michael Böhlen formen el primer grup. Aquest grup es caracteritza perquè la restricció d'integritat fa referència, per a un temps vàlid donat, a un únic punt de temps de transacció, que concretament és el punt de temps *Actual*. Si haguéssim de donar un nom a aquest grup el podríem anomenar *restriccions d'integritat actuals*. Una

restricció d'integritat actual només pot ser de tipus per condicions o amb un sincronisme en temps vàlid i temps de transacció alhora, que faran que els temps de tots els predicats de la restricció d'integritat siguin iguals.

Formalment una restricció d'integritat actual seria del tipus:

$$ic \leftarrow p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) [\wedge q(Y, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \dots] \wedge \{condicions / sincronisme \text{ temps vàlid i sincronisme temps transacció}\}$$

On els claudàtors $[\]$ indiquen que podem posar, opcionalment, tants predicats com vulguem i les claus $\{ \}$ indiquen que hem d'escollir entre una o diverses de les alternatives que hi trobem.

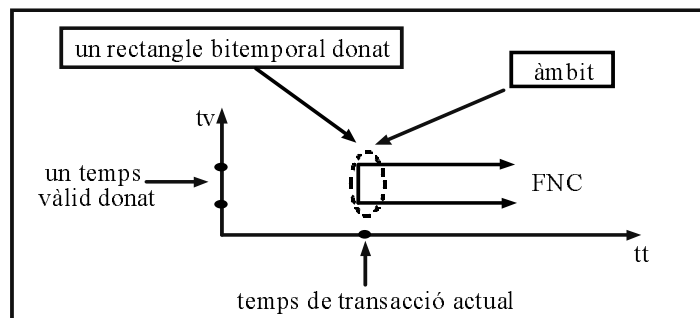


Figura 5.8. Àmbit de les restriccions d'integritat actuals

És interessant fer la distinció entre les restriccions d'integritat estàtiques, intraestat i inraelement de Michael Böhlen, ja que si treballéssim amb bases de dades en temps de transacció només tindríem restriccions d'integritat estàtiques, dinàmiques o de transició. Si les bases de dades utilitzades fossin en temps vàlid, aleshores les restriccions podrien ser intraestat o interestat. Només en el cas de bases de dades bitemporals es poden donar tots els tipus de restriccions d'integritat temporals, i és precisament en aquest cas, el nostre, quan coincideixen les restriccions estàtiques, intraestat i inraelement. L'àmbit, doncs, d'aquests tres tipus de restriccions d'integritat actuals el podem veure a la figura 5.8, que es correspon exactament amb l'àmbit comú que havíem mostrat a la figura 5.4.

Analitzem una restricció d'integritat actual, com és el cas per exemple d'*icI*:

$$icI \leftarrow assignatures(C,P,T_{vi},T_{vf},T_{ti},T_{tf}) \wedge P < 10000$$

IcI s'haurà de comprovar en temps de transacció *Actual* i haurà de complir-se des de temps vàlid T_{vi} fins a temps vàlid T_{vf} .

Una restricció d'integritat actual, com ja hem comentat abans, només pot ser de tipus per condicions o amb un sincronisme de temps vàlid i temps de transacció alhora, ja que si fos del tipus de restriccions síncrones en temps vàlid, periòdiques en temps de transacció o dependents d'una data límit en temps de transacció, hauria de fer alguna referència al temps de transacció passat i aleshores deixaria de ser una restricció d'integritat actual per passar a ser en temps de transacció. I si fos del tipus de restriccions síncrones en temps de transacció, periòdiques en temps vàlid o dependents d'una data límit en temps vàlid, hauria de fer alguna referència al temps vàlid passat i aleshores deixaria de ser una restricció d'integritat actual per passar a ser en temps vàlid. Per tant, *icI* ha de ser una restricció d'integritat actual per condicions o amb sincronisme total i, efectivament, ho és perquè en aquest cas hi ha una condició que s'ha de fer complir.

5.4.5. Segon grup: restriccions d'integritat en temps de transacció

Algunes de les restriccions d'integritat dinàmiques o de transició de la taxonomia de Michael Böhlen formen el segon grup. Aquest grup es caracteritza perquè la restricció d'integritat fa referència, per a un temps vàlid donat, a diferents punts de temps de transacció, un dels quals és sempre el punt de temps *Actual*. Si haguéssim de donar un nom a aquest grup el podríem anomenar *restriccions d'integritat en temps de transacció*. Una restricció d'integritat en temps de transacció pot ser de tipus per condicions, síncrona en temps vàlid, periòdica en temps de transacció i dependent d'una data límit en temps de transacció.

Formalment una restricció d'integritat en temps de transacció seria del tipus:

$$ic \leftarrow p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) [\wedge q(Y, T_{vi}, T_{vf}, T_{2ti}, T_{2tf}) \dots] \wedge \{ \text{condicions/sincronisme} \\ \text{temps valid/periodicitat temps transacció/data limit temps transacció} \}$$

punts de temps de transacció on s'han de fer complir, un dels quals és sempre el temps de transacció *Actual*. A la figura 5.10 podem veure l'àmbit més genèric possible de les restriccions d'integritat de transició.

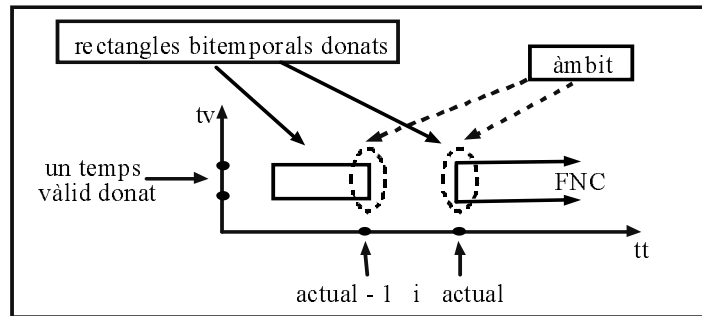


Figura 5.10. Àmbit de les restriccions d'integritat de transició de la taxonomia de Michael Böhlen

5.4.6. Tercer grup: restriccions d'integritat en temps vàlid

Les restriccions d'integritat interestat i les interelement, excloses del quart grup, en la notació de la taxonomia de Michael Böhlen, formen el tercer grup. Aquest grup es caracteritza per tenir el mateix punt de temps de transacció, sempre l'*Actual* i diferents punts de temps vàlid. Si haguéssim de donar un nom a aquest grup el podríem anomenar *restriccions d'integritat en temps vàlid*. Una restricció d'integritat de temps vàlid pot ser de tipus per condicions, síncrona en temps de transacció, periòdica en temps vàlid i dependent d'una data límit en temps vàlid. La particularitat més interessant d'aquest grup és la transformació de les restriccions d'integritat bitemporals, excloses del quart grup, en restriccions d'integritat en temps vàlid.

Formalment una restricció d'integritat en temps vàlid seria del tipus:

$$ic \leftarrow p(X, T1_{vi}, T1_{vf}, T_{ti}, T_{tf}) [\wedge q(Y, T2_{vi}, T2_{vf}, T_{ti}, T_{tf}) \dots] \wedge \{ \text{condicions} / \text{sincronisme temps transacció} / \text{periodicitat temps vàlid} / \text{data límit temps vàlid} \}$$

On els claudàtors $[]$ indiquen que podem posar, opcionalment, tants predicats com vulguem i les claus $\{ \}$ indiquen que hem d'escollir entre una o diverses de les alternatives que hi trobem.

$Ic3$ seria una restricció d'integritat en temps vàlid:

$$ic3 \leftarrow \text{matricula}(S, 'bases\ de\ dades', T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \\ \text{matricula}(S, 'enginyeria\ del\ software', TI_{vi}, TI_{vf}, T_{ti}, T_{tf}) \wedge T_{vi} \leq TI_{vf}$$

Si analitzem $ic3$ veiem que per comprovar aquesta restricció d'integritat hem de tenir en compte dos temps vàlids diferents, un d'anterior a l'altre.

$Ic3$ és concretament una restricció d'integritat en temps vàlid sincrona en temps de transacció i per condicions perquè defineix unes condicions que s'han de fer complir.

Les restriccions d'integritat en temps vàlid es caracteritzen perquè s'han de fer complir en diferents punts o períodes de temps vàlid per al punt de temps de transacció *Actual*. Les restriccions d'integritat en temps vàlid poden reflectir la història del temps vàlid en un període de temps de transacció específic, que serà el mateix a tots els predicats de la restricció. A la figura 5.11 podem veure l'àmbit més genèric possible de les restriccions d'integritat en temps vàlid.

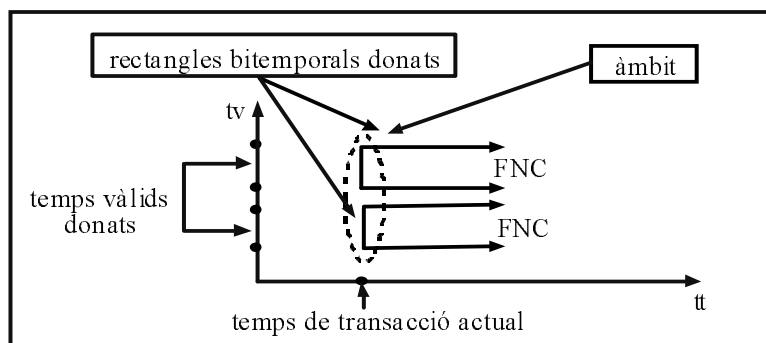


Figura 5.11. Àmbit de les restriccions d'integritat en temps vàlid

5.4.7. Quart grup: restriccions d'integritat bitemporals

Les restriccions d'integritat interelement de la taxonomia de Michael Böhlen que no van a parar al tercer grup formen el quart grup. Aquest grup es caracteritza per tenir diferents punt de temps de transacció, un dels quals és sempre l'*Actual*, i diferents punts de temps vàlid. Si haguéssim de donar un nom a aquest grup el podríem anomenar, per la seva complexitat, *restriccions d'integritat bitemporals*. Una restricció d'integritat bitemporal pot ser de tipus per condicions, periòdica en temps vàlid, periòdica en temps de transacció, dependent d'una data límit en temps vàlid i dependent d'una data límit en temps de transacció. També pot tenir sincronismes en temps vàlid i en temps de transacció, però mai tots dos alhora en tots els predicats.

Formalment una restricció d'integritat bitemporal seria del tipus:

$$ic \leftarrow p(X, T1_{vi}, T1_{vf}, T1_{ti}, T1_{tf}) [\wedge q(Y, T2_{vi}, T2_{vf}, T2_{ti}, T2_{tf}) \dots] \wedge \{ \text{condicions} / (\text{sincronisme temps transacció} / \text{sincronisme temps vàlid}) / \text{periodicitat temps transacció} / \text{data límit temps transacció} / \text{periodicitat temps vàlid} / \text{data límit temps vàlid} \}$$

On els claudàtors $[]$ indiquen que podem posar, opcionalment, tants predicats com vulguem, les claus $\{ \}$ indiquen que hem d'escollir entre una o diverses de les alternatives que hi trobem, i els parèntesis $()$ indiquen que només podem escollir una de les alternatives que hi trobem.

Ic4 seria una restricció d'integritat bitemporal:

$$ic4 \leftarrow \text{signatures}(C, P, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \text{signatures}(C, PI, T1_{vi}, T1_{vf}, T1_{ti}, T1_{tf}) \wedge PI < P \wedge T_{vf} < T1_{vi} \wedge T_{ti} < T1_{ti} - 10$$

Si analitzem *ic4* veiem que per comprovar aquesta restricció d'integritat hem de tenir en compte dos intervals de temps vàlids diferents, cadascun anterior a l'altre.

Ic4 és, concretament, una restricció d'integritat bitemporal, per condicions i

dependent d'una data límit en temps de transacció.

A la figura 5.12 podem veure l'àmbit més genèric possible de les restriccions d'integritat bitemporals.

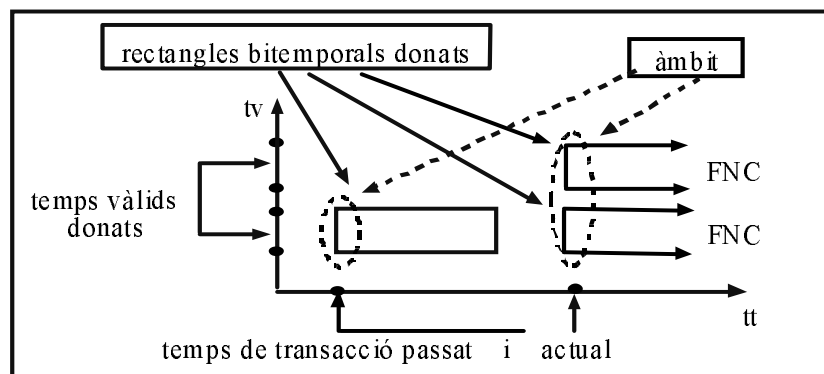


Figura 5.12. Àmbit de les restriccions d'integritat bitemporals

5.4.8. Avantatges de la nova classificació

Amb aquesta nova classificació en quatre grups ens adaptem al cas de les bd-dbts, on les restriccions d'integritat que trobarem les podrem emmarcar dins d'un d'aquests grups. D'aquesta manera, sabent les característiques de cada grup, podem conèixer propietats de les restriccions d'integritat que ens ajudin a treure avantatges per a la comprovació de la integritat en bd-dbts. Podrem veure tot això en el capítol següent, on presentem el nostre mètode de comprovació de restriccions d'integritat en bd-dbts.

La primera versió d'aquesta classificació es va presentar a [Mar00]. Un cop comprovada la seva aplicació a la comprovació de restriccions d'integritat en bd-dbts a [Mar01], es va completar i estendre tal com ha estat presentada en aquest capítol.

A la figura 5.13 podem veure un quadre resum de la classificació obtinguda com a resultat de la nostra anàlisi. La figura 5.13 mostra la interrelació entre els nostres quatre grups (restriccions d'integritat) i els sis grups de Michael Böhlen (restriccions d'integritat de Michael Böhlen) amb una línia. El símbol \surd indica

quines propietats expressades pels nostres set criteris (criteris), obtinguts de la taxonomia d'Anne Doucet *et al.*, compleix cada grup. Cal comentar que el primer grup només pot tenir sincronisme en temps de transacció i temps vàlid alhora. En canvi, en el cas del quart grup, només pot tenir sincronisme en temps vàlid o en temps de transacció, però mai alhora.

Criteris ↓	Restriccions d'integritat de Michael Böhlen				
	estàtiques	intraestat	intraelement	dinàmiques interestat interelement transició	
	Restriccions d'integritat	Primer grup actuals	Segon grup en t. de transacció	Terçer grup en temps vàlid	Quart grup bitemporals
Restriccions d'integritat per condicions	√	√	√	√	√
Restriccions d'integritat síncrones en t. vàlid	}		√		(√)
Restriccions d'integritat síncrones en t. transacció		√		√	(√)
Restriccions d'integritat periòdiques en t. vàlid				√	√
Restriccions d'integritat periòdiques en t. trans.			√		√
Restriccions d'integritat d.d.l. en t. vàlid				√	√
Restriccions d'integritat d.d.l. en t. transacció.			√		√

Figura 5.13. Quadre resum de la classificació de restriccions d'integritat del capítol

6. Descripció del mètode de comprovació de restriccions d'integritat en bases de dades deductives bitemporals

En aquest capítol presentem el nostre mètode de comprovació de restriccions d'integritat en bd-dbts. En el nostre mètode són cabdals els conceptes d'esdeveniment, regles de transició i regles d'esdeveniment, formalitzats a [Oli89] pel *model dels esdeveniments interns* com un enfocament per al disseny de models conceptuals deductius de sistemes d'informació, del qual ja hem parlat extensament en el capítol 3 així com de les seves aplicacions. A [Oli89] el temps vàlid i el temps de transacció són equivalents. En el nostre cas treballarem amb bases de dades bitemporals i distingim entre temps vàlid i temps de transacció. Per tant, el primer pas serà adaptar aquests conceptes clau al nostre àmbit.

6.1. Esdeveniments temporals

Un estat d'una bd-dbt [JSS92a] correspon a la fotografia de la base de dades definida per l'aplicació, en el nostre cas, de l'operació fotogràfica de temps de transacció [JCE+93]. Donats un estat D d'una bd-dbt en temps de transacció $t_{actual-1}$, on t_{actual} és el temps de transacció *Actual*, una transacció TR que es doni en temps de transacció t_{actual} i un estat D' de la mateixa bd-dbt actualitzada en temps de transacció t_{actual} . I tenint en compte, com hem vist al capítol 4, que la transacció TR consistirà en un conjunt de fets a inserir o a esborrar amb uns temps vàlids d'inici i finalització donats, però que sempre s'actualitzarà en temps de transacció *Actual*.

Donat un predicat p , un vector de constants x , un temps vàlid d'inici t_{vi} , un temps vàlid de finalització t_{vf} , un temps de transacció d'inici t_{ti} i un temps de transacció de finalització t_{tf} . Assumint que en l'estat D tenim $p(x, t_{vi}, t_{vf}, t_{ti}, t_{tf})$ i donat el mateix predicat p avaluat a l'estat D' , que ara anomenarem $p'(x, t_{vi}, t_{vf}, t_{ti}, t_{tf})$, es poden presentar dos casos:

$$p'(x, t_{vi}, t_{vf}, t_{ti}, t_{tf}) \text{ es continuï donant a } D' \\ (p(x, t_{vi}, t_{vf}, t_{ti}, t_{tf}) \text{ i } p'(x, t_{vi}, t_{vf}, t_{ti}, t_{tf}) \text{ tots dos són certs}). \quad (1)$$

$$p'(x, t_{vi}, t_{vf}, t_{ti}, t_{tf}) \text{ no es doni a } D' \\ (p(x, t_{vi}, t_{vf}, t_{ti}, t_{tf}) \text{ és cert, però } p'(x, t_{vi}, t_{vf}, t_{ti}, t_{tf}) \text{ és fals}). \quad (2)$$

I assumint que en l'estat D' tenim $p'(x, t_{vi}, t_{vf}, t_{ti}, t_{tf})$ i donat el mateix predicat p avaluat a l'estat D , també són possibles dos casos:

$$p(x, t_{vi}, t_{vf}, t_{ti}, t_{tf}) \text{ es continuï donant a } D \\ (p(x, t_{vi}, t_{vf}, t_{ti}, t_{tf}) \text{ i } p'(x, t_{vi}, t_{vf}, t_{ti}, t_{tf}) \text{ tots dos són certs}). \quad (3)$$

$$p(x, t_{vi}, t_{vf}, t_{ti}, t_{tf}) \text{ no es doni a } D \\ (p'(x, t_{vi}, t_{vf}, t_{ti}, t_{tf}) \text{ és cert, però } p(x, t_{vi}, t_{vf}, t_{ti}, t_{tf}) \text{ és fals}). \quad (4)$$

En el cas (2), figura 6.1, diem que s'ha produït un *esdeveniment d'esborrat temporal* durant la transició. L'esdeveniment d'esborrat el denotem $\delta p(x, t_{vi}, t_{vf})$, tal com havíem vist en el capítol 4.

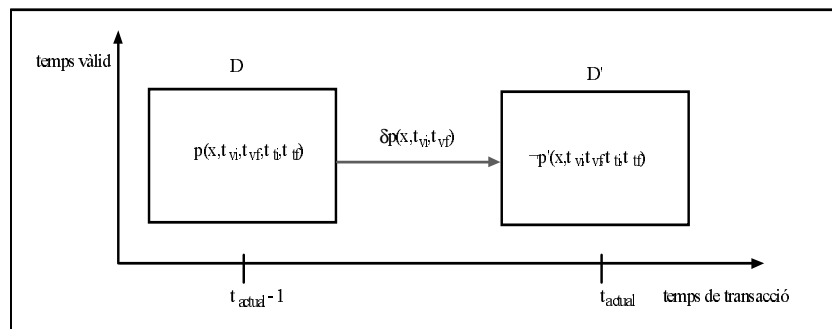


Figura. 6.1. Esdeveniment d'esborrat

En el cas (4), figura 6.2, diem que s'ha produït un *esdeveniment d'inserció temporal* durant la transició. L'esdeveniment d'inserció el denotem $ip(x, t_{vi}, t_{vf})$, tal com havíem vist en el capítol 4.

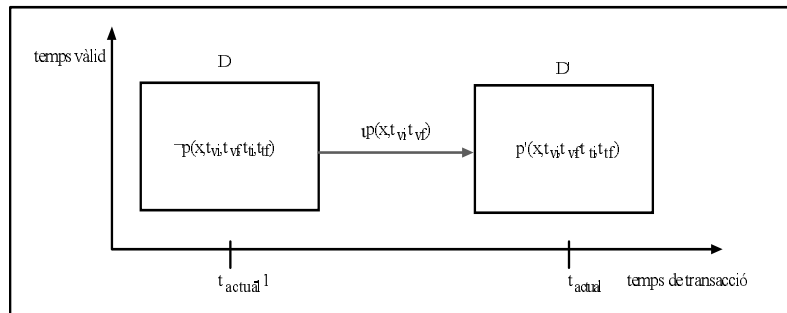


Figura. 6.2. Esdeveniment d'inserció

Tal com ja hem vist en el capítol 4, $\neg p(x, t_{vi}, t_{vf}, t_{ti}, t_{tf})$ només vol dir que $p(x)$ no hi és exactament entre t_{vi} i t_{vf} i, respecte al temps de transacció no hi és des de t_{ti} fins a t_{tf} . Si $p(x)$ existeix entre uns temps vàlids t_{vi} i t_{vf} , aleshores ha d'existir amb uns temps de transacció anteriors per poder dir que no existeix i si $p(x)$ existeix entre uns temps de transacció t_{ti} i t_{tf} , aleshores ha d'existir amb uns temps vàlids anteriors o posteriors al període entre t_{vi} i t_{vf} per poder dir que no existeix.

En el mateix capítol 4, hem vist que encara que un esdeveniment es produeix en un període de temps vàlid donat $ip(x, t_{vi}, t_{vf})$, també necessitem saber en quin temps de transacció es dona i, per tant, necessitarem un esdeveniment d'inserció que tindrà temps vàlid i temps de transacció. Aquest nou esdeveniment l'anomenarem:

$$i^*p(x, t_{vi}, t_{vf}, t_{ti}, t_{tf}) \text{ on } t_{ti} = t_{actual} \text{ i } t_{tf} = FNC$$

Amb l'esdeveniment d'esborrat $\delta p(x, t_{vi}, t_{vf})$ també necessitarem un esdeveniment d'esborrat amb temps vàlid i temps de transacció, que expressi que p finalitza en temps de transacció *Actual* i a partir d'ara $\neg p$ tindrà un temps de transacció d'inici *Actual* i un temps de transacció de finalització d'*FNC*. Aquest nou predicat l'anomenarem:

$$\delta^*p(x, t_{vi}, t_{vf}, t_{ti}, t_{tf}) \text{ on } t_{ti} = t_{actual} \text{ i } t_{tf} = FNC$$

Som conscients que ι^*p i δ^*p sempre tenen un temps de transacció d'inici *Actual* i un temps de transacció de finalització *FNC*, però cal explicitar-ho perquè es puguin utilitzar a les regles de transició. Formalment, associem un predicat d'esdeveniment d'inserció ι^*p a cada predicat bàsic, derivat o d'inconsistència p i un predicat d'esdeveniment d'esborrat δ^*p a cada predicat bàsic o derivat, definits com:

$$\begin{aligned} \forall X, T_{vi}, T_{vf}, T_{ti}, T_{tf} (\iota^*p(X, T_{vi}, T_{vf}, Actual, FNC) \leftrightarrow \\ p'(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf})) \end{aligned} \quad (5)$$

$$\begin{aligned} \forall X, T_{vi}, T_{vf}, T_{ti}, T_{tf} (\delta^*p(X, T_{vi}, T_{vf}, Actual, FNC) \leftrightarrow \\ p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg p'(X, T_{vi}, T_{vf}, T_{ti}, T_{tf})) \end{aligned} \quad (6)$$

on X és un vector de variables, T_{vi} és la variable de temps vàlid d'inici, T_{vf} és la variable de temps vàlid de finalització, T_{ti} és la variable de temps de transacció d'inici i T_{tf} és la variable de temps de transacció de finalització. Hem posat directament *Actual* i *FNC* on sabem que aquests són els temps vàlids d'inici i finalització.

De les definicions anteriors i tenint en compte que per inserir un fet no ha d'existir a la bd-dbt i per esborrar-lo ha d'existir a la bd-dbt, obtenim les equivalències següents:

$$\begin{aligned} \forall X, T_{vi}, T_{vf}, T_{ti}, T_{tf} (p'(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \leftrightarrow \\ [p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg \delta^*p(X, T_{vi}, T_{vf}, Actual, FNC)] \vee \\ \iota^*p(X, T_{vi}, T_{vf}, Actual, FNC)) \end{aligned} \quad (7)$$

$$\begin{aligned} \forall X, T_{vi}, T_{vf}, T_{ti}, T_{tf} (\neg p'(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \leftrightarrow \\ [\neg p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg \iota^*p(X, T_{vi}, T_{vf}, Actual, FNC)] \vee \\ \delta^*p(X, T_{vi}, T_{vf}, Actual, FNC)) \end{aligned} \quad (8)$$

que denoten el predicat p' en termes del predicat p i els predicats donats per la transició.

Si p és un predicat derivat, aleshores ι^*p i δ^*p representen, respectivament, insercions i esborrats induïts.

Si p és un predicat d'inconsistència, aleshores si ι^*p es dona durant la transició, voldrà dir que s'està produint una violació de les restriccions d'integritat. Cal tenir en compte que el predicat d'inconsistència δ^*p no es donarà en cap transició, ja que assumim que la bd-dbt és consistent abans de la transacció i, per tant, p és sempre fals.

També utilitzarem les definicions anteriors (5) i (6) per als predicats bàsics. En aquest cas, ι^*p i δ^*p representaran esdeveniments donats directament per la transacció.

Quan es dona un esdeveniment d'esborrat en realitat no esborrem la informació, sinó que l'esborrem lògicament prenent nota que a partir d'ara aquesta informació no existeix, però va existir en un temps de transacció anterior.

Exemple. Suposem la transacció TR en temps de transacció *Actual 3* sobre la bd-dbt següent:

$assignatures_r(lògica, 10000, 1, Ara, 2, FNC)$
$estudiants_r(jordi, 1, Ara, 2, FNC)$
$estudiants_r(maria, 1, Ara, 2, FNC)$
$matrícula_r(jordi, lògica, 1, Ara, 2, FNC)$
$matrícula_r(maria, lògica, 1, Ara, 2, FNC)$

Figura 6.3. Base de dades d'exemple en temps de transacció 2

$TR = \{assignatures(bases\ de\ dades, 12000, 2),\ matrícula(jordi, bases\ de\ dades, 2),\ \delta matrícula(maria, lògica, 1, Ara),\ matrícula(maria, lògica, 1, 2)\}$

En aquest cas es tracta d'una transacció coherent, que suposem que no viola cap restricció d'integritat, i després de l'execució de TR obtenim la bd-dbt següent:

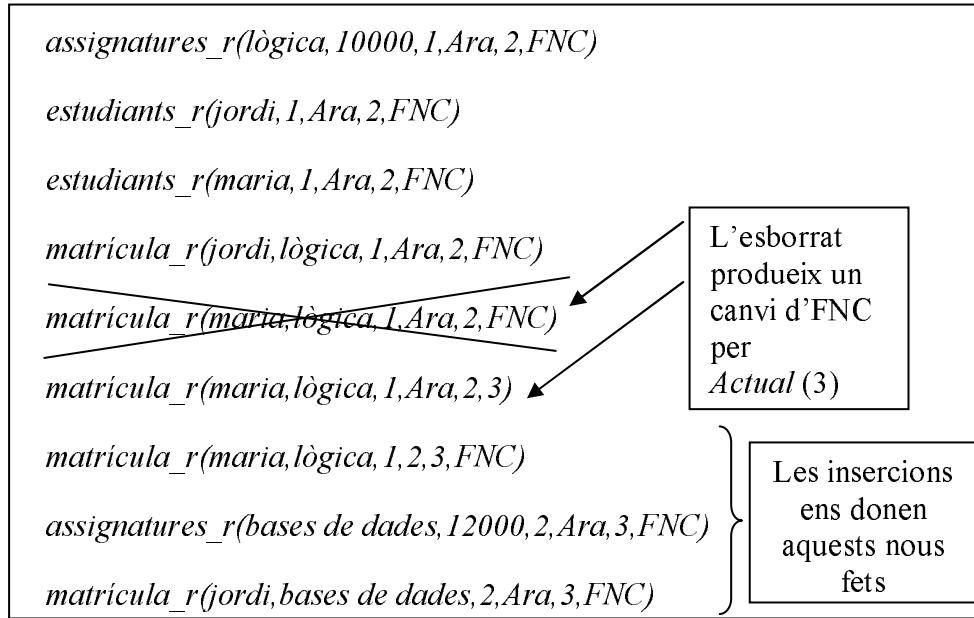


Figura 6.4. Base de dades d'exemple en temps de transacció 3

6.2. Regles de transició temporals

Donada la regla deductiva o d'inconsistència $p \leftarrow L_1, \dots, L_n$. Aquesta regla és avaluada en la bd-dbt en la forma $p' \leftarrow L_1', \dots, L_n'$, on L_i' ($i = 1, \dots, n$) s'obté reemplaçant el predicat q d' L_i per q' . Aleshores si rescrivim cada literal del cos de la regla per la seva definició equivalent donada a (7) o (8), obtenim una nova regla anomenada *regla de transició temporal*, que defineix el predicat p' en la bd-dbt actualitzada en termes dels predicats en temps de transacció *Actual- Δt* que apareixen en el cos de la regla i els esdeveniments ocorreguts durant la transició amb temps de transacció *Actual*. Més concretament, si L_i' és un literal positiu ordinari $q_i'(X_i, T_{vi}, T_{vf}, T_{ti}, T_{tf})$ apliquem (7) i el reemplaçem per:

$$(q_i(X_i, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg \delta * q_i(X_i, T_{vi}, T_{vf}, Actual, FNC)) \vee \iota * q_i(X_i, T_{vi}, T_{vf}, Actual, FNC)$$

i si L_i' és un literal negatiu ordinari $\neg q_i'(X_i, T_{vi}, T_{vf}, T_{ti}, T_{tf})$ apliquem (8) i obtenim:

$$(\neg q_i(X_i, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg \iota * q_i(X_i, T_{vi}, T_{vf}, Actual, FNC)) \vee \delta * q_i(X_i, T_{vi}, T_{vf}, Actual, FNC)$$

Si L_i és un predicat avaluable, només cal reemplaçar L_i' (positiu o negatiu) pel seu L_i actual.

Serà més senzill referir-se a les expressions obtingudes fins ara de la forma següent:

$$\begin{aligned} A(L_i') &= (q_i(X_i, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg \delta * q_i(X_i, T_{vi}, T_{vf}, Actual, FNC)) \\ &\quad \text{si } L_i' = q_i(X_i, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \\ &= (\neg q_i(X_i, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg \iota * q_i(X_i, T_{vi}, T_{vf}, Actual, FNC)) \\ &\quad \text{si } L_i' = \neg q_i(X_i, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \\ &= L_i \quad \text{si } L_i \text{ és avaluable} \\ N(L_i') &= \iota * q_i(X_i, T_{vi}, T_{vf}, Actual, FNC) \quad \text{si } L_i' = q_i(X_i, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \\ &= \delta * q_i(X_i, T_{vi}, T_{vf}, Actual, FNC) \quad \text{si } L_i' = \neg q_i(X_i, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \end{aligned}$$

Tant $A(L_i')$ com $N(L_i')$ expressen condicions per a cada L_i' que sigui cert. $A(L_i')$ correspon al cas que L_i' es doni perquè L_i sigui cert Antigament, o sigui en temps de transacció *Actual- Δt* i no s'hagi esborrat; mentre que $N(L_i')$ correspon al cas que $N(L_i')$ es doni perquè sigui Nou, induït o donat per la transacció, i fals abans. Notem que $A(L_i') \rightarrow L_i$ i $N(L_i') \rightarrow \neg L_i$.

Amb aquesta notació, les equivalències (7) i (8) esdevenen:

$$\begin{aligned} \forall X, T_{vi}, T_{vf}, T_{ti}, T_{tf} (p'(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \leftrightarrow \\ A(p'(X, T_{vi}, T_{vf}, T_{ti}, T_{tf})) \vee N(p'(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}))) \end{aligned} \quad (9)$$

$$\begin{aligned} \forall X, T_{vi}, T_{vf}, T_{ti}, T_{tf} (\neg p'(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \leftrightarrow \\ A(\neg p'(X, T_{vi}, T_{vf}, T_{ti}, T_{tf})) \vee N(\neg p'(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}))) \end{aligned} \quad (10)$$

I aplicant (9) i (10) a cadascun dels literals L_i' ($i = 1, \dots, n$), obtenim:

$$p'(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \leftarrow \bigwedge_{i=1}^{i=n} [A(L_i') \vee N(L_i') \mid A(L_i')] \quad (11)$$

on la primera opció es pren si L_i' és un literal ordinari, i la segona si L_i' és avaluable. Després de distribuir \bigwedge sobre \vee , obtenim un conjunt equivalent 2^k regles de transició, cadascuna amb la forma general següent:

$$p_j'(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \leftarrow \bigwedge_{i=1}^{i=n} [A(L_i') \mid N(L_i')] \quad \text{amb } j = 1, \dots, 2^k \quad (12)$$

on k és el nombre de literals ordinaris en la regla $p'(X, T_{vi}, T_{vf}, T_{ti}, T_{tf})$, i

$$p'(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \leftarrow p_j'(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \quad \text{amb } j = 1, \dots, 2^k \quad (13)$$

i per tant:

$$p'(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \leftarrow p_1'(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \vee \dots \vee p_{2^k}'(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \quad (14)$$

$$\text{on } p_1'(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \leftarrow A(L_1') \wedge \dots \wedge A(L_n')$$

...

$$p_{2^k}'(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \leftarrow N(L_1') \wedge \dots \wedge N(L_n')$$

Som conscients del gran nombre de regles de transició obtingudes i, des dels inicis del plantejament del mètode, hem treballat per obtenir simplificacions que les redueixin dràsticament i que veurem en aquest mateix capítol en apartats posteriors.

Per començar a simplificar: en el cas de les restriccions d'integritat ens trobem que $\neg uc_1'$ sempre serà certa perquè hem assumit que la bd-dbt serà consistent en un temps de transacció anterior a *Actual*. Això serà així excepte en el temps inicial de la bd-dbt, quan es dona la primera transacció i si hi ha alguna restricció d'integritat amb tots els predicats negats. En aquest cas s'hauria de mirar

aquesta primera regla de transició només una vegada, en aquest temps inicial, que ja havíem anomenat t_{t0} .

Per exemple, en la restricció d'integritat *ic2* del capítol 5:

$$ic2 \leftarrow estudiants(S, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge matricula(S, C, T_{vi}, T_{vf}, TI_{ti}, TI_{tf}) \wedge T_{ti} \leq TI_{ti} \wedge T_{tf} < TI_{tf}$$

La primera regla de transició que obtenim:

$$ic2_1' \leftarrow estudiants(S, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg \delta * estudiants(S, T_{vi}, T_{vf}, Actual, FNC) \wedge matricula(S, C, T_{vi}, T_{vf}, TI_{ti}, TI_{tf}) \wedge \neg \delta * matricula(S, C, T_{vi}, T_{vf}, Actual, FNC) \wedge T_{ti} \leq TI_{ti} \wedge T_{tf} < TI_{tf}$$

sempre s'avaluarà a fals perquè d'altra manera suposaria que la bd-dbt fos inconsistent en un temps de transacció anterior a *Actual* i, com estem segurs que no és així, podem eliminar sempre la primera regla de transició que obtinguem.

6.3. Regles d'esdeveniments temporals

Donat un predicat derivat $p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf})$. Els esdeveniments temporals d'inserció definits a (5) com:

$$\forall X, T_{vi}, T_{vf}, T_{ti}, T_{tf} (\iota * p(X, T_{vi}, T_{vf}, Actual, FNC) \leftrightarrow p'(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}))$$

I reemplaçant $p'(X, T_{vi}, T_{vf}, T_{ti}, T_{tf})$ per la seva definició equivalent de (13), obtenim:

$$\iota * p(X, T_{vi}, T_{vf}, Actual, FNC) \leftarrow p_i'(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \quad \text{amb } i = 2, \dots, 2^k \quad (15)$$

El conjunt de regles que acabem d'obtenir (15); l'anomenem *regles d'esdeveniments temporals d'inserció*. Aquestes ens permeten deduir quan es donen en una transició $\iota * p$ (insercions induïdes). La regla corresponent a $i=1$, tal com hem explicat a la secció anterior, es pot eliminar, ja que no pot produir cap

ι^*p . A més, si p és un predicat d'inconsistència, podem eliminar $\neg p$ de la regla d'esdeveniment temporal d'inserció, ja que partim del fet que la base de dades era consistent en temps de transacció anterior a l'*Actual* i per tant $\neg ic$ sempre serà cert.

Si p és un predicat derivat, procedirem seguint el mètode explicat a [Oli91]. Donada una regla que defineixi p , la conjunció dels literals del cos de la regla, tenint un vector de variables no temporals incloses o iguals a les variables del cap de la regla, la denotarem $U(p)$. $E(p)$ serà la conjunció dels literals del cos de la regla, on trobarem les variables que no estan en el cap de la regla. Si $p_i'(X, T_{vi}, T_{vf}, T_{ti}, T_{tf})$ té algun literal $N(L_h)$ a $U(p_i'(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}))$, aleshores podem eliminar $\neg p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf})$ de la regla d'esdeveniment intern. Si $p_i'(X, T_{vi}, T_{vf}, T_{ti}, T_{tf})$ té tots els literals d' $U(p_i'(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}))$ de la forma $A(L_h)$, aleshores podem eliminar aquests literals de $\neg p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf})$ de la regla d'esdeveniment intern i mantenir només els literals d' $E(p_i'(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}))$.

Donat el predicat derivat $p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf})$. Els esdeveniments temporals d'esborrat definits a (6) com:

$$\forall X, T_{vi}, T_{vf}, T_{ti}, T_{tf} (\delta^*p(X, T_{vi}, T_{vf}, Actual, FNC) \leftrightarrow p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg p'(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}))$$

I reemplaçant $p'(X, T_{vi}, T_{vf}, T_{ti}, T_{tf})$ per la seva definició equivalent a (14), i després d'aplicar les transformacions proposades a [Oli91], obtenim el conjunt de regles per a cada L_h d' $U(p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}))$:

$$\begin{aligned} \delta^*p(X, T_{vi}, T_{vf}, Actual, FNC) &\leftarrow L_1 \wedge \dots \wedge L_{h-1} \wedge [\iota^*L_h \vee \delta^*L_h] \wedge \\ L_{h+1} \wedge \dots \wedge L_n \wedge E(p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf})) &\wedge \neg p_i'(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \end{aligned} \quad (16)$$

on a $[\iota^*L_h \vee \delta^*L_h]$, la primera opció és per a literals negats i la segona per a literals positius. I per a cada L_h d' $E(p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}))$, obtenim:

$$\begin{aligned} \delta^*p(X, T_{vi}, T_{vf}, Actual, FNC) &\leftarrow U(p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf})) \wedge \\ L_1 \wedge \dots \wedge L_{h-1} \wedge [\iota^*L_h \vee \delta^*L_h] \wedge L_{h+1} \wedge \dots \wedge L_n \wedge \neg p_i'(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \end{aligned} \quad (17)$$

El conjunt de regles que acabem d'obtenir a (16) i (17); l'anomenem *regles d'esdeveniments temporals d'esborrat*. Aquestes ens permeten deduir quan es donen en una transició δ^*p (esborrats induïts).

6.4. La base de dades augmentada

Donada una bd-dbt BD . Denotem *bd-dbt augmentada* $A(BD)$, basant-nos en el concepte de base de dades deductiva augmentada $A(D)$ definit a [Oli91], a la bd-dbt BD , en les seves regles de transició temporals i en les seves regles d'esdeveniments temporals. Mitjançant l' $A(BD)$ i la resolució SLDNF, es pot fer la comprovació de les restriccions d'integritat en bd-dbts, tal com explicarem a la secció següent.

Si la resolució SLDNF és completa per a BD , aleshores també serà completa per a $A(BD)$, tal com a [Oli91] es demostra per a D i $A(D)$.

6.5. Comprovació de restriccions d'integritat en bases de dades deductives bitemporals

La bd-dbt augmentada descrita en la secció anterior pot ser utilitzada directament per comprovar si una transacció produeix inconsistències.

Donada una bd-dbt BD , una bd-dbt augmentada $A(BD)$ i una transacció TR . Si TR produeix alguna violació de la integritat, aleshores algun dels esdeveniments temporal d'inserció t^*ic es donarà durant la transició. Utilitzant el procediment de resolució SLDNF, TR violarà la restricció d'integritat ic si té èxit l'objectiu $\leftarrow t^*ic$ tenint com a conjunt d'entrada $A(BD) \cup TR$. Si totes les branques de l'espai de cerca SLDNF per a $A(BD) \cup TR \cup \{\leftarrow t^*ic\}$ són branques errades, aleshores TR no violarà ic , com es pot veure a la figura 6.5.

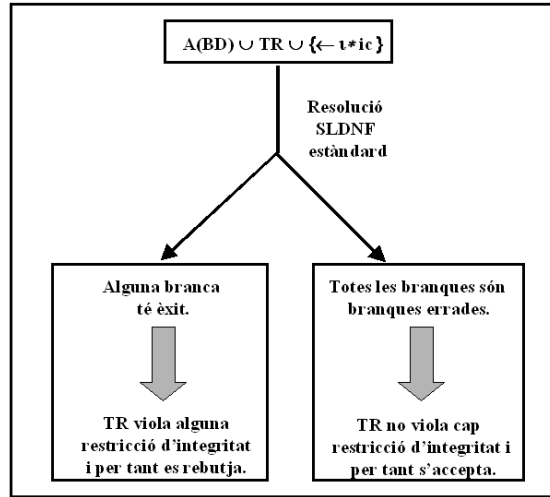


Figura 6.5. El mètode dels esdeveniments interns per a una base de dades deductiva bitemporal augmentada $A(BD)$, una transacció TR i una restricció d'integritat ic

Donat $t*ic$, podem utilitzar les regles de transició d' ic per comprovar la consistència de la bd-dbt. A continuació, il·lustrarem la utilitat de les regles de transició temporals per a la comprovació de restriccions d'integritat en l'àmbit bitemporal, tenint en compte l'estudi realitzat en el capítol 5, i assegurant que efectivament les regles de transició detecten tots els casos possibles en què es poden violar les restriccions d'integritat temporals.

6.6. Restriccions d'integritat actuals

Considerem la restricció d'integritat actual icI :

$$icI \leftarrow assignatures(C,P, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge P < 10000$$

Les regles de transició temporals que obtenim després de reemplaçar els literals i distribuir \wedge sobre \vee són:

$$icI' \leftarrow icI_2'$$

$$icI_2' \leftarrow t*assignatures(C,P, T_{vi}, T_{vf}, Actual, FNC) \wedge P < 10000$$

Aquesta regla de transició es podrà donar perquè si inserim una assignatura en temps de transacció *Actual*, amb un preu menor de 10.000, aleshores estariem violant *ic1*.

$TR1 = \{assignatures(algorismes, 8000, 2, Ara)\}$

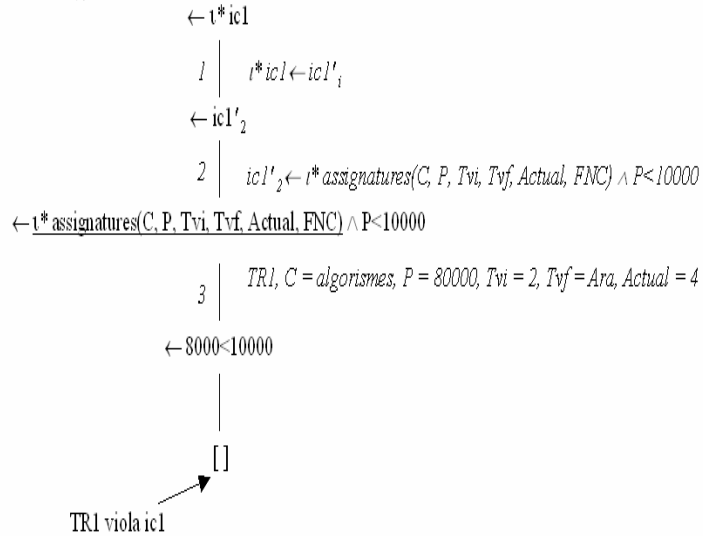


Figura 6.6. Violació d'*ic1* per *ic1₂'*

Suposem una transacció *TR1* en temps de transacció *Actual 4* sobre la base de dades que hem presentat abans a la figura 6.4. *TR1* és una transacció coherent i violarà *ic1* per *ic1₂'*. A la figura 6.6 podem veure l'arbre de derivació de com s'aplicaria el nostre mètode de comprovació per detectar la violació d'*ic1*.

Notem que en aquest cas la violació de la integritat de la bd-dbt s'arriba a detectar sense accedir en cap moment a la base de dades i que quan s'accedeix a la transacció *t* i *t** es poden unificar, ja que *t** només afegeix temps de transacció a *t*.

Tenint en compte que una restricció d'integritat actual, tal com hem vist al capítol 5, només pot tenir condicions i sincronismes de temps vàlid i temps de transacció alhora, ja que els temps vàlids i els de transacció de tots els predicats que apareguin a la seva definició han de ser iguals, analitzarem tots els casos possibles que es poden donar en les restriccions d'integritat actuals.

Cas 1. Suposem la següent restricció d'integritat actual amb tots els predicats de la definició positius:

$$ic \leftarrow p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge q(Y, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \text{condicions}$$

Les regles de transició temporals serien:

$$ic_2' \leftarrow p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg \delta * p(X, T_{vi}, T_{vf}, Actual, FNC) \wedge \iota * q(Y, T_{vi}, T_{vf}, Actual, FNC) \wedge \text{condicions}$$

Aquesta regla de transició es pot eliminar perquè no violarà mai *ic*, ja que en aquest cas si s'insereix *q* en temps de transacció d'inici *Actual*, mai no serà igual al temps de transacció d'inici de *p*.

$$ic_3' \leftarrow \iota * p(X, T_{vi}, T_{vf}, Actual, FNC) \wedge q(Y, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg \delta * q(Y, T_{vi}, T_{vf}, Actual, FNC) \wedge \text{condicions}$$

Aquesta regla de transició es pot eliminar perquè no violarà mai *ic*, ja que en aquest cas si s'insereix *p* en temps de transacció d'inici *Actual*, mai no serà igual al temps de transacció d'inici de *q*.

$$ic_4' \leftarrow \iota * p(X, T_{vi}, T_{vf}, Actual, FNC) \wedge \iota * q(Y, T_{vi}, T_{vf}, Actual, FNC) \wedge \text{condicions}$$

Aquesta regla de transició es l'única que realment pot violar *ic*, ja que en aquest cas si s'insereix *p* en temps de transacció d'inici *Actual*, serà igual al temps de transacció d'inici de *q*. Ara, doncs, caldrà mirar si es compleixen les condicions de la definició per comprovar si es viola o no la restricció d'integritat.

Cas 2. Suposem la següent restricció d'integritat actual amb predicats positius i predicats negats en la definició:

$$ic \leftarrow p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg q(Y, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \text{condicions}$$

Les regles de transició temporals serien:

$$ic_2' \leftarrow p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg \delta * p(X, T_{vi}, T_{vf}, Actual, FNC) \wedge \delta * q(Y, T_{vi}, T_{vf}, Actual, FNC) \wedge \text{condicions}$$

Aquesta regla de transició es pot eliminar perquè no violarà mai *ic*, ja que en aquest cas si s'esborra *q*, el temps de transacció d'inici de $\neg q$ serà *Actual* i mai no serà igual al temps de transacció d'inici de *p*.

$$ic_3' \leftarrow \iota * p(X, T_{vi}, T_{vf}, Actual, FNC) \wedge \neg q(Y, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg \iota * q(Y, T_{vi}, T_{vf}, Actual, FNC) \wedge \text{condicions}$$

Aquesta regla de transició es pot eliminar perquè no violarà mai *ic*, ja que en aquest cas si s'insereix *p* en temps de transacció d'inici *Actual*, mai no serà igual al temps de transacció d'inici de $\neg q$.

$$ic_4' \leftarrow \iota * p(X, T_{vi}, T_{vf}, Actual, FNC) \wedge \delta * q(Y, T_{vi}, T_{vf}, Actual, FNC) \wedge \text{condicions}$$

Aquesta regla de transició es pot donar, ja que en aquest cas si s'insereix *p* en temps de transacció d'inici *Actual*, serà igual al temps de transacció d'inici de $\neg q$, que ara s'esborra amb un temps de transacció d'inici *Actual*. Ara, doncs, caldrà mirar si es compleixen les condicions de la definició per comprovar si es viola o no la restricció d'integritat.

Cas 3. Suposem la següent restricció d'integritat actual amb tots els predicats de la definició negats:

$$ic \leftarrow \neg p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg q(Y, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \text{condicions}$$

Les regles de transició temporals serien:

$$ic_2' \leftarrow \neg p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg \iota * p(X, T_{vi}, T_{vf}, Actual, FNC) \wedge \delta * q(Y, T_{vi}, T_{vf}, Actual, FNC) \wedge \text{condicions}$$

Aquesta regla de transició es pot eliminar perquè no violarà mai *ic*, ja que en aquest cas si s'esborra *q* en temps de transacció d'inici *Actual*, mai no serà igual al temps de transacció d'inici de $\neg p$.

$$ic_3' \leftarrow \delta * p(X, T_{vi}, T_{vf}, Actual, FNC) \wedge \neg q(Y, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg \iota * q(Y, T_{vi}, T_{vf}, Actual, FNC) \wedge \text{condicions}$$

Aquesta regla de transició es pot eliminar perquè no violarà mai *ic*, ja que en aquest cas si s'esborra *p* en temps de transacció d'inici *Actual*, mai no serà igual al temps de transacció d'inici de $\neg q$.

$$ic_4' \leftarrow \delta * p(X, T_{vi}, T_{vf}, Actual, FNC) \wedge \delta * q(Y, T_{vi}, T_{vf}, Actual, FNC) \wedge \text{condicions}$$

Aquesta regla de transició es l'única que realment pot violar *ic*, ja que en aquest cas si s'esborra *p* en temps de transacció d'inici *Actual*, serà igual al temps de transacció d'inici de $\neg q$, que també s'esborra amb temps de transacció d'inici *Actual*. Ara, doncs, caldrà mirar si es compleixen les condicions de la definició per comprovar si es viola o no la restricció d'integritat.

6.7. Restriccions d'integritat en temps de transacció

Considerem la restricció d'integritat en temps de transacció:

$$ic_2 \leftarrow \text{estudiants}(S, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \text{matrícula}(S, C, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge T_{ti} \leq T_{ti} \\ \wedge T_{tf} < T_{tf}$$

Les regles de transició temporals que obtenim després de reemplaçar els literals i distribuir \wedge sobre \vee són:

$$ic_{2'} \leftarrow ic_{2'} \quad i = 2, \dots, 4$$

$$ic_{2_2'} \leftarrow \text{estudiants}(S, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg \delta * \text{estudiants}(S, T_{vi}, T_{vf}, Actual, FNC) \wedge \\ \iota * \text{matrícula}(S, C, T_{vi}, T_{vf}, Actual, FNC) \wedge T_{ti} \leq Actual \wedge T_{tf} < FNC$$

Aquesta regla de transició es podrà donar perquè si volem inserir una matrícula en temps de transacció *Actual*, precisament, el que haurem de comprovar és que hi hagi un estudiant en temps de transacció $T_{ti} \leq Actual$ i en temps de transacció T_{tf} igual a *FNC*.

$$ic_{2_3'} \leftarrow \iota * \text{estudiants}(S, T_{vi}, T_{vf}, Actual, FNC) \wedge \text{matrícula}(S, C, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \\ \neg \delta * \text{matrícula}(S, C, T_{vi}, T_{vf}, Actual, FNC) \wedge Actual \leq T_{ti} \wedge FNC < T_{tf}$$

Aquesta regla de transició mai no es podrà donar perquè no es pot donar el cas que *FNC* sigui més petit que cap altre temps de transacció.

$$ic_{2_4'} \leftarrow \iota * \text{estudiants}(S, T_{vi}, T_{vf}, Actual, FNC) \wedge \\ \iota * \text{matrícula}(S, C, T_{vi}, T_{vf}, Actual, FNC) \wedge Actual \leq Actual \wedge FNC < FNC$$

Aquesta regla de transició mai no es podrà donar perquè no es pot donar el cas que *FNC* sigui més petit que cap altre temps de transacció.

$TR2 = \{unmatricula(maria, logica, 3, 6)\}$

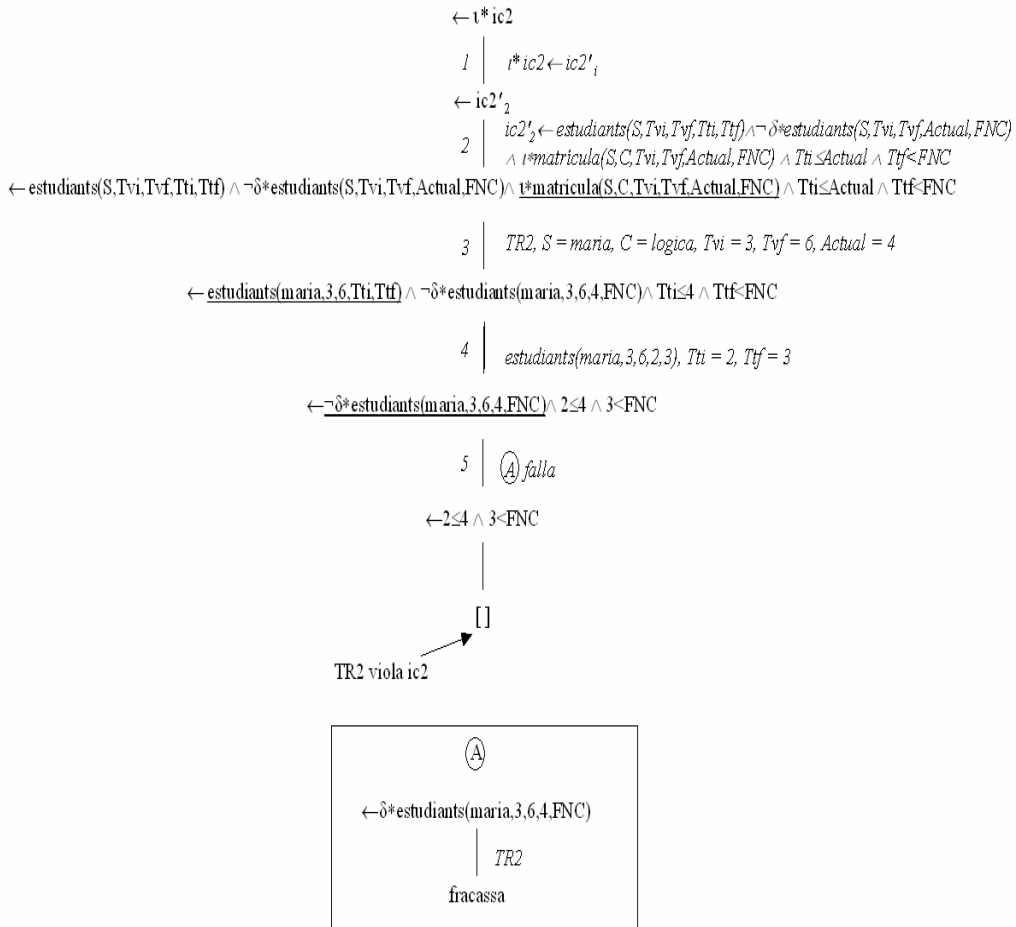


Figura 6.7. Violació d'*ic2* per *ic2'*

Suposem una transacció *TR2* en temps de transacció *Actual 4* sobre la base de dades que hem presentat abans a la figura 6.4, però suposem ara que la *maria* ha estat esborrada com a estudiant *i*, per tant, tenim el fet següent: *estudiants_r(maria, 3, 6, 2, 3)*. *TR2* és una transacció coherent i violarà *ic2* per *ic2'*. A la figura 6.7 podem veure l'arbre de derivació de com s'aplicaria el

nostre mètode de comprovació per detectar la violació d'*ic2*.

Els passos 1 i 2 són passos de la resolució SLDNF, on les regles de la base de dades augmentada actuen com a clàusules d'entrada.

En el pas 3 els predicats seleccionats fan referència a la transacció i per resoldre hem d'anar a instanciar a la transacció. En el pas 4 el predicat seleccionat fa referència a la base de dades i per resoldre hem d'anar a instanciar a la base de dades.

En el pas A, el literal seleccionat és: $\neg \delta * \text{estudiants}(\text{maria}, 3, 6, 4, \text{FNC})$. Per arribar a una derivació amb èxit, l'espai de cerca SLDNF ha de fracassar finitament per l'arbre subsidiari de: $\{\leftarrow \delta * \text{estudiants}(\text{maria}, 3, 6, 4, \text{FNC})\}$.

Cal fer constar que en el tercer pas podríem haver escollit el literal $\text{estudiants}(S, T_{vi}, T_{vf}, T_{ti}, T_{tf})$ en lloc del literal $\delta * \text{matrícula}(S, C, T_{vi}, T_{vf}, \text{Actual}, \text{FNC})$. Però, tenint present que en la majoria de casos el nombre de fets és molt més gran que el nombre d'esdeveniments que es donen en una transacció, seguirem l'estratègia d'escollir primer els esdeveniments d'una transacció abans dels fets d'una regla de transició.

Tenint en compte que una restricció d'integritat en temps de transacció, tal com hem vist al capítol 5, pot tenir condicions, sincronismes en temps vàlid i periodicitats i dates límits en temps de transacció, els temps vàlids de tots els predicats que apareguin a la definició han de ser iguals i els temps de transacció són diferents, analitzarem tots els casos possibles que es poden donar en les restriccions d'integritat en temps de transacció. Tractarem conjuntament les condicions, els sincronismes en temps vàlid i també les periodicitats en temps de transacció i les dates límits en temps de transacció, ja que són tots, en certa manera, diferents tipus de condicions que utilitzen o no variables temporals. En els diversos casos que analitzem, ho agruparem tot sota el nom de *condicions*. Suposarem sempre que T_{ti} és diferent de TI_{ti} i/o T_{tf} és diferent de TI_{tf} , d'altra banda, *ic* seria una restricció d'integritat actual.

Cas 1. Suposem la següent restricció d'integritat en temps de transacció amb tots els predicats de la definició positius:

$$ic \leftarrow p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge q(Y, T_{vi}, T_{vf}, TI_{ti}, TI_{tf}) \wedge \text{condicions}$$

Les regles de transició temporals serien:

$$ic_2' \leftarrow p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg \delta * p(X, T_{vi}, T_{vf}, Actual, FNC) \wedge \iota * q(Y, T_{vi}, T_{vf}, Actual, FNC) \wedge \text{condicions}$$

Aquesta regla de transició pot violar *ic*, ja que en aquest cas si s'insereix *q* en temps de transacció d'inici *Actual*, no serà igual al temps de transacció d'inici de *p*, que ja hi era. Ara, doncs, caldrà mirar si es compleixen les condicions de la definició per comprovar si és necessària aquesta regla de transició.

$$ic_3' \leftarrow \iota * p(X, T_{vi}, T_{vf}, Actual, FNC) \wedge q(Y, T_{vi}, T_{vf}, TI_{ti}, TI_{tf}) \wedge \neg \delta * q(Y, T_{vi}, T_{vf}, Actual, FNC) \wedge \text{condicions}$$

Aquesta regla de transició pot violar *ic*, ja que en aquest cas si s'insereix *p* en temps de transacció d'inici *Actual*, mai no serà igual al temps de transacció d'inici de *q*, que ja hi era. Ara, doncs, caldrà mirar si es compleixen les condicions de la definició per comprovar si és necessària aquesta regla de transició.

$$ic_4' \leftarrow \iota * p(X, T_{vi}, T_{vf}, Actual, FNC) \wedge \iota * q(Y, T_{vi}, T_{vf}, Actual, FNC) \wedge \text{condicions}$$

Aquesta regla de transició es pot eliminar, ja que en aquest cas si s'insereix *p* en temps de transacció d'inici *Actual*, serà igual al temps de transacció d'inici de *q*. I el mateix passarà amb el temps de transacció de finalització. Si es donés aquest cas, *ic* no hagués estat una restricció d'integritat en temps de transacció sinó que hagués estat actual.

Cas 2. Suposem la següent restricció d'integritat en temps de transacció amb predicats positius i predicats negats en la definició:

$$ic \leftarrow p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg q(Y, T_{vi}, T_{vf}, TI_{ti}, TI_{tf}) \wedge \text{condicions}$$

Les regles de transició temporals serien:

$$ic_2' \leftarrow p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg \delta * p(X, T_{vi}, T_{vf}, Actual, FNC) \wedge \delta * q(Y, T_{vi}, T_{vf}, Actual, FNC) \wedge \text{condicions}$$

Aquesta regla de transició es pot donar perquè en aquest cas si s'esborra *q*, el temps de transacció d'inici de $\neg q$ serà *Actual* i serà diferent del temps de

transacció d'inici de p . Ara, doncs, caldrà mirar si es compleixen les condicions de la definició per comprovar si és necessària aquesta regla de transició.

$$ic_3' \leftarrow \iota * p(X, T_{vi}, T_{vf}, Actual, FNC) \wedge \neg q(Y, T_{vi}, T_{vf}, TI_{ti}, TI_{tf}) \wedge \neg \iota * q(Y, T_{vi}, T_{vf}, Actual, FNC) \wedge \text{condicions}$$

Aquesta regla de transició es pot donar perquè en aquest cas si s'insereix p en temps de transacció d'inici $Actual$, mai no serà igual al temps de transacció d'inici de $\neg q$. Ara, doncs, caldrà mirar si es compleixen les condicions de la definició per comprovar si és necessària aquesta regla de transició.

$$ic_4' \leftarrow \iota * p(X, T_{vi}, T_{vf}, Actual, FNC) \wedge \delta * q(Y, T_{vi}, T_{vf}, Actual, FNC) \wedge \text{condicions}$$

Aquesta regla de transició es pot eliminar perquè no violarà mai ic , ja que en aquest cas si s'insereix p en temps de transacció d'inici $Actual$, serà igual al temps de transacció d'inici de $\neg q$, que ara s'esborra amb un temps de transacció d'inici $Actual$. I el mateix passarà amb el temps de transacció de finalització. Si es donés aquest cas, ic no hagués estat una restricció d'integritat en temps de transacció sinó que hagués estat actual.

Cas 3. Suposem la següent restricció d'integritat en temps de transacció amb tots els predicats de la definició negats:

$$ic \leftarrow \neg p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg q(Y, T_{vi}, T_{vf}, TI_{ti}, TI_{tf}) \wedge \text{condicions}$$

Les regles de transició temporals serien:

$$ic_2' \leftarrow \neg p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg \iota * p(X, T_{vi}, T_{vf}, Actual, FNC) \wedge \delta * q(Y, T_{vi}, T_{vf}, Actual, FNC) \wedge \text{condicions}$$

Aquesta regla de transició es pot donar perquè en aquest cas si s'esborra q en temps de transacció d'inici $Actual$, mai no serà igual al temps de transacció d'inici de $\neg p$. Ara, doncs, caldrà mirar si es compleixen les condicions de la definició per comprovar si és necessària aquesta regla de transició.

$$ic_3' \leftarrow \delta * p(X, T_{vi}, T_{vf}, Actual, FNC) \wedge \neg q(Y, T_{vi}, T_{vf}, TI_{ti}, TI_{tf}) \wedge \neg \iota * q(Y, T_{vi}, T_{vf}, Actual, FNC) \wedge \text{condicions}$$

Aquesta regla de transició es pot donar perquè en aquest cas si s'esborra p en temps de transacció d'inici *Actual*, mai no serà igual al temps de transacció d'inici de $\neg q$. Ara, doncs, caldrà mirar si es compleixen les condicions de la definició per comprovar si és necessària aquesta regla de transició.

$$ic_4' \leftarrow \delta^*p(X, T_{vi}, T_{vf}, Actual, FNC) \wedge \delta^*q(Y, T_{vi}, T_{vf}, Actual, FNC) \wedge \text{condicions}$$

Aquesta regla de transició es pot eliminar, ja que en aquest cas si s'esborra p en temps de transacció d'inici *Actual*, serà igual al temps de transacció d'inici de $\neg q$, que també s'esborrarà en temps de transacció *Actual*. I el mateix passarà amb el temps de transacció de finalització. Si es donés aquest cas, ic no hagués estat una restricció d'integritat en temps de transacció sinó que hagués estat actual.

6.8. Restriccions d'integritat en temps vàlid

Considerem la restricció d'integritat en temps vàlid:

$$ic_3 \leftarrow \text{matrícula}(S, 'bases de dades', T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \\ \text{matrícula}(S, 'enginyeria del software', TI_{vi}, TI_{vf}, T_{ti}, T_{tf}) \wedge T_{vi} \leq TI_{vf}$$

Les regles de transició temporals que obtenim després de reemplaçar els literals i distribuir \wedge sobre \vee són:

$$ic_3' \leftarrow ic_3 \quad i = 2, \dots, 4$$

$$ic_{3_2}' \leftarrow \text{matrícula}(S, 'bases de dades', T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \\ \neg \delta^* \text{matrícula}(S, 'bases de dades', T_{vi}, T_{vf}, Actual, FNC) \wedge \\ \iota^* \text{matrícula}(S, 'enginyeria del software', TI_{vi}, TI_{vf}, Actual, FNC) \wedge \\ T_{vi} \leq TI_{vf}$$

Aquesta regla de transició no es podrà donar perquè si volem inserir una matrícula a l'assignatura d'*enginyeria del software* en temps de transacció d'inici *Actual*, però ja teníem una matrícula del mateix estudiant a *bases de dades* en un temps de transacció d'inici, aleshores podem dir que els temps de transacció mai no seran iguals.

$$\begin{aligned}
 ic3_3' \leftarrow & \iota * matricula(S, 'bases de dades', T_{vi}, T_{vf}, Actual, FNC) \wedge \\
 & matricula(S, 'enginyeria del software', T1_{vi}, T1_{vf}, T_{ti}, T_{tf}) \wedge \\
 & \neg \delta * matricula(S, 'enginyeria del software', T1_{vi}, T1_{vf}, Actual, FNC) \wedge \\
 & T_{vi} \leq T1_{vf}
 \end{aligned}$$

Aquesta regla de transició no es podrà donar perquè si volem inserir una matrícula a l'assignatura de *bases de dades* en temps de transacció d'inici *Actual*, però ja teníem una matrícula del mateix estudiant a *enginyeria del software* en un temps de transacció d'inici, aleshores podem dir que els temps de transacció mai no seran iguals.

$$\begin{aligned}
 ic3_4' \leftarrow & \iota * matricula(S, 'bases de dades', T_{vi}, T_{vf}, Actual, FNC) \wedge \\
 & \iota * matricula(S, 'enginyeria del software', T1_{vi}, T1_{vf}, Actual, FNC) \wedge \\
 & T_{vi} \leq T1_{vf}
 \end{aligned}$$

Només es podrà donar aquesta darrera regla de transició perquè l'única manera de violar aquesta restricció d'integritat és si inserim dues matrícules, una a *bases de dades* i una a *enginyeria del software* del mateix estudiant, en el mateix temps de transacció *Actual* i amb un temps vàlid d'inici de *bases de dades* menor o igual que el final d'*enginyeria del software*.

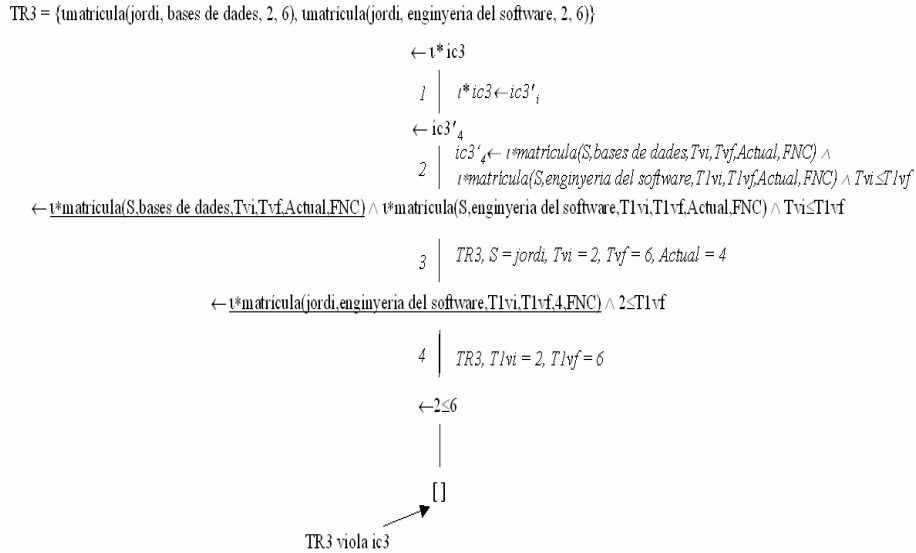


Figura 6.8. Violació d'*ic3* per *ic3_4'*

Suposem una transacció *TR3* en temps de transacció *Actual 4* sobre la base de dades que hem presentat abans a la figura 6.4, però suposem ara que en *jordi* no ha estat mai matriculat de *bases de dades*. *TR3* és una transacció coherent i violarà *ic3* per *ic34'*. A la figura 6.8 podem veure l'arbre de derivació de com s'aplicaria el nostre mètode de comprovació per detectar la violació d'*ic3*.

Notem que en aquest cas la violació de la integritat de la bd-dbt s'arriba a detectar sense accedir en cap moment a la base de dades.

Tenint en compte que una restricció d'integritat de temps vàlid, tal com hem vist al capítol 5, pot tenir condicions, sincronismes en temps de transacció, periodicitats i dates límits en temps vàlid, els temps de transacció de tots els predicats que apareguin a la definició han de ser iguals i els temps vàlids són diferents, analitzarem tots els casos possibles que es poden donar en les restriccions d'integritat en temps vàlid. Tractarem conjuntament les condicions, els sincronismes en temps de transacció i també les periodicitats en temps vàlid i les dates límits en temps vàlid, ja que són tots, en certa manera, diferents tipus de condicions que utilitzen o no variables temporals. En els diversos casos que analitzem, ho agruparem tot sota el nom de *condicions*. Suposarem sempre que T_{vi} és diferent de TI_{vi} i/o T_{vf} és diferent de TI_{vf} , d'altra banda, *ic* seria una restricció d'integritat actual.

Cas 1. Suposem la següent restricció d'integritat en temps vàlid amb tots els predicats de la definició positius:

$$ic \leftarrow p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge q(Y, TI_{vi}, TI_{vf}, T_{ti}, T_{tf}) \wedge \text{condicions}$$

Les regles de transició temporals serien:

$$ic_2' \leftarrow p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg \delta * p(X, T_{vi}, T_{vf}, Actual, FNC) \wedge \\ \iota * q(Y, TI_{vi}, TI_{vf}, Actual, FNC) \wedge \text{condicions}$$

Aquesta regla de transició no pot violar *ic*, ja que en aquest cas si s'insereix *q* en temps de transacció d'inici *Actual*, no serà igual al temps de transacció d'inici de *p*, que ja hi era.

$$ic_3' \leftarrow \iota * p(X, T_{vi}, T_{vf}, Actual, FNC) \wedge q(Y, TI_{vi}, TI_{vf}, T_{ti}, T_{tf}) \wedge \\ \neg \delta * q(Y, TI_{vi}, TI_{vf}, Actual, FNC) \wedge \text{condicions}$$

Aquesta regla de transició no pot violar ic , ja que en aquest cas si s'insereix p en temps de transacció d'inici *Actual*, no serà igual al temps de transacció d'inici de q , que ja hi era.

$$ic_4' \leftarrow \iota * p(X, T_{vi}, T_{vf}, Actual, FNC) \wedge \iota * q(Y, TI_{vi}, TI_{vf}, Actual, FNC) \wedge \text{condicions}$$

Aquesta regla de transició es pot donar, ja que en aquest cas si s'insereix p en temps de transacció d'inici *Actual*, serà igual al temps de transacció d'inici de q . I el mateix passarà amb el temps de transacció de finalització. Ara, doncs, caldrà mirar si es compleixen les condicions de la definició per comprovar si és necessària aquesta regla de transició.

Cas 2. Suposem la següent restricció d'integritat en temps vàlid amb predicats positius i predicats negats en la definició:

$$ic \leftarrow p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg q(Y, TI_{vi}, TI_{vf}, T_{ti}, T_{tf}) \wedge \text{condicions}$$

Les regles de transició temporals serien:

$$ic_2' \leftarrow p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg \delta * p(X, T_{vi}, T_{vf}, Actual, FNC) \wedge \delta * q(Y, TI_{vi}, TI_{vf}, Actual, FNC) \wedge \text{condicions}$$

Aquesta regla de transició es pot eliminar perquè no violarà mai ic , ja que en aquest cas si s'esborra q , el temps de transacció d'inici de $\neg q$ serà *Actual* i mai no serà igual al temps de transacció d'inici de p .

$$ic_3' \leftarrow \iota * p(X, T_{vi}, T_{vf}, Actual, FNC) \wedge \neg q(Y, TI_{vi}, TI_{vf}, T_{ti}, T_{tf}) \wedge \neg \iota * q(Y, TI_{vi}, TI_{vf}, Actual, FNC) \wedge \text{condicions}$$

Aquesta regla de transició es pot eliminar perquè no violarà mai ic , ja que en aquest cas si s'insereix p en temps de transacció d'inici *Actual*, mai no serà igual al temps de transacció d'inici de $\neg q$.

$$ic_4' \leftarrow \iota * p(X, T_{vi}, T_{vf}, Actual, FNC) \wedge \delta * q(Y, TI_{vi}, TI_{vf}, Actual, FNC) \wedge \text{condicions}$$

Aquesta regla de transició es pot donar perquè en aquest cas si s'insereix p en temps de transacció d'inici *Actual*, serà igual al temps de transacció d'inici de $\neg q$, que ara s'esborra amb un temps de transacció d'inici *Actual*. Ara, doncs, caldrà mirar si es compleixen les condicions de la definició per comprovar si és necessària aquesta regla de transició.

Cas 3. Suposem la següent restricció d'integritat en temps vàlid amb tots els predicats de la definició negats:

$$ic \leftarrow \neg p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg q(Y, TI_{vi}, TI_{vf}, T_{ti}, T_{tf}) \wedge \text{condicions}$$

Les regles de transició serien:

$$ic_2' \leftarrow \neg p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg \iota * p(X, T_{vi}, T_{vf}, \text{Actual}, \text{FNC}) \wedge \\ \delta * q(Y, TI_{vi}, TI_{vf}, \text{Actual}, \text{FNC}) \wedge \text{condicions}$$

Aquesta regla de transició es pot eliminar perquè no violarà mai ic , ja que en aquest cas si s'esborra q en temps de transacció d'inici *Actual*, mai no serà igual al temps de transacció d'inici de $\neg p$.

$$ic_3 \leftarrow \delta * p(X, T_{vi}, T_{vf}, \text{Actual}, \text{FNC}) \wedge \neg q(Y, TI_{vi}, TI_{vf}, T_{ti}, T_{tf}) \wedge \\ \neg \iota * q(Y, TI_{vi}, TI_{vf}, \text{Actual}, \text{FNC}) \wedge \text{condicions}$$

Aquesta regla de transició es pot eliminar perquè no violarà mai ic , ja que en aquest cas si s'esborra p en temps de transacció d'inici *Actual*, mai no serà igual al temps de transacció d'inici de $\neg q$.

$$ic_4 \leftarrow \delta * p(X, T_{vi}, T_{vf}, \text{Actual}, \text{FNC}) \wedge \delta * q(Y, TI_{vi}, TI_{vf}, \text{Actual}, \text{FNC}) \wedge \\ \text{condicions}$$

Aquesta regla de transició és l'única que en aquest cas pot violar ic , ja que en aquest cas si s'esborra p en temps de transacció d'inici *Actual*, serà igual al temps de transacció d'inici de $\neg q$, que també s'esborra amb temps de transacció d'inici *Actual*. Ara, doncs, caldrà mirar si es compleixen les condicions de la definició per comprovar si és necessària aquesta regla de transició.

6.9. Restriccions d'integritat bitemporals

Considerem la restricció d'integritat bitemporal:

$$ic4 \leftarrow signatures(C,P,T_{vi},T_{vf},T_{ti},T_{tf}) \wedge signatures(C,PI,TI_{vi},TI_{vf},TI_{ti},TI_{tf}) \\ \wedge PI < P \wedge T_{vf} < TI_{vi} \wedge T_{ti} < TI_{ti}-10$$

Les regles de transició temporals que obtenim després de reemplaçar els literals i distribuir \wedge sobre \vee són:

$$ic4' \leftarrow ic4_i' \quad i = 2, \dots, 4$$

$$ic4_2' \leftarrow signatures(C,P,T_{vi},T_{vf},T_{ti},T_{tf}) \wedge \\ \neg \delta * signatures(C,P,T_{vi},T_{vf},Actual,FNC) \wedge \\ \iota * signatures(C,PI,TI_{vi},TI_{vf},Actual,FNC) \wedge PI < P \wedge T_{vf} < TI_{vi} \wedge \\ T_{ti} < Actual-10$$

Aquesta regla de transició pot violar ic , ja que en aquest cas si s'insereix una signatura C en temps de transacció d'inici $Actual$, no serà igual al temps de transacció d'inici de l'assignatura C , que ja hi era.

$$ic4_3' \leftarrow \iota * signatures(C,P,T_{vi},T_{vf},Actual,FNC) \wedge \\ signatures(C,PI,TI_{vi},TI_{vf},TI_{ti},TI_{tf}) \wedge \\ \neg \delta * signatures(C,PI,TI_{vi},TI_{vf},Actual,FNC) \wedge PI < P \wedge T_{vf} < TI_{vi} \wedge \\ Actual < TI_{ti}-10$$

Aquesta regla de transició no es pot donar, ja que en aquest cas si s'insereix una signatura C en temps de transacció d'inici $Actual$, mai no serà més petit que el temps de transacció d'inici de l'assignatura C , que ja hi era.

$$ic4_4' \leftarrow \iota * signatures(C,P,T_{vi},T_{vf},Actual,FNC) \wedge \\ \iota * signatures(C,PI,TI_{vi},TI_{vf},Actual,FNC) \wedge PI < P \wedge T_{vf} < TI_{vi} \wedge \\ Actual < Actual-10$$

Aquesta regla de transició no es pot donar, ja que en aquest cas es tractaria d'una restricció d'integritat en temps vàlid perquè els temps de transacció de tota la restricció d'integritat serien iguals.

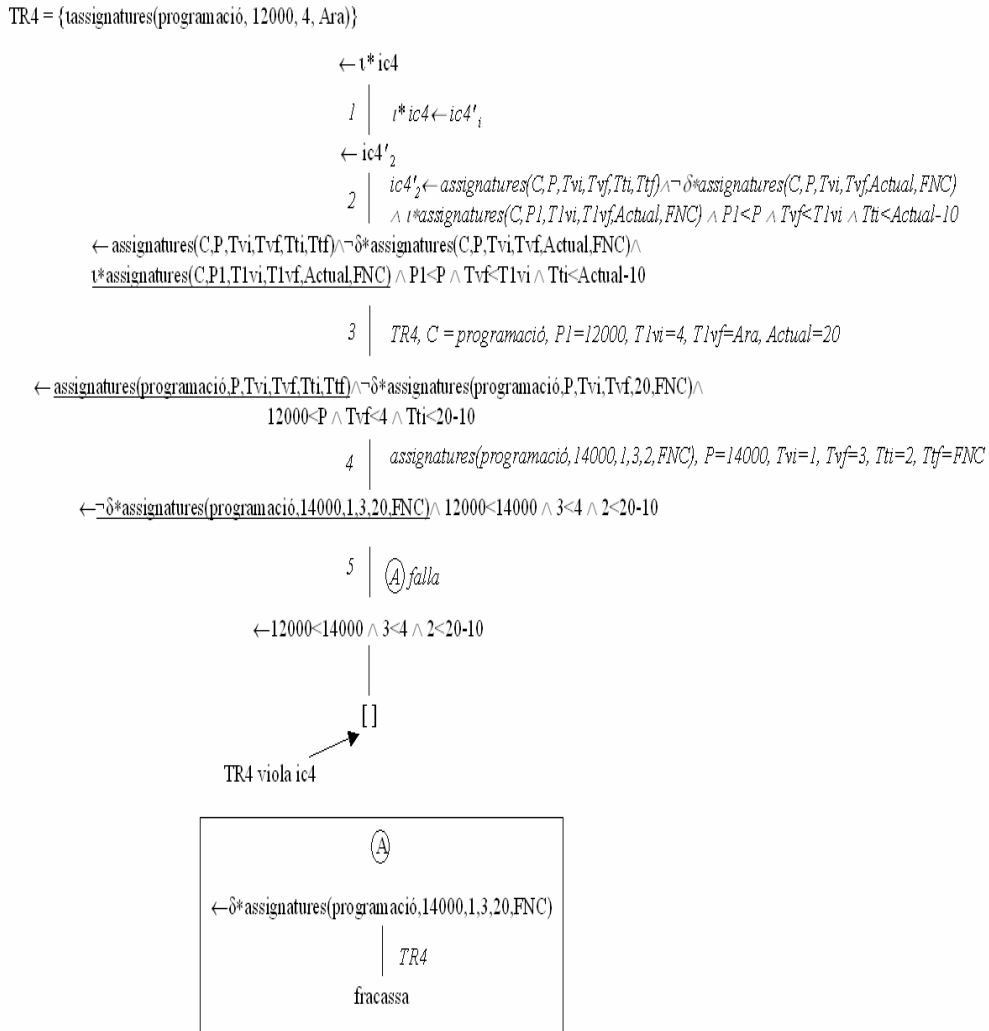


Figura 6.9. Violació d'ic4 per ic4₂'

Suposem una transacció TR4 en temps de transacció Actual 20 sobre la base de dades que hem presentat abans a la figura 6.4, suposant que també tenim: assignatures_r(programació, 14000, 1, 3, 2, FNC). TR4 és una transacció coherent i violarà ic4 per ic4₂'. A la figura 6.9 podem veure l'arbre de derivació de com s'aplicaria el nostre mètode de comprovació per detectar la violació d'ic4.

Els passos 1 i 2 són passos de la resolució SLDNF, on les regles de la base de dades augmentada actuen com a clàusules d'entrada.

En els passos 3 i 4 els predicats seleccionats fan referència a la transacció i a la base de dades, respectivament, i per tant per resoldre hem d'anar a instanciar a la transacció i a la base de dades, respectivament.

En el pas A, el literal seleccionat és: $\neg \delta^* \text{signatures}(\text{programació}, 14000, 1, 3, 20, \text{FNC})$. Per arribar a una derivació amb èxit, l'espai de cerca SLDNF ha de fracassar finitament per l'arbre subsidiari de: $\{\leftarrow \delta^* \text{signatures}(\text{programació}, 14000, 1, 3, 20, \text{FNC})\}$.

Tenint en compte que una restricció d'integritat bitemporal, tal com hem vist al capítol 5, pot tenir condicions, sincronismes en temps vàlid i en temps de transacció, però mai entre tots els predicats de la definició de la restricció d'integritat, periodicitats en temps vàlid i temps de transacció i dates límits en temps vàlid i temps de transacció, els temps vàlids i els temps de transacció són tots diferents. Analtzarem tots els casos possibles que es poden donar en les restriccions d'integritat bitemporals. Tractarem conjuntament les condicions, els sincronismes, les periodicitats i les dates límits, ja que són tots, en certa manera, diferents tipus de condicions que utilitzen o no variables temporals. En els diversos casos que analitzem, ho agruparem tot sota el nom de *condicions*. Suposarem sempre que T_{vi} és diferent de TI_{vi} i/o T_{vf} és diferent de TI_{vf} , i que T_{ti} és diferent de TI_{ti} i/o T_{tf} és diferent de TI_{tf} ; d'altra banda, *ic* seria una restricció d'integritat d'algun dels altres tipus que hem vist.

Cas 1. Suposem la següent restricció d'integritat bitemporal amb tots els predicats de la definició positius:

$$ic \leftarrow p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge q(Y, TI_{vi}, TI_{vf}, TI_{ti}, TI_{tf}) \wedge \text{condicions}$$

Les regles de transició temporals serien:

$$ic_2' \leftarrow p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg \delta^* p(X, T_{vi}, T_{vf}, \text{Actual}, \text{FNC}) \wedge \iota^* q(Y, TI_{vi}, TI_{vf}, \text{Actual}, \text{FNC}) \wedge \text{condicions}$$

Aquesta regla de transició pot violar *ic*, ja que en aquest cas si s'insereix *q* en temps de transacció d'inici *Actual*, no serà igual al temps de transacció d'inici de

p , que ja hi era. Ara, doncs, caldrà mirar si es compleixen les condicions de la definició per comprovar si és necessària aquesta regla de transició.

$$ic_3' \leftarrow \iota * p(X, T_{vi}, T_{vf}, Actual, FNC) \wedge q(Y, TI_{vi}, TI_{vf}, TI_{ti}, TI_{tf}) \wedge \neg \delta * q(Y, TI_{vi}, TI_{vf}, Actual, FNC) \wedge \text{condicions}$$

Aquesta regla de transició pot violar ic , ja que en aquest cas si s'insereix p en temps de transacció d'inici *Actual*, mai no serà igual al temps de transacció d'inici de q , que ja hi era. Ara, doncs, caldrà mirar si es compleixen les condicions de la definició per comprovar si és necessària aquesta regla de transició.

$$ic_4' \leftarrow \iota * p(X, T_{vi}, T_{vf}, Actual, FNC) \wedge \iota * q(Y, TI_{vi}, TI_{vf}, Actual, FNC) \wedge \text{condicions}$$

Aquesta regla de transició es pot eliminar, ja que en aquest cas si s'insereix p en temps de transacció d'inici *Actual*, serà igual al temps de transacció d'inici de q . I el mateix passarà amb el temps de transacció de finalització. Si es donés aquest cas, ic no hagués estat una restricció d'integritat bitemporal sinó que ho hagués estat en temps vàlid.

Cas 2. Suposem la següent restricció d'integritat bitemporal amb predicats positius i predicats negats en la definició:

$$ic \leftarrow p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg q(Y, TI_{vi}, TI_{vf}, TI_{ti}, TI_{tf}) \wedge \text{condicions}$$

Les regles de transició temporals serien:

$$ic_2' \leftarrow p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg \delta * p(X, T_{vi}, T_{vf}, Actual, FNC) \wedge \delta * q(Y, TI_{vi}, TI_{vf}, Actual, FNC) \wedge \text{condicions}$$

Aquesta regla de transició es pot donar perquè en aquest cas si s'esborra q , el temps de transacció d'inici de $\neg q$ serà *Actual* i mai serà igual al temps de transacció d'inici de p . Ara, doncs, caldrà mirar si es compleixen les condicions de la definició per comprovar si és necessària aquesta regla de transició.

$$ic_3' \leftarrow \iota * p(X, T_{vi}, T_{vf}, Actual, FNC) \wedge \neg q(Y, TI_{vi}, TI_{vf}, TI_{ti}, TI_{tf}) \wedge \neg \iota * q(Y, TI_{vi}, TI_{vf}, Actual, FNC) \wedge \text{condicions}$$

Aquesta regla de transició es pot donar perquè en aquest cas si s'insereix p en temps de transacció d'inici *Actual*, mai no serà igual al temps de transacció d'inici de $\neg q$. Ara, doncs, caldrà mirar si es compleixen les condicions de la definició per comprovar si és necessària aquesta regla de transició.

$$ic_4' \leftarrow \iota * p(X, T_{vi}, T_{vf}, Actual, FNC) \wedge \delta * q(Y, TI_{vi}, TI_{vf}, Actual, FNC) \wedge \text{condicions}$$

Aquesta regla de transició es pot eliminar perquè no violarà mai ic , ja que en aquest cas si s'insereix p en temps de transacció d'inici *Actual*, serà igual al temps de transacció d'inici de $\neg q$, que ara s'esborra amb un temps de transacció d'inici *Actual*. I el mateix passarà amb el temps de transacció de finalització. Si es donés aquest cas, ic no hagués estat una restricció d'integritat bitemporal sinó que ho hagués estat en temps vàlid.

Cas 3. Suposem la següent restricció d'integritat bitemporal amb tots els predicats de la definició negats:

$$ic \leftarrow \neg p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg q(Y, TI_{vi}, TI_{vf}, TI_{ti}, TI_{tf}) \wedge \text{condicions}$$

Les regles de transició temporals serien:

$$ic_2' \leftarrow \neg p(X, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg \iota * p(X, T_{vi}, T_{vf}, Actual, FNC) \wedge \delta * q(Y, TI_{vi}, TI_{vf}, Actual, FNC) \wedge \text{condicions}$$

Aquesta regla de transició es pot donar perquè en aquest cas si s'esborra q en temps de transacció d'inici *Actual*, mai no serà igual al temps de transacció d'inici de $\neg p$. Ara, doncs, caldrà mirar si es compleixen les condicions de la definició per comprovar si és necessària aquesta regla de transició.

$$ic_3' \leftarrow \delta * p(X, T_{vi}, T_{vf}, Actual, FNC) \wedge \neg q(Y, TI_{vi}, TI_{vf}, TI_{ti}, TI_{tf}) \wedge \neg \iota * q(Y, TI_{vi}, TI_{vf}, Actual, FNC) \wedge \text{condicions}$$

Aquesta regla de transició es pot donar perquè en aquest cas si s'esborra p en temps de transacció d'inici *Actual*, mai no serà igual al temps de transacció d'inici de $\neg q$. Ara, doncs, caldrà mirar si es compleixen les condicions de la definició per comprovar si és necessària aquesta regla de transició.

$$ic_4' \leftarrow \delta * p(X, T_{vi}, T_{vf}, Actual, FNC) \wedge \delta * q(Y, Tl_{vi}, Tl_{vf}, Actual, FNC) \wedge \text{condicions.}$$

Aquesta regla de transició no es pot donar, ja que en aquest cas si s'esborra p en temps de transacció d'inici *Actual*, serà igual al temps de transacció d'inici de $\neg q$. I el mateix passarà amb el temps de transacció de finalització. Si es donés aquest cas, ic no hagués estat una restricció d'integritat bitemporal sinó que ho hagués estat en temps vàlid

6.10. Simplificacions de les regles de transició temporals

En analitzar en profunditat els diferents tipus de restriccions d'integritat temporals que es poden donar en bd-dbts, hem obtingut els resultats que es presenten a la figura 6.10.

Restriccions d'Integritat	Primer grup actuals	Segon grup en temps de transacció	Tercer grup en temps vàlid	Quart grup bitemporals
Cas 1: Predicats positius	ic4	ic2, ic3	ic4	ic2, ic3
Cas 2: Predicats positius i negats	ic4	ic2, ic3	ic4	ic2, ic3
Cas 3: Predicats negats	ic4	ic2, ic3	ic4	ic2, ic3

Figura 6.10. Quadre resum de les simplificacions obtingudes per regles d'integritat de dos literals en el seu cos

En el quadre que es mostra a la figura 6.10 es tenen en compte restriccions d'integritat amb només dos literals en el seu cos i, per tant, tres possibles regles de transició: $ic2$, $ic3$ i $ic4$.

Si estenguéssim el quadre de la figura 6.10 a restriccions d'integritat amb més de dos literals, en el primer i el tercer grup sempre tindriem només la darrera regla de transició, i en el segon i el quart grup totes les regles de transició menys la darrera, com es pot veure a la figura 6.11.

Restriccions d'Integritat	Primer grup actuals	Segon grup en temps de transacció	Tercer grup en temps vàlid	Quart grup bitemporals
Cas 1: Predicats positius	Darrera regla	Totes les regles menys la darrera	Darrera regla	Totes les regles menys la darrera
Cas 2: Predicats positius i negats	Darrera regla	Totes les regles menys la darrera	Darrera regla	Totes les regles menys la darrera
Cas 3: Predicats negats	Darrera regla	Totes les regles menys la darrera	Darrera regla	Totes les regles menys la darrera

Figura 6.11. Quadre resum general de les simplifikacions obtingudes per regles d'integritat

De tota manera, fins i tot aquest quadre pot simplifikar-se molt més, estudiant les condicions de les regles de transició, com hem anat veient en els exemples concrets de cadascun dels nostres grups de restriccions d'integritat. Els casos més bàsics de regles de transició que es poden eliminar són les que tenen condicions d'algun dels següents tipus per temps de transacció:

Actual < Actual, FNC < T, FNC < FNC.

O per temps vàlid:

Ara < T, Ara < Ara.

En la versió del nostre mètode presentada a [MS96b] havíem obtingut altres tipus de simplifikacions que ara no són aplicables perquè les assumpcions que vam prendre en el seu moment han canviat per ampliar el nombre de tipus de restriccions d'integritat que pugui definir el dissenyador.

6.11. Regles de transició després d'aplicar el mètode de descomposició de restriccions d'integritat

En l'apartat 5.4.3 del capítol anterior hem explicat com analitzar les condicions de les restriccions d'integritat per obtenir una descomposició amb una part d'un grup i una altra part d'un altre. Aquesta descomposició ens afegeix més avantatges a les ja obtingudes a l'apartat 6.10. Concretament, els arbres de derivació de les regles de transició de la part d'una restricció d'integritat actual o en temps vàlid no fan cap accés a la bd-dbt i, per tant, farem que sempre siguin les primeres en comprovar-se. De manera que, si els arbres de derivació d'aquestes regles de transició violen la restricció d'integritat, ja no caldrà

continuar fent resolució SLDNF per les altres regles de transició. Veiem un exemple de la utilitat del mètode de descomposició.

Considerem la restricció d'integritat en temps de transacció:

$$ic5 \leftarrow matricula(S,C,T_{vi},T_{vf},T_{ti},T_{tf}) \wedge \neg curs(C,T_{vi},T_{vf},TI_{ti},TI_{tf}) \wedge TI_{tf} \leq T_{tf} \wedge T_{ti} < TI_{tf}$$

Ic5 és una restricció d'integritat amb una part actual i una part en temps de transacció. Descomponguem-la:

$$ic5 \leftarrow ic5_i \quad i = 1, \dots, 3$$

Ic5 actual:

$$ic5_3 \leftarrow matricula(S,C,T_{vi},T_{vf},T_{ti},T_{tf}) \wedge \neg curs(C,T_{vi},T_{vf},T_{ti},T_{tf})$$

Ic5 en temps de transacció:

$$ic5_1 \leftarrow matricula(S,C,T_{vi},T_{vf},T_{ti},T_{tf}) \wedge \neg curs(C,T_{vi},T_{vf},TI_{ti},TI_{tf}) \wedge TI_{tf} < T_{tf} \wedge T_{ti} < TI_{tf}$$

$$ic5_2 \leftarrow matricula(S,C,T_{vi},T_{vf},T_{ti},T_{tf}) \wedge \neg curs(C,T_{vi},T_{vf},TI_{ti},T_{tf}) \wedge T_{ti} < T_{tf}$$

Les regles de transició que obtenim després de reemplaçar els literals i distribuir \wedge sobre \vee són:

Ic5 actual:

$$ic5_3' \leftarrow ic5_{3i}' \quad i = 2, \dots, 4$$

$$ic5_{32}' \leftarrow matricula(S,C,T_{vi},T_{vf},T_{ti},T_{tf}) \wedge \neg \delta * matricula(S,C,T_{vi},T_{vf},Actual,FNC) \wedge \delta * curs(C,T_{vi},T_{vf},Actual,FNC)$$

$$ic5_{33}' \leftarrow \iota * matricula(S,C,T_{vi},T_{vf},Actual,FNC) \wedge \neg curs(C,T_{vi},T_{vf},T_{ti},T_{tf}) \wedge \neg \iota * curs(C,T_{vi},T_{vf},Actual,FNC)$$

De fet, no hagués calgut calcular $ic5_{32}'$ i $ic5_{33}'$, perquè, com hem vist a l'apartat 6.7, mai no es donarà cap cas en què és violi *ic5* actual per aquestes

regles de transició. Aquestes regles no s'arriben a generar.

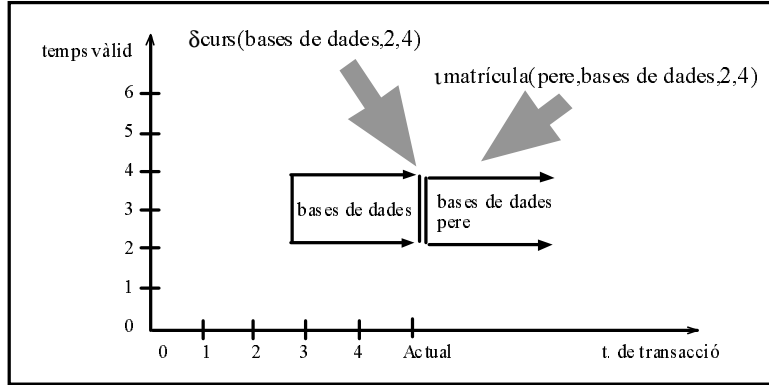


Figura 6.12. Violació d'ic5 per ic5₃₄'

$$ic5_{34}' \leftarrow \iota * matricula(S, C, T_{vi}, T_{vf}, Actual, FNC) \wedge \delta * curs(C, T_{vi}, T_{vf}, Actual, FNC)$$

Aquesta darrera regla de transició, en canvi, es podrà donar perquè si volem inserir una matrícula en temps de transacció d'inici *Actual*, però hi havia aquest curs ofert des de fa un temps de transacció d'inici donat i ara s'esborra amb un temps de transacció d'inici *Actual*, aleshores els temps de transacció d'inici són iguals i els de finalització poden ser-ho també.

Ic5 en temps de transacció:

$$ic5_i' \leftarrow ic5_{1i}' \quad i = 2, 3$$

$$ic5_{12}' \leftarrow matricula(S, C, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg \delta * matricula(S, C, T_{vi}, T_{vf}, Actual, FNC) \wedge \delta * curs(C, T_{vi}, T_{vf}, Actual, FNC) \wedge FNC < T_{tf} \wedge T_{ti} < FNC$$

Aquesta regla de transició no es podrà donar perquè si volem esborrar un curs en temps de transacció *Actual*, voldrà dir que el temps de transacció d'inici serà *Actual*. Tot i així si hi havia matrícules en aquest curs que no s'esborren en temps de transacció *Actual*, aleshores T_{tf} seria *FNC* i, per tant, $T_{ti} < Actual$, però sens dubte *FNC* no seria més petit que T_{tf} , que també seria *FNC*.

$$ic5_{13}' \leftarrow \iota * matricula(S, C, T_{vi}, T_{vf}, Actual, FNC) \wedge \neg curs(C, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg \iota * curs(C, T_{vi}, T_{vf}, Actual, FNC) \wedge T_{tf} < FNC \wedge Actual < T_{tf}$$

Aquesta regla de transició no es podrà donar mai perquè si volem inserir una matrícula en temps de transacció d'inici *Actual*, però no hi havia aquest curs ofert des de fa un temps de transacció d'inici donat T_{ti} , aleshores no pot ser que $T_{ti} < FNC$ i al mateix temps $Actual < T_{ti}$.

$$ic5_{14}' \leftarrow \neg *matricula(S, C, T_{vi}, T_{vf}, Actual, FNC) \wedge \delta *curs(C, T_{vi}, T_{vf}, Actual, FNC) \wedge FNC < FNC \wedge Actual < FNC$$

Aquesta regla de transició mai no es podrà donar perquè si volem inserir una matrícula en temps de transacció d'inici *Actual*, però hi havia aquest curs ofert des de fa un temps de transacció d'inici donat i ara s'esborra amb un temps de transacció d'inici *Actual*, aleshores no pot ser que $FNC < FNC$. De fet, aquesta regla de transició ja no s'hauria generat d'entrada.

$$ic5_{2'} \leftarrow ic5_{2i'} \quad i = 2, \dots, 4$$

$$ic5_{22}' \leftarrow matricula(S, C, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg \delta *matricula(S, C, T_{vi}, T_{vf}, Actual, FNC) \wedge \delta *curs(C, T_{vi}, T_{vf}, Actual, FNC) \wedge T_{ti} < FNC$$

Aquesta regla de transició es podrà donar perquè si volem esborrar un curs en temps de transacció d'inici *Actual*, però hi havia matrícules en aquest curs que no s'esborren en temps de transacció *Actual*, aleshores els temps de transacció de finalització poden ser iguals i, com segur que $T_{ti} < FNC$, aleshores estariem violant *ic5*.

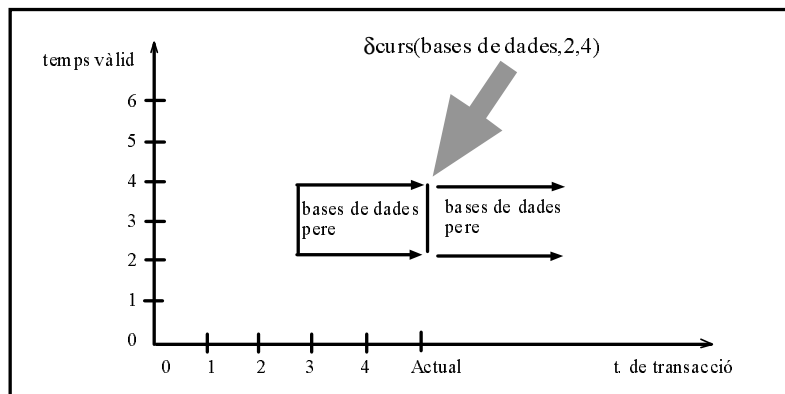


Figura 6.13. Violació d'*ic5* per $ic5_{22}'$

$$ic5_{23}' \leftarrow \iota * matricula(S, C, T_{vi}, T_{vf}, Actual, FNC) \wedge \neg curs(C, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg \iota * curs(C, T_{vi}, T_{vf}, Actual, FNC) \wedge Actual < T_{tf}$$

Aquesta regla de transició es podrà donar perquè si volem inserir una matrícula en temps de transacció d'inici *Actual*, però no hi havia aquest curs ofert des de fa un temps de transacció d'inici donat T_{ti} i els temps de transacció de finalització són iguals, aleshores pot ser que $Actual < T_{tf}$.

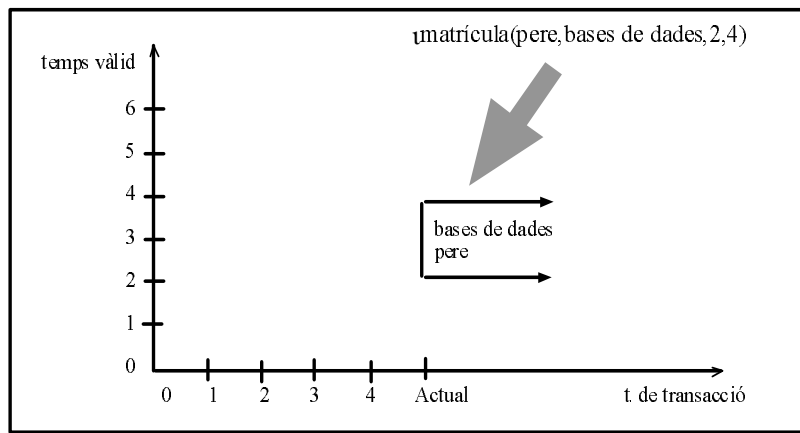


Figura 6.14. Violació d'ic5 per ic5₂₃'

$$ic5_{24}' \leftarrow \iota * matricula(S, C, T_{vi}, T_{vf}, Actual, FNC) \wedge \delta * curs(C, T_{vi}, T_{vf}, Actual, FNC) \wedge Actual < FNC$$

Aquesta regla de transició, de fet, ja no s'hauria generat d'entrada.

6.12. Restriccions d'integritat amb predicats derivats

Per simplificar, fins ara, només hem desenvolupat el mètode per restriccions d'integritat sense cap predicat derivat en la definició. Veiem ara el comportament d'una restricció d'integritat amb predicats derivats a la definició.

Considerem la part actual de la restricció d'integritat *ic5*:

$$ic5_3 \leftarrow matricula(S, C, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg curs(C, T_{vi}, T_{vf}, T_{ti}, T_{tf})$$

Amb la regla de deducció:

$$curs(C, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \leftarrow assignatures(C, P, T_{vi}, T_{vf}, T_{ti}, T_{tf})$$

Les regles de transició que obtenim després de tenir en compte les simplificacions anteriors i de reemplaçar els literals i distribuir \wedge sobre \vee són:

$$ic5_3' \leftarrow ic5_{34}'$$

$$ic5_{34}' \leftarrow \iota * matricula(S, C, T_{vi}, T_{vf}, Actual, FNC) \wedge \delta * curs(C, T_{vi}, T_{vf}, Actual, FNC)$$

En aquesta regla de transició apareix una regla d'esdeveniments d'esborrat:

$$\delta * curs(C, T_{vi}, T_{vf}, Actual, FNC) \leftarrow \delta * assignatures(C, P, T_{vi}, T_{vf}, Actual, FNC) \wedge \neg curs'(C, T_{vi}, T_{vf}, T_{ti}, T_{tf})$$

on:

$$curs_1'(C, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \leftarrow curs(C, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \wedge \neg \delta * curs(C, T_{vi}, T_{vf}, Actual, FNC)$$

$$curs_2'(C, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \leftarrow \iota * curs(C, T_{vi}, T_{vf}, Actual, FNC)$$

$$curs'(C, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \leftarrow curs_1'(C, T_{vi}, T_{vf}, T_{ti}, T_{tf})$$

$$curs'(C, T_{vi}, T_{vf}, T_{ti}, T_{tf}) \leftarrow curs_2'(C, T_{vi}, T_{vf}, T_{ti}, T_{tf})$$

Suposem una transacció *TR5* en temps de transacció *Actual 4* sobre la base de dades que hem presentat a la figura 6.4, però ara amb els fets següents:

$$\begin{aligned} & assignatures_r(logica, 14000, 1, 3, 4, FNC) \\ & estudiants_r(jordi, 1, 3, 4, FNC) \end{aligned}$$

TR5 és una transacció coherent i violarà *ic5₃* per *ic5₃₄'*. A la figura 6.15 podem veure l'arbre de derivació de com s'aplicaria el nostre mètode de comprovació per detectar la violació d'*ic5₃*.

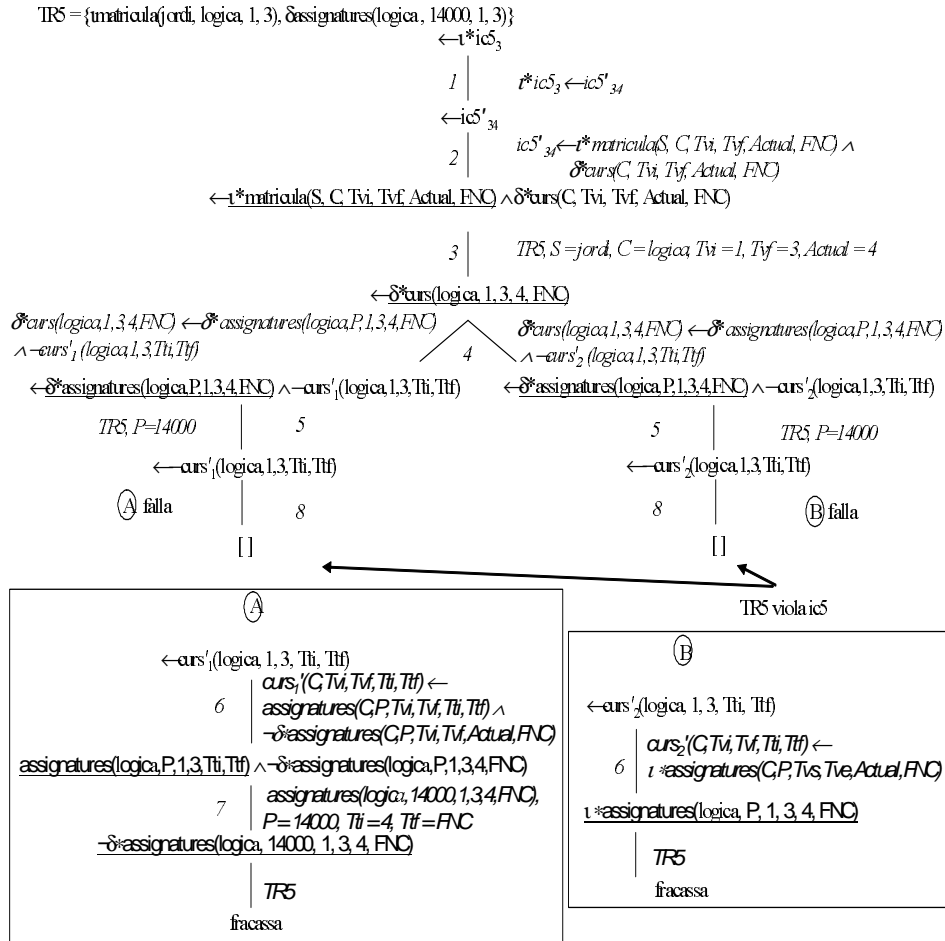


Figura 6.15. Violació d' $ic5_3$ per $ic5'_{34}$

Els passos 1, 2, 4 i 6 són passos de la resolució SLDNF, on les regles de la base de dades augmentada actuen com a clàusules d'entrada.

En els números 3 i 5, els predicats seleccionats fan referència a la transacció i per resoldre hem d'anar a instanciar a la transacció. En el número 7, els predicats seleccionats fan referència a la base de dades i per resoldre hem d'anar a instanciar a la base de dades. En el número 8, utilitzem arbres subsidiaris per continuar la derivació.

6.13. Comparació amb altres mètodes

Molts pocs mètodes de comprovació de restriccions d'integritat en bd-dbts incorporen el temps de forma explícita, com es pot veure en la bibliografia d'aquesta àrea de [WJW98].

Hi ha mètodes, com el mètode de Chomicki [Cho92], [Cho95] o el de Wüthrich [Wüt91], que utilitzen la lògica temporal per formular les restriccions d'integritat. Però en emprar la lògica temporal, aquests mètodes no contenen informació temporal explícita com en el nostre cas.

Podríem haver escollit el llenguatge *Event Calculus* basat en la lògica, com han fet [KS86], [SK95], [Sri95] o [VDD95], per expressar les restriccions d'integritat temporals, peròensem que aquesta elecció no és important, considerant que l'objectiu d'aquest treball és un nou mètode de comprovació de bd-dbts independent del llenguatge.

El mètode de Plexousakis [Ple93], [Ple96] formula les restriccions d'integritat en un llenguatge de representació del coneixement, lligat a Telos [MBJ+90]. Telos adopta el model de temps basat en els intervals d'Allen [All83] per representar la informació temporal. El mètode consisteix en generar una estructura simplificada parametritzada per a cada literal de les restriccions d'integritat i les regles de deducció que es veuen afectades per una transacció, i necessita construir un graf de dependència per comprovar si es violen les restriccions d'integritat. El mètode per trobar la forma simplificada és una extensió del mètode de Nicolas [Nic82], que inclou tractament temporal.

Resumint, l'únic mètode amb tractament explícit de temps en bd-dbts que hem trobat és el mètode de Plexousakis que es concentra en la simplificació de restriccions d'integritat temporals amb un gran nombre dels operadors temporals d'Allen existents entre els intervals. En canvi, en el nostre cas treballem amb punts de temps i ni tan sols utilitzem els operadors temporals d'Allen. Hi ha una gran diferència entre treballar amb punts de temps o amb intervals de temps com es pot veure a [BBJ98]. A més, el mètode de Plexousakis necessita un metaintèrpret del llenguatge de Telos per parametritzar l'estructura simplificada, mentre que nosaltres utilitzem la resolució SLDNF directament proveïda des d'un sistema Prolog. Aquestes importants diferències fan difícilment comparable aquest mètode amb el presentat en aquest treball.

7. Termes dependents temporalment

En el capítol anterior hem presentat un mètode de comprovació de restriccions d'integritat per a bases de dades deductives amb dues dimensions de temps. El temps de transacció és el temps en què un fet s'emmagatzema a la base de dades. Això fa que en la dimensió de temps de transacció no ens puguem referir al futur i el passat només es pugui consultar, però no modificar. Aquestes limitacions de la semàntica del temps de transacció han estat aprofitades per reduir considerablement les regles de transició del nostre mètode. En canvi, respecte a la dimensió de temps vàlid, hem comprovat que, com que és un temps en què un fet és cert en la realitat modelada, les seves característiques especials fan que qualsevol cosa es pugui donar en aquesta dimensió de temps. És per aquest motiu que destinarem un capítol sencer a analitzar amb més profunditat el comportament dels termes dels predicats d'una base de dades respecte al temps vàlid.

L'exploració del comportament temporal dels termes dels predicats d'una base de dades ens ha portat a identificar dos tipus diferents de termes: els termes independents del temps i els termes dependents del temps. Diem que un terme és independent del temps si té un valor específic per un interval de temps i aquest valor no canvia en cap dels possibles subinterval·ls d'aquest interval de temps. Normalment, els termes són independents del temps. Per exemple, si una persona es diu *Jordi* durant tota la seva vida, també es diu *Jordi* durant qualsevol subinterval de la seva vida.

No obstant això, en altres casos, hem detectat que la informació d'alguns termes durant un interval de temps pot ser diferent de la informació que voldríem obtenir per als seus subintervals. Diem que un terme és *dependent del temps* si té un valor específic per un interval de temps, però aquest valor pot ser diferent en els subintervals d'aquest interval de temps. Presentarem un exemple on mostrarem el concepte de *terme dependent del temps*.

Suposem que volem obtenir la quantitat d'aigua que va ploure ahir entre les 9 i les 14 i entre les 15 i les 21 a partir de la informació dels predicats de les bases de dades de tres estacions meteorològiques diferents ($M1$, $M2$ i $M3$). Tal com podem veure a la figura 7.1, les estacions meteorològiques proveeixen informació diferent i fins i tot incompleta, cosa que realment fa molt difícil –o ens podem atrevir a dir que impossible– obtenir la informació en què estem interessats.

M1:	Qtt	Temps	M2:	Qtt	Temps	M3:	Qtt	Temps
	30	[9,12)		20	[9,11)		50	[9,12)
	60	[18,22)		30	[18,22)		20	[12,16)
							20	[16,20)

Figura 7.1. Informació proveïda per tres bases de dades d'estacions meteorològiques diferents

Per poder obtenir la quantitat de pluja entre les 9 i les 14 i entre les 15 i les 21, la idea que proposem és completar la informació d' $M1$, $M2$ i $M3$. El primer pas per obtenir aquesta quantitat és cobrir el que anomenarem *forats d'informació*. Per exemple, a $M1$ veiem que tenim que entre les 9 i les 11 van caure 30 litres d'aigua i entre les 18 i les 21 van caure 60 litres, però no tenim informació sobre què va passar entre les 12 i les 17. És en aquest cas que diem que entre les 12 i les 17 hem detectat un forat d'informació. A la figura 7.2 podem veure que el forat d' $M1$ el cobrim amb na , el forat d' $M2$ amb nb i el forat d' $M3$ és cobert amb nc . Quan ja no tinguem forats d'informació a la base de dades, el segon pas per completar-la serà calcular els nous valors per a tots els seus subintervals. A la figura 7.2 mostrem la idea de les bases de dades completades d' $M1$, $M2$ i $M3$ amb els valors calculats per a tots els subintervals. Com que en aquest capítol proposarem diverses opcions per cobrir forats d'informació i calcular aquests valors, de moment hem utilitzat les lletres a_i , a'_i , b_i , b'_i , c_i , c'_i , na_i , nb_i i nc_i per denotar aquests valors, on els subíndex i estaran entre 1 i el nombre màxim de subintervals que cal generar per cada interval.

M1 ^{comp} :	Qtt	Temps	M2 ^{comp} :	Qtt	Temps	M3 ^{comp} :	Qtt	Temps
	30	[9,12)		20	[9,11)		50	[9,12)
	a ₁	[9,10)		b ₁	[9,10)		c ₁	[9,10)
	a ₂	[10,11)		b ₂	[10,11)		...	
	a ₃	[11,12)		nb	[11,18)		c ₅	[10,12)
	a ₄	[9,11)		nb ₁	[11,12)		40	[12,20)
	a ₅	[10,12)		...			c' ₁	[12,13)
	na	[12,18)		nb ₂₇	[12,18)		...	
	na ₁	[12,13)		30	[18,22)		c' ₃₅	[13,20)
	...			b' ₁	[18,19)		nc	[20,22)
	na ₂₀	[13,18)		...			nc ₁	[20,21)
	60	[18,22)		b' ₉	[19,22)		nc ₂	[21,22)
	a' ₁	[18,19)						
	...							
	a' ₉	[19,22)						

Figura 7.2. Les bases de dades completades de les tres estacions meteorològiques

A la figura 7.2 podem veure que en el cas d' $M3^{comp}$ hem calculat els valors per a dos intervals i un forat d'informació, mentre que a $M3$ teníem tres intervals adjacents i un forat d'informació. Aquest fet és degut a la decisió d'aplicar fusió en alguns casos. Els primers dos intervals adjacents d' $M3$ ($[9,12)$ i $[12,16)$) no tenen la mateixa informació no temporal i per tant no podem aplicar-hi fusió. Hem de fer notar que, amb termes dependents temporalment, per detectar si dos intervals tenen la mateixa informació no temporal, és imprescindible tenir en compte la funció que s'aplicarà per derivar els valors dels subinterval·ls. Els altres dos intervals adjacents d' $M3$ ($[12,16)$ i $[16,20)$) tenen exactament la mateixa informació no temporal i considerem que en tots dos casos s'aplica una funció, que en aquest cas és la mateixa, amb la qual cosa decidim aplicar una variant de la fusió descrita en el capítol 4 que tinguin en compte l'addició d'informació de termes dependents temporalment.

Quan treballem amb termes dependents temporalment, la idea teòrica és obtenir una *base de dades completada* que determini informació per a tots i cada un dels subinterval·ls de la base de dades. Com hem explicat abans, per arribar a la base de dades completada, normalment el primer pas és detectar els forats d'informació existents i cobrir-los, per obtenir el que anomenem una *base de*

dades coberta. Un cop fet això, ja només ens caldrà generar la informació per a tots els subintervalls de la base de dades coberta. Malauradament, aquesta idea té un cost molt elevat, de manera que en aquest capítol explicarem com podem arribar a obtenir la informació que cerquem sense generar necessàriament els valors per a tots els subintervalls de temps. A la figura 7.3 podem veure el que anomenarem *informació rellevant*, que, com peces d'un puzzle, són exactament els subintervalls que necessitem per donar solució a la consulta de publicar la informació meteorològica d'ahir entre les 9 i les 14, i entre les 15 i les 21. Com que només volem saber la informació entre les 9 i les 14 i entre les 15 i les 21, de la informació rellevant obtinguda sumarem els valors de cada base de dades entre les 9 i les 14, i entre les 15 i les 21. Un cop fet això, ja només ens quedarà aplicar una funció que calculi la mitjana per arribar a la informació que estàvem cercant. Al llarg d'aquest capítol anirem formalitzant els conceptes que hem introduït amb aquest exemple.

M1 ^{comp} :	Qtt	Temps	M2 ^{comp} :	Qtt	Temps	M3 ^{comp} :	Qtt	Temps
30		[9,12)	20		[9,11)	50		[9,12)
na ₁₂		[12,15)	nb ₁₉		[11,15)	c' ₁₆		[12,15)
na ₁₅		[15,18)	nb ₁₈		[15,18)	c' ₃₀		[15,20)
60		[18,22)	30		[18,22)	nc		[20,22)

Figura 7.3. Informació rellevant de les bases de dades completades d'M1, M2 i M3

7.1. Comparació amb el treball previ relacionat

No hem estat els primers a tractar un tema força proper al raonament sobre canvis tenint en compte el temps. Hem trobat un bon nombre de recerques relacionades amb aquest tema, algunes de les quals fins i tot poden ser profitosament incorporades al nostre treball. Començarem fent referència a un interessant i recent treball sobre la semàntica de les dades temporals seguint enfocaments de les àrees de lingüística i intel·ligència artificial. A [TS04] es presenten dos tipus de fets, *tèlics* i *atèlics*, i s'emfasitza que la semàntica de l'associació entre els fets i el temps depèn del tipus de fet que s'estigui considerant. Un exemple d'un fet atèlic podria ser el sou d'un empleat de 2.000 euros des d'1 fins a 30. Un exemple d'un fet tèlic seria que en *Jordi* telefonés a la *Maria* des d'1 fins a 15 amb un cost de 2 euros i des de 16 fins a 30 amb un

cost de 3 euros. La diferència entre els dos fets anteriors, tal com també es planteja a [BBJ98], es basa principalment en la manera com s'aplicarà l'operació de fusió [BSS96] sobre aquests fets. Mentre que sobre els fets atèlics es pot aplicar l'operació de fusió, un fet tèlic es considera primitiu i no descomponible, la qual cosa voldria dir que no podríem pas aplicar l'operació de fusió entre les dues trucades telefòniques de l'exemple anterior, ja que es tractaria de trucades diferents. Un altre exemple d'un fet tèlic és que en *Jordi* construeixi una casa entre 1 i 30. En aquest altre exemple, el problema és més que l'aplicació o no de l'operació de fusió, ja que en aquest cas no podem trobar valors per als subintervalls entre 1 i 30. De tota manera, aquest segon exemple de fet tèlic sol ser considerat en l'àmbit de la intel·ligència artificial, però en l'àmbit de les bases de dades normalment trobem fets atèlics o fets tèlics per als quals podem derivar valors en els seus subintervalls. Per exemple, tant en el cas del sou com de les trucades telefòniques podem derivar informació per a tots els subintervalls entre 1 i 30. Nosaltes, tal com es proposa també a [TS04], suposarem fets atèlics o fets tèlics per als quals podem derivar valors, tenint present a més que la nostra anàlisi es concentra en la dependència temporal dels termes d'un predicat, més que en la dependència del temps dels fets en conjunt, tal com anirem explicant en les properes seccions.

Normalment el primer pas per obtenir una base de dades completada és el cobriment dels forats d'informació. En raonament temporal, s'han proposat diversos enfocaments sobre aquest tema, com per exemple [All84], [McD82] i [Sho88]. Concretament, a [Sho88] es defineix el concepte de *persistència* com l'assumpció que alguna cosa no canviarà fins que no s'hi faci alguna acció que la canviï. La utilització de la informació que hi ha a l'esquerra del forat pot ser una idea interessant per cobrir forats d'informació. També es podria considerar cobrir el forat amb la informació de la seva dreta. I fins i tot altres possibilitats podrien ser la utilització de funcions d'interpolació com les que es proposen a [SS87], [JS96] i [BJW00].

En bases de dades amb múltiples granularitats de temps [DEL+00] s'explica que les granularitats temporals i la indeterminació no són més que dues cares de la mateixa moneda. Nosaltes estem completament d'acord amb aquesta afirmació i, com hem comentat en capítols anteriors, assumirem una única granularitat de temps. Per tant, encara que l'estudi de treballar amb granularitats diferents queda totalment fora de l'àmbit del nostre treball, hem trobat algunes aportacions que podrem tenir en compte, com per exemple a [BJW00]. A

[DS98] la informació temporal indeterminada es defineix com que sabem segur que un fet que volem emmagatzemar a la base de dades és cert, però no sabem quan s'ha esdevingut. En canvi, els nostres termes dependents temporalment volen dir que sabem segur que un fet que volem emmagatzemar a la base de dades és cert, sabem exactament quan s'ha esdevingut, però el que no sabem és el seu valor. Treballs desenvolupats en l'àrea d'informació temporal indeterminada són, per exemple, [CP00] i [GM01]. Concretament, a [GM01] s'introdueix la idea d'utilitzar probabilitats, que podria ser molt interessant també en el nostre cas com a treball futur.

El segon pas per obtenir una base de dades completada és la derivació de valors per a tots els subinterval·ls de temps. Novament, un ben conegut concepte de raonament temporal explicat a [Sho88], que és el concepte de *liquiditat*, pot ser aplicat als nostres termes independents temporalment. El concepte de *liquiditat* es dóna quan es compleixen l'herència descendent i l'herència ascendent al mateix temps. L'*herència descendent* es dóna quan una cosa que es compleix en un interval es compleix també en tots els seus subinterval·ls. L'*herència ascendent* es dóna quan una cosa que es compleix en tots els subinterval·ls d'un interval també es compleix en el mateix interval. No obstant això, els nostres termes dependents temporalment no són exactament líquids, en el sentit que no es pot utilitzar el mateix valor per heretar-lo ni ascendentment ni descendentment.

En l'àrea de raonament i representació del coneixement, l'addició de la noció de granularitat de temps en els sistemes basats en el coneixement ha estat una de les solucions més efectives al problema de la gestió de fenòmens periòdics; en podem trobar un exemple a [NS92]. En l'àrea de bases de dades, les granularitats de temps apareixen principalment com una eina formal per gestionar les característiques intrínseques dels calendaris; en seria un exemple [BJW00]. A [BJW00], els conceptes de *persistència*, *herència descendent* i *herència ascendent* són aplicats a atributs per aconseguir derivar informació per a diferents grànuls de temps. A [BJW00] es presenten dos tipus d'assumpcions: la basada en punts i la basada en interval·ls de temps. L'assumpció basada en punts de temps deriva valors implícits a partir de valors explícits, ja que les informacions temporals només estan emmagatzemades en un punt de temps i han de fer possible deduir informació per a la resta de punts. L'exemple més clar d'aquesta assumpció és l'aplicació de la persistència, ja que, si considerem que una cosa és certa fins que no hi hagi un nou canvi, podem deduir informació a

partir d'un cert punt fins a arribar al punt on apareix el canvi. L'assumpció basada en intervals és capaç de convertir informació per a un grànul de la informació d'altres grànuls de granularitat diferent. Aquesta assumpció no és res més que l'aplicació de les propietats d'herència descendent, herència ascendent i liquiditat. Les assumpcions basades en punts de temps i en intervals de temps es poden combinar. Encara que nosaltres no treballem amb granularitats diferents, podem aprofitar gran part d'aquesta teoria per a la informació independent temporalment, ja que existeixen força similituds, però no pas per a la informació dependent temporalment. En aquest cas, el nostre problema és més un problema semàntic que no pas de conversió, en què del que es tracta és d'analitzar el comportament dels atributs i decidir quina significació té el que trobem dins d'un interval de temps per a qualsevol subinterval de temps.

Les bases de dades completades requereixen un conjunt d'operacions adequat per manipular-les. En bases de dades temporals, [LM97] proposa el llenguatge *IXSQL* per a la gestió d'intervals amb la utilització de les operacions *fold* i *unfold*. L'operació *unfold* canvia cada tuple d'una relació respecte a un atribut A_i per un conjunt de tuples que contenen només punts de temps per A_i . L'operació *fold* agafa parells de tuples respecte a un atribut A_i , amb els mateixos valors en els atributs diferents de A_i , i els fusiona, i es repeteix aquest procediment fins que no quedi cap parell de tuples per fusionar. El problema d'aquestes operacions, com els mateixos autors comenten, és que el cost de la seva implementació és prohibitiu. El nostre enfocament millora considerablement la proposta de [LM97], ja que nosaltres no hem de generar informació per a tots els punts de temps, sinó només per als subintervals que nosaltres hem anomenat *rellevants*. Aquesta idea de considerar només la informació rellevant va ser lleugerament perfilada a [BBJ98]. Addicionalment, a [Tom97] trobem el llenguatge *SQL/TP*, un enfocament basat en punts de temps per a extensions temporals d'*SQL*. De tota manera, *SQL/TP* també fixa problemes presents en la semàntica dels llenguatges temporals basats en intervals de temps. [Tom97] defineix l'operació *normalization* sobre un interval de temps que transforma un conjunt arbitrari de consultes en un conjunt equivalent de consultes, en els quals, si la porció de dades d'un tuple es refereix a un interval $Q_i(D)$ i a un altre interval $Q_j(D)$, aleshores sempre es donarà el cas que aquests intervals coincideixin o siguin disjunts. *SQL/TP* garanteix només un cost polinòmic (lineal) creixent en mida per a una subclasse de consultes formulades amb lògica de primer ordre. Aquest fet és una de les raons que ens han portat a basar la nostra operació de *tall* en l'operació de *normalization*.

7.2. La base de dades completada

Abans de definir formalment una base de dades completada, definirem què entenem per *termes independents* i *dependents temporalment*.

Informalment, diem que un terme és *independent temporalment* si té un valor específic per un interval de temps i aquest valor és el mateix per a tots els subinterval·ls d'aquest interval de temps. Conceptualment, un terme independent temporalment rep la notació: *IND*. Un terme és *dependent temporalment* si té un valor concret per un interval de temps i aquest valor pot ser diferent en els subinterval·ls d'aquest interval. En aquest cas, aquest nou valor s'ha de poder derivar de la informació existent a la base de dades. Conceptualment, un terme dependent temporalment rep la notació: *DEP*.

Per il·lustrar aquests dos tipus de termes hem escollit un predicat *empleats* amb els termes: *Nom*, *Sou* i *Departament* i els temps vàlid d'inici V_s i de finalització V_e per codificar l'interval I . *Nom* és un terme que possiblement és el mateix per a tota la vida d'un empleat; per tant, és independent del temps. En canvi, *Sou* és un terme dependent temporalment perquè quan demanem informació per un subinterval de l'interval que tenim, el valor que esperem que ens doni és diferent del que tenim emmagatzemat. *Departament* pot canviar diverses vegades per a un empleat determinat, però això no impedeix que sigui un terme independent temporalment, ja que per a un interval determinat corresponent a l'assignació d'un empleat a un departament, tots els seus subinterval·ls tindran el mateix valor. La dependència temporal es pot veure com una característica complementària de la distinció *tèlic/atèlic* de [TS04]. Per exemple, *Nom* és *atèlic* i independent temporalment, mentre que *Sou* és també *atèlic* però dependent temporalment. No podem dir el mateix respecte al treball de [BJW00] perquè tots els termes que hem trobat eren independents temporalment.

Donat un predicat $p(K, A, I)$ amb els arguments següents: K és el conjunt de termes K_1, \dots, K_m que formen la clau, A és el conjunt de termes no temporals A_1, \dots, A_n i I és el terme de l'interval de temps. Donat un fet $p(k, x, t)$ on els valors $k \in K$, els valors $x \in A$ (el valor $x_i \in A_i$) i el valor $t \in I$.

Definició 1. Un argument A_i d'un predicat p és *independent temporalment* si per a tots els fets del predicat p , tots els subintervalls $t' \subset t$ satisfan que $p(k, \dots, x_i, \dots, t) \Rightarrow p(k, \dots, x_i, \dots, t')$.

Definició 2. Un argument A_i d'un predicat p és *dependent temporalment* si existeix un fet del predicat p , on un subinterval $t' \subset t$ satisfà que $p(k, \dots, x_i, \dots, t)$ es compleix & $p(k, \dots, x_i, \dots, t')$ no.

K sempre serà un terme o un conjunt de termes independents temporalment perquè, com explicarem després, K ens determinarà el conjunt de fets que haurem de considerar en el mateix grup a l'hora de cobrir una base de dades.

Definició 3. La base de dades completada, anomenada F^{comp} , es defineix com a:

Per a tots els fets de tots els predicats $p(K, A_1, \dots, A_n, I)$ de la base de dades, utilitzem $compl[f, I]p$ per trobar els valors per a tots els subintervalls $I' \subset I$.

En termes d'una *interpretació d'Herbrand* \mathcal{I} , definim formalment $compl[f, [V_s, V_e]]p$ per a cada terme A_i com a:

$\mathcal{I}(compl[f, [V_s, V_e]]p(K, \dots, f(A_i, [V_s, V_e], [VT_s, VT_e]), \dots, [VT_s, VT_e]), \theta)$ es compleix si $[VT_s, VT_e] \subset [V_s, V_e]$ & $is_dep(p, i) \Rightarrow \mathcal{I}(p(K, \dots, A_i, \dots, [V_s, V_e]), \theta)$ es compleix

$\mathcal{I}(compl[f, [V_s, V_e]]p(K, \dots, A_i, \dots, [VT_s, VT_e]), \theta)$ es compleix si $[VT_s, VT_e] \subset [V_s, V_e]$ & $no\ is_dep(p, i) \Rightarrow \mathcal{I}(p(K, \dots, A_i, \dots, [V_s, V_e]), \theta)$ es compleix

El predicat $is_dep(Pred, ArgNr)$ es compleix si l'argument A_i , on $i = ArgNr$, d'un predicat $Pred$ és dependent temporalment, i fals en qualsevol altre cas. $compl[f, [V_s, V_e]]p$ genera tots els subintervalls dels fets d'un predicat i , aplicant una funció f , obté els valors per als termes dependents temporalment de tots els subintervalls de $[V_s, V_e]$. Si $[V_s, V_e]$ no s'especifica, aleshores la mateixa funció f s'aplica a tots els fets amb la mateixa clau. Per als termes independents temporalment, $compl[f, [V_s, V_e]]p$ també genera tots els subintervalls, però, independentment de la funció f especificada, sempre tenen el mateix valor. Aquesta és la raó per la qual per als termes independents temporalment la funció f és completament opcional.

Encara que podríem completar una base de dades no coberta, normalment necessitarem completar bases de dades cobertes. Considerem un exemple d'aquest darrer cas on el predicat *empleats* és el que hem introduït abans amb els dos fets següents:

empleats ('jordi', 72000, 1, 1, 3)
empleats ('jordi', 20000, 2, 7, 9)

Si formulem la consulta següent:

← *empleats* ('jordi', *Sou*, *Dep*, 2, 9)?

Entre 3 i 7 tenim un forat d'informació. Després de cobrir aquest forat d'informació, generar els valors dels subintervals per a la base de dades coberta i obtenir la base de dades completada, podrem donar resposta a la consulta anterior.

Per distingir la base de dades original de la base de dades obtinguda després de fer el cobriment dels forats d'informació, haurem de fer servir la definició següent:

Definició 4. La base de dades coberta, anomenada F^{cov} , es defineix com a $F \cup F^*$, on F és el conjunt de tots els fets de la base de dades.

Definició 5. F^* es defineix com a:

$$\begin{aligned}
 F^* &= F_{p_1}^* \cup \dots \cup F_{p_i}^* \cup \dots \cup F_{p_{nb}}^* \\
 F_{p_i}^* &= F_{k_1}^* \cup \dots \cup F_{k_j}^* \cup \dots \cup F_{k_{nv}}^* \\
 F_{k_j}^* &= \{primerF_{k_{j1}}^* / cobrirF_{k_{j1}}^*\} \cup cobrirF_{k_{j2}}^* \cup \dots \cup cobrirF_{k_{jl}}^* \cup \dots \cup \\
 &\quad cobrirF_{k_{jh-1}}^* \cup \{darrerF_{k_{jh}}^* / cobrirF_{k_{jh}}^*\} \\
 &\quad \{primerF_{k_{j1}}^* / cobrirF_{k_{j1}}^* / darrerF_{k_{j1}}^*\} = \Gamma \{primer / cobrir / darrer\} [f_r] F_{k_{jr}}^*
 \end{aligned}$$

$F_{k_j}^*$ és el conjunt de tots els h forats d'informació entre els fets amb el mateix valor k_j a la clau del predicat p_i , nv és el nombre de valors k diferents de p_i , nb és el nombre de predicats de F i n és el nombre d'arguments de p_i . Γ indica que l'operació *primer/cobrir/darrer* s'aplicaran a $F_{k_{jr}}^*$ amb la corresponent funció f_r per a tots els arguments.

Més endavant explicarem amb detall les operacions que apareixen a la definició 5.

Com hem comentat en el capítol 4, el fet de tenir dimensions temporals fa que hàgim de tenir presents dependències funcionals temporals. Suposem que K és la clau i K^T és la clau temporal de $p(K, A_1, \dots, A_n, I)$. K^T serveix per identificar de forma única tots els fets del predicat p . Però, en el nostre cas, la clau K és especialment important per agrupar els fets a l'hora de cobrir una base de dades. La idea és força semblant a la proposada per [SS87] de cara a generar les seqüències de temps.

Exemple. Suposem que tenim els fets següents, on l'atribut *Nom* és clau:

empleats ('jordi', 72000, 1, 1, 3)
empleats ('jordi', 20000, 2, 7, 9)
empleats ('maria', 100000, 1, 1, 5)
empleats ('maria', 40000, 1, 7, 9)

Per cobrir els forats d'informació, agruparem separatament el fets segons la seva clau, en aquest cas, l'atribut *Nom*. Per tant, d'una banda tindrem els de l'empleat *jordi*, i de l'altra els de l'empleada *maria*, com es pot veure a la figura 7.4. Per tant, si $F_{empleats}$ és el conjunt de tots els fets del predicat *empleats*, F_{jordi} és el conjunt de tots els fets amb el mateix valor *jordi* a la seva clau i F_{maria} és el conjunt de tots els fets amb el mateix valor *maria* a la seva clau, aleshores considerem que, en aquest cas:

$$F_{empleats} = F_{jordi} \cup F_{maria}$$

$$F_{empleats}^* = F_{jordi}^* \cup F_{maria}^*$$

Els rectangles ratllats de la figura 7.4 indiquen el conjunt de forats F_{jordi}^* i F_{maria}^* .

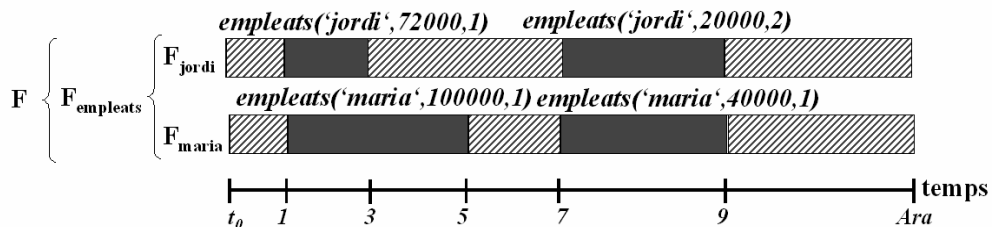


Figura 7.4. Agrupació dels fets del predicat *empleats* per ser coberts

A més, el conjunt de tots els fets F de la base de dades per a tots els predicats p_i , on $1 \leq i \leq nb$, i on nb és el nombre total de predicats ($nb = 1$ en el nostre exemple), és:

$$\begin{aligned} F &= F_{empleats} \\ F^* &= F_{empleats}^* \\ F^{cov} &= F \cup F^* \end{aligned}$$

Com hem vist amb l'exemple anterior, el cobriment del conjunt de forats $F_{empleats}^*$ consisteix a agrupar els fets amb la mateixa clau i aleshores aplicar l'operació de $cobrir[f]p$. La figura 7.5 mostra com es poden trobar els forats d'informació per ser coberts.

$cobrir[f]p$

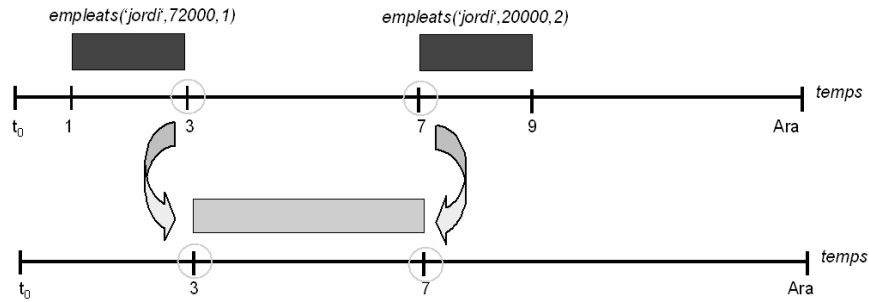


Figura 7.5. $cobrir[f]p$

Definició 6. En termes d'una interpretació d'Herbrand \mathcal{I} , definim formalment $cobrir[f]p$ com a:

$\mathcal{I}(cobrir[f]p(K, \dots, f(p, i, Y_i, [T1, VT_s], Z_i, [VT_e, T2], [VT_s, VT_e]), \dots, [VT_s, VT_e]), \theta)$ es compleix si

- $\mathcal{I}(p(K, \dots, Y_i, \dots, [T1, VT_s]), \theta)$ es compleix &
- $\mathcal{I}(p(K, \dots, Z_i, \dots, [VT_e, T2]), \theta)$ es compleix & $VT_s < VT_e$ &
- per a tot V_s & V_e $\{ \mathcal{I}(p(K, \dots, W_i, \dots, [V_s, V_e]), \theta)$ es compleix &
- $intersect([VT_s, VT_e], [V_s, V_e]) = \emptyset \}$

Parametritzem l'operació de *cobrir* per permetre que la nova informació es pugui obtenir amb una funció que es pugui utilitzar segons com es vulgui: la informació de la dreta, la informació de l'esquerra, totes dues (per exemple la mitjana), i fins i tot, que es pugui afegir informació mitjançant una constant. També es podrien fer servir funcions d'interpolació, com per exemple algunes de les que es proposen a [SS97], [JS96] i [BJW00]. La condició $VT_s < VT_e$ ens assegura que *cobrir* detecta els forats d'informació saltant tots els fets encavalcats o adjacents que no hagin estat fusionats.

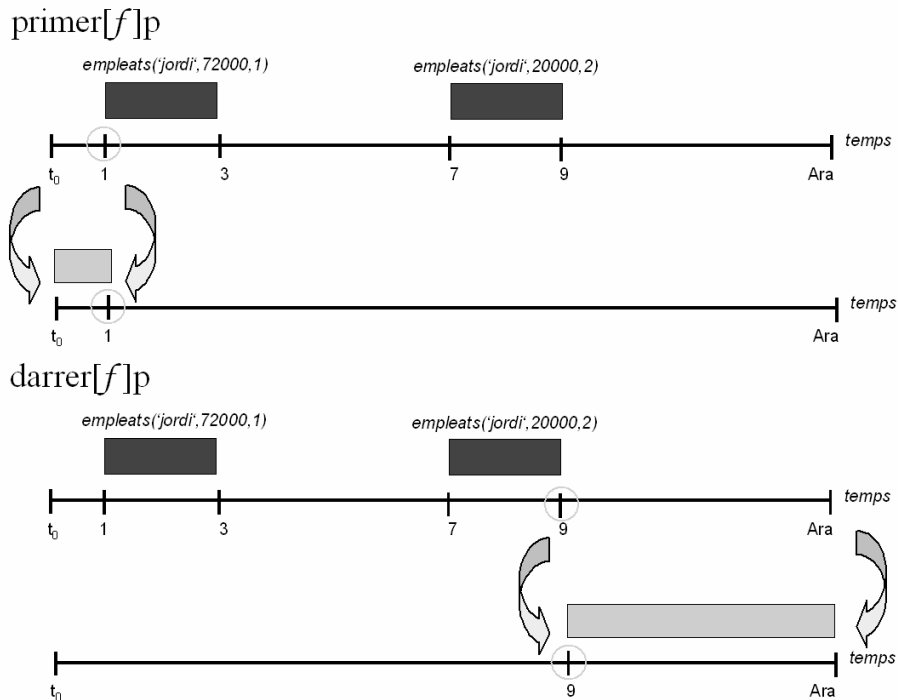


Figura 7.6. *primer[f]p* i *darrer[f]p*

La figura 7.6 ens mostra que *cobrir[f]p* no pot detectar el primer forat d'informació, ja que no tenim informació a l'esquerra del forat, i el mateix passa amb el darrer forat respecte a la informació de la dreta. Per resoldre aquest problema, calcularem els valors del primer i el darrer forat amb les operacions següents:

Definició 7. En termes d'una interpretació d'Herbrand \mathcal{I} , definim formalment *primer*[f]p com a:

$\mathcal{I}(\text{primer}[f]p(K, \dots, f(p, i, -, -, Z_i, [VT_e, T2), [t_0, VT_e]), \dots, [t_0, VT_e]), \theta)$ es compleix si

$\mathcal{I}(p(K, \dots, Z_i, \dots, [VT_e, T2]), \theta)$ es compleix & $t_0 < VT_e$ &
per a tot V_s & V_e $\{\mathcal{I}(p(K, \dots, W_i, \dots, [V_s, V_e]), \theta)$ es compleix &
 $\text{intersect}([t_0, VT_e], [V_s, V_e]) = \emptyset\}$

Definició 8. En termes d'una interpretació d'Herbrand \mathcal{I} , definim formalment *darrer*[f]p com a:

$\mathcal{I}(\text{darrer}[f]p(K, \dots, f(p, i, Y_i, [T1, VT_s], -, -, [VT_s, Ara]), \dots, [VT_s, Ara]), \theta)$ es compleix si

$\mathcal{I}(p(K, \dots, Y_i, \dots, [T1, VT_s]), \theta)$ es compleix & $VT_s < Ara$ &
per a tot V_s & V_e $\{\mathcal{I}(p(K, \dots, W_i, \dots, [V_s, V_e]), \theta)$ es compleix &
 $\text{intersect}([VT_s, Ara], [V_s, V_e]) = \emptyset\}$

Fem notar que en els casos en què *cobrir*[f]p detecti el primer o el darrer forat, aleshores les condicions $t_0 < VT_e$ per a *primer*[f]p i $VT_s < Ara$ per a *darrer*[f]p, respectivament, no retornaran forats.

Exemple. A les figures 7.5 i 7.6 hem mostrat com es poden cobrir els forats d'informació per a *jordi*. Considerem que decidim aplicar una funció *distanciam* que deriva els nous valors del *Sou* utilitzant la mitjana dels valors que tenim a la dreta i a l'esquerra, sempre tenint en compte la distància. Per al terme *Departament*, usarem una funció *esq* que utilitzarà només la informació de l'esquerra, excepte per al primer forat, en què s'utilitzarà la informació de la dreta (funció *dret*).

$$\text{distanciam}(\text{Pred}, \text{ArgNr}, \text{OV1}, \text{I1}, \text{OV2}, \text{I2}, \text{I}') = (\text{OV1} / \text{distancia}(\text{I1}) + \text{OV2} / \text{distancia}(\text{I2})) / 2 * \text{distancia}(\text{I}')$$

$$\text{distanciam}(\text{Pred}, \text{ArgNr}, \text{OV1}, \text{I1}, -, -, \text{I}') = \text{OV1} / \text{distancia}(\text{I1}) * \text{distancia}(\text{I}')$$

$$\text{distanciam}(\text{Pred}, \text{ArgNr}, -, -, \text{OV2}, \text{I2}, \text{I}') = \text{OV2} / \text{distancia}(\text{I2}) * \text{distancia}(\text{I}')$$

$$\text{dret}(\text{Pred}, \text{ArgNr}, \text{OV1}, \text{I1}, \text{OV2}, \text{I2}, \text{I}') = \text{OV2}$$

$$esq(Pred, ArgNr, OV1, I1, OV2, I2, I') = OV1$$

$$F_{jordi}^* = primerF_{jordi}^* \cup cobrirF_{jordi}^* \cup darrerF_{jordi}^*$$

$$primerF_{jordi}^* = primer[distanciam]empleats('jordi', Sou, -, [V_s, V_e])^* \wedge \\ primer[dret]empleats('jordi', -, Dep, [V_s, V_e])^*$$

$$cobrirF_{jordi}^* = cobrir[distanciam]empleats('jordi', Sou, -, [V_s, V_e])^* \wedge \\ cobrir[esq]empleats('jordi', -, Dep, [V_s, V_e])^*$$

$$darrerF_{jordi}^* = darrer[distanciam]empleats('jordi', Sou, -, [V_s, V_e])^* \wedge \\ darrer[esq]empleats('jordi', -, Dep, [V_s, V_e])^*$$

Assumim que $t_0 = 0$ i Ara té el valor 14. La funció f pot ser tan complexa o simple com calgui. En el nostre exemple, utilitzant *distanciam*, *dret* i *esq* obtenim els fets següents per a F_{jordi}^* :

empleats ('jordi', 36000, 1, 0, 1)
empleats ('jordi', 92000, 1, 3, 7)
empleats ('jordi', 50000, 2, 9, 14)

Fixem-nos que només tenim en compte la informació al costat del forat. Es podrien considerar altres possibilitats estenent la definició de *cobrir[f]p*.

cobrir[f]p

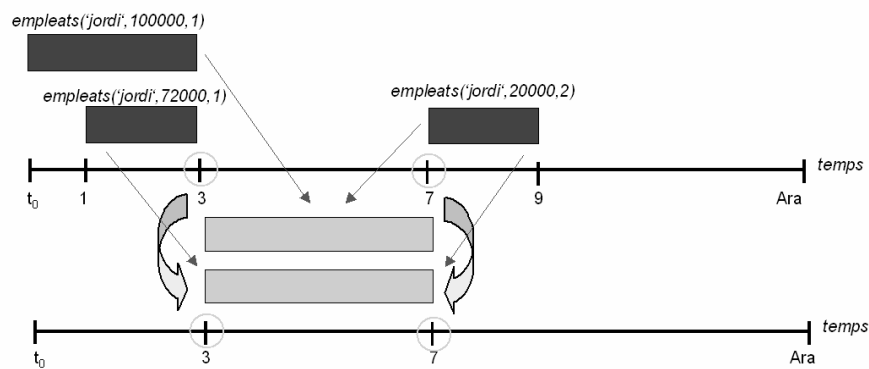


Figura 7.7. Un cas especial de *cobrir[f]p*

Per a intervals de temps adjacents o encavalcats, només es té en compte l'interval de temps més proper al forat que s'ha de considerar, excepte en el cas que es presenta a la figura 7.7, que produeix més d'un valor per a un sol forat.

La idea és aplicar una funció f per derivar nous valors. Per a termes independents temporalment es pot utilitzar una simple funció constant per derivar sempre el mateix valor. Per a termes dependents temporalment, la funció pot ser tan complexa com ens calgui, per exemple tenint en compte, creixement o decreixement, valors que canvien de manera contínua en el temps.

Exemple. Considerem F_{jordi}^{cov} :

empleats ('jordi', 36000, 1, 0, 1)
empleats ('jordi', 72000, 1, 1, 3)
empleats ('jordi', 92000, 1, 3, 7)
empleats ('jordi', 20000, 2, 7, 9)
empleats ('jordi', 50000, 2, 9, 14)

Utilitzant *compl[ffp]* apliquem una simple funció basada en la distància per al primer argument, on $DerivVal = OrigVal * distancia(I') / distancia(I)$; I és l'interval de temps del valor original i I' és l'interval de temps del valor derivat. Per al segon argument aplicarem una funció constant:

$$constant(OV, I, I') = OV$$

$$distancia(OV, I, I') = OV * distancia(I') / distancia(I)$$

Després de fer *compl[ff]empleats*:

compl[distancia]empleats('jordi', Sou, -, [V_s , V_e]) \wedge
compl[constant]empleats('jordi', -, Dep, [V_s , V_e]) \wedge
compl[distancia]empleats('jordi', Sou, -, [V_s , V_e])^{*} \wedge
compl[constant]empleats('jordi', -, Dep, [V_s , V_e])^{*}

s'obtenen els nous valors derivats:

empleats ('jordi', 36000, 1, 1, 2)
empleats ('jordi', 36000, 1, 2, 3)

empleats ('jordi', 23000, 1, 3, 4)
empleats ('jordi', 23000, 1, 4, 5)
empleats ('jordi', 23000, 1, 5, 6)
empleats ('jordi', 23000, 1, 6, 7)
empleats ('jordi', 46000, 1, 3, 5)
empleats ('jordi', 46000, 1, 4, 6)
empleats ('jordi', 46000, 1, 5, 7)
empleats ('jordi', 69000, 1, 3, 6)
empleats ('jordi', 69000, 1, 4, 7)
 ...
empleats ('jordi', 40000, 2, 10, 14)

En aquesta secció hem presentat els passos teòrics per completar una base de dades, encara que pràcticament no arribarem mai a generar una base de dades completada. Per aquesta raó pensem que seria interessant oferir *compl* i *cobrir* com a operacions sobre la base de dades completada, de manera que es puguin usar quan realment calguin, per resoldre consultes dependents temporalment o comprovar restriccions d'integritat dependents temporalment.

7.3. Operacions sobre la base de dades completada

La base de dades completada, ateses les seves característiques particulars, necessita unes operacions especials que aprofitin tota la seva potència de manera eficient. La unió temporal, intersecció temporal, diferència temporal, producte cartesià temporal, combinació temporal, selecció temporal i projecció temporal són les mateixes que les que s'han definit per a altres bases de dades temporals (per exemple [Sno00]). Per tant, en aquesta secció només presentarem les noves operacions que siguin específiques per a la manipulació de la base de dades completada. Les operacions *compl* i *cobrir* també podran ser utilitzades com a operacions per manipular la base de dades completada.

- **Operació *tall***

Definim l'operació de *tall* d'un conjunt de fets de dos predicats amb esquemes compatibles p_1 i p_2 com a:

$$tall(p_1, p_2) = \{ (\pi_1 p'_1 \cap \pi_1 p'_2) \bowtie p \mid p \in F_1^{comp} \cup F_2^{comp} \wedge \pi_1 p'_1 \text{ overlaps } \pi_1 p'_2 \wedge p'_1 \in F_1 \wedge p'_2 \in F_2 \}$$

on $[V1_s, V1_e]$ overlaps $[V2_s, V2_e]$ és equivalent a

$$V1_s \leq V2_s \wedge V1_e \leq V2_e \wedge V2_s < V1_e;$$

$$i [V1_s, V1_e] \cap [V2_s, V2_e] = [\max(V1_s, V2_s), \min(V1_e, V2_e)]$$

La figura 7.8 mostra gràficament la idea de l'operació de *tall* sobre els predicats *p1* i *p2*. L'operació de *tall* ens dona la mateixa informació per a exactament els mateixos intervals de temps en els dos predicats on s'aplica. Aquesta és la informació que hem anomenat anteriorment *rellevant*. La idea és utilitzar les bases de dades cobertes de *p1* i *p2* per trobar els subintervals de temps rellevants. Un cop fet això, només ens queda obtenir la informació per a aquests subintervals utilitzant les bases de dades completades de *p1* i *p2*.

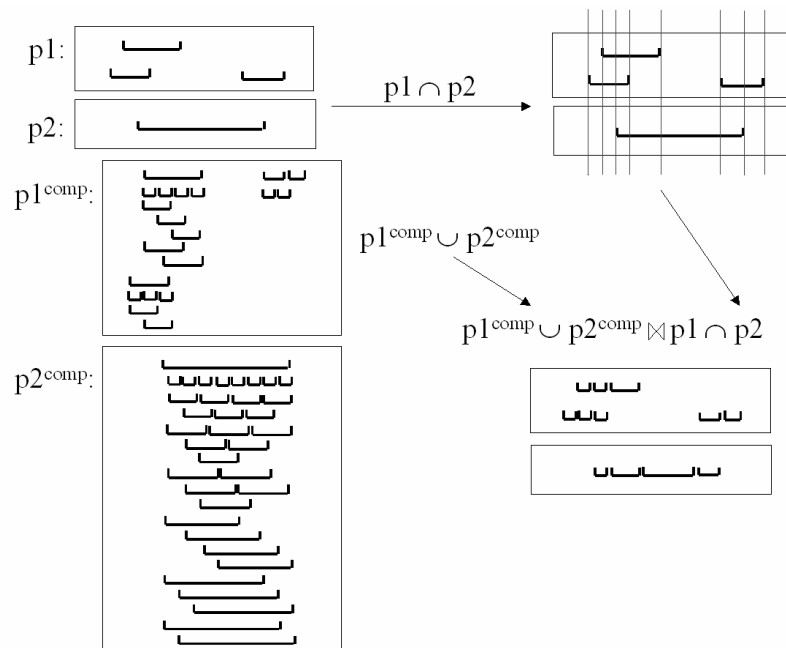


Figura 7.8. $tall(p_1, p_2)$

Exemple. Considerem els fets:

emp_1 ('jordi', 72000, 1, 1, 3)

emp_1 ('jordi', 50000, 1, 5, 7)

emp_2 ('jordi', 60000, 1, 1, 4)

Com hem comentat abans, normalment l'operació *compl* es sol fer sobre bases de dades cobertes. Considerem doncs, les bases de dades cobertes de emp_1 i emp_2 següents:

emp'_1 ('jordi', 72000, 1, 1, 3)

emp'_1 ('jordi', 72000, 1, 3, 5)

emp'_1 ('jordi', 50000, 1, 5, 7)

emp'_2 ('jordi', 60000, 1, 1, 4)

emp'_2 ('jordi', 60000, 1, 4, 7)

$$\pi_1 emp'_1 \cap \pi_1 emp'_2 = \{[1,3], [3,4], [4,5], [5,7]\}$$

Aleshores la informació *rellevant* es pot obtenir combinant el resultat de la intersecció anterior amb la informació obtinguda de la unió de les bases de dades completades d' emp_1 i emp_2 :

emp_1 ('jordi', 72000, 1, 1, 3)

emp_1 ('jordi', 36000, 1, 3, 4)

emp_1 ('jordi', 36000, 1, 4, 5)

emp_1 ('jordi', 50000, 1, 5, 7)

emp_2 ('jordi', 40000, 1, 1, 3)

emp_2 ('jordi', 20000, 1, 3, 4)

emp_2 ('jordi', 20000, 1, 4, 5)

emp_2 ('jordi', 40000, 1, 5, 7)

Fixem-nos que l'operació *tall* implícitament utilitza l'operació *compl* amb un interval de temps que ha de ser determinat per una consulta o restricció d'integritat dependent temporalment. En el nostre exemple, aquest interval és $[1, 7)$.

- **Operació fusió dependent**

L'operació de fusió es defineix a [BSS96] com una operació que no permet intervals de temps vàlid adjacents o encavalcats amb la mateixa informació no temporal. Per als termes independents temporalment amb intervals de temps vàlid adjacents o encavalcats es pot utilitzar l'operació de fusió de [BSS96], però per als intervals de temps vàlid adjacents o encavalcats de fets amb termes dependents temporalment necessitem definir una variant de l'operació de fusió:

$$\begin{aligned} dfusió(p) = \{ < \dots, f(A_i), \dots, p.[V_{is}, V_{je}] > \mid p_i \in F^{comp} \wedge p_j \in F^{comp} \\ & \wedge value_equivalent(\pi_{A_i, I} p_i, \pi_{A_j, I} p_j) \\ & \wedge \pi_{[V_{is}, V_{ie}]} p_i adjacent \pi_{[V_{js}, V_{je}]} p_j \} \end{aligned}$$

on:

$[V1_s, V1_e] adjacent [V2_s, V2_e]$ és equivalent a $V1_e = V2_s$;

$value_equivalent(\pi_{A_i, I} p_i, \pi_{A_j, I} p_j)$ és cert si A_i i A_j tenen la mateixa funció f de generació de valors per als subintervals.

Exemple. Considerem el fets:

$$\begin{aligned} emp_1 ('jordi', 60000, 1, 1, 3) \\ emp_1 ('jordi', 60000, 1, 3, 5) \end{aligned}$$

Després d'aplicar l'operació *dfusió* el nou valor es calcularia aplicant la funció definida per *Sou*, que haurà de ser la mateixa per a tots dos fets, i obtindriem:

$$emp_1 ('jordi', 120000, 1, 1, 5)$$

De tota manera, algunes vegades fins i tot necessitem fusionar intervals de temps vàlid adjacents amb diferent informació no temporal. En aquest cas, proposem la utilització de l'operació *suma*:

$$\begin{aligned} suma(p) = \{ < \dots, \pi_{A_i} p_i + \pi_{A_j} p_j, \dots, p.[V_{is}, V_{je}] > \mid p_i \in F^{comp} \wedge p_j \in F^{comp} \\ & \wedge \pi_{[V_{is}, V_{ie}]} p_i adjacent \pi_{[V_{js}, V_{je}]} p_j \} \end{aligned}$$

- **Operació selecció temporal completada**

La introducció de la idea de la base de dades completada obre la possibilitat d'expressar noves condicions a l'operació de selecció, on no siguin els intervals de temps de la base de dades original els que satisfacin les condicions, sinó implícitament tots els seus subinterval·ls.

Definirem la selecció temporal completada d'un conjunt de fets d'un predicat p com a:

$$\sigma^{comp}(condició)p(K,X,I) = \{p'(K,Y,I') \mid p' \in F^{comp} \wedge I' \subset I \wedge p' \text{ satisfà la condició}\}$$

Exemple. Considerem el fet:

$$emp_1('jordi', 60000, 1, 1, 3)$$

del qual derivem els subinterval·ls:

$$emp_1('jordi', 20000, 1, 1, 2)$$

$$emp_1('jordi', 40000, 1, 2, 3)$$

Si volem obtenir els subinterval·ls on el *Sou* sigui més gran o igual que 25000, hauríem de fer $\sigma^{comp}(Sou \geq 25000)empleats('jordi', Sou, 1, 1, 3)$, i obtindríem:

$$emp_1('jordi', 40000, 1, 2, 3)$$

- **Funcions d'agregació**

Les funcions d'agregació són essencials per resoldre la majoria dels problemes en què es necessiten les bases de dades completades. En el cas de les bases de dades completades, proposem ni més ni menys que utilitzar les funcions d'agregació habituals: *SUM*, *MAX*, *MIN*, *AVG* i *COUNT* sobre un conjunt de valors d'un terme d'un predicat $p \in F^{comp}$. La diferència és ara, doncs, que les funcions d'agregació s'apliquen tenint en compte els subinterval·ls. A la propera secció presentem un exemple de l'aplicació de les funcions d'agregació.

7.4. Consultes dependents temporalment

Les operacions que es defineixen a la secció 7.3 es poden utilitzar per formular consultes a la base de dades completada. Tenint termes dependents temporalment, la formulació d'una consulta $Q(BD)$ és especialment complexa o impossible de fer. Nosaltres proposem transformar la consulta a la base de dades original per una consulta a la base de dades completada $Q'(BD^{comp})$:

$$Q(BD) \rightsquigarrow Q'(BD^{comp})$$

Les consultes a la base de dades completada tenen la forma següent: $Q'[I_1, \dots, I_n]\phi$, on $[I_1, \dots, I_n]$ són el conjunt de subintervalls de temps de la consulta, i ϕ és la fórmula que usa les operacions que s'expliquen a la secció 7.3 per manipular la base de dades completada.

Com a exemple de solució de plantejament de consultes amb termes dependents temporalment, necessitem tornar a l'exemple de les estacions meteorològiques que hem presentat per introduir el problema. La consulta demanava la mesura de pluja entre les 9 i les 14 i entre les 15 i les 21 partint de la informació de tres estacions meteorològiques diferents ($M1$, $M2$ i $M3$).

Primer de tot, cobrim $M1$, $M2$ i $M3$ amb l'operació *cobrir*; aleshores, teòricament, generem la base de dades completada, i usant les operacions per manipular la base de dades completada, la consulta pot ser senzillament plantejada amb la consulta:

$$Q'[[9,15), [15,22)]AVG [\pi_{qts}suma(cobrir(M1) \cup cobrir(M2) \cup cobrir(M3))]$$

En aquest cas, la informació dels subintervalls *rellevants* es pot obtenir només tallant els intervals que apareixen a la consulta, $[9,15)$ i $[15,22)$. Aquesta és la raó per la qual realment no es genera tota la base de dades completada, sinó només la informació dels subintervalls *rellevants*. S'ha de tenir present que la forma d'una consulta dependent temporalment és: $Q'[I_1, \dots, I_n]\phi$, la qual cosa implica que ϕ s'aplica a I_1, \dots, I_n . Concretament, a la consulta anterior, l'operació *suma* i la funció *AVG* s'apliquen a $[9,15)$ i $[15,22)$, com es pot veure a la figura 7.9.

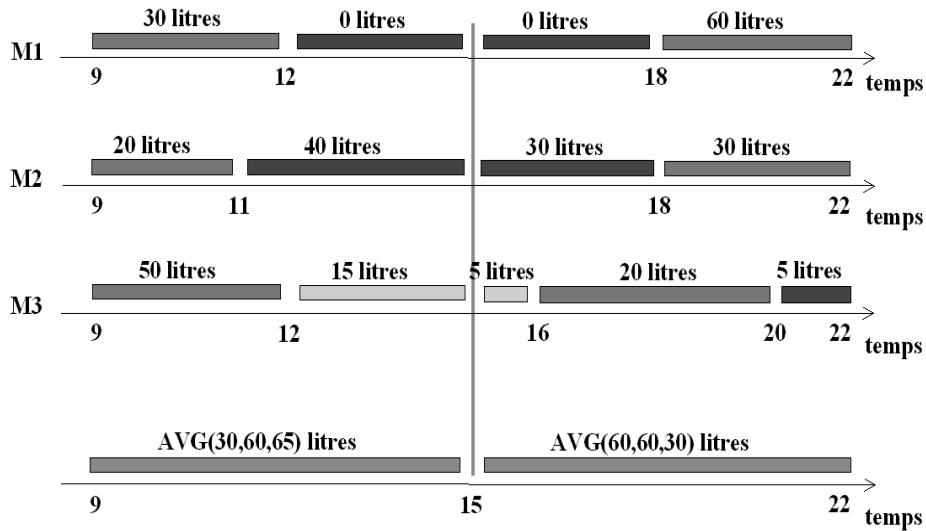


Figura 7.9. Una solució gràfica de la consulta de les estacions meteorològiques

7.5. Restriccions d'integritat dependents temporalment

Les restriccions d'integritat representen coneixement obtingut del món o el domini de la base de dades. Una restricció d'integritat no és més que una consulta tancada que ha de ser sempre certa després de qualsevol actualització de la base de dades. Per tant, per treballar amb restriccions d'integritat dependents temporalment, proposem l'ús de les mateixes operacions que hem presentat per resoldre consultes dependents temporalment. Les nostres fórmules són fórmules tancades (sentències) amb forma de denegació. Per motius d'uniformitat, tal com també hem fet al capítol 5, associarem a cada restricció d'integritat un predicat d'inconsistència *ic*.

Considerem la base de dades d'exemple amb el predicat *empleats*. Suposem que tenim el fet *empleats('jordi', 72000, 1, 1, 3)* del predicat *empleats* i definim la restricció d'integritat següent per impedir que un mateix empleat pugui tenir sous diferents a intervals encavalcats:

$$ic1 \leftarrow Q'[[1,3]]\sigma (Sou1 \neq Sou2 \wedge I_2 \text{ overlaps } I_1) \\ empleats(Nom, Sou1, Dpt, I_1) \wedge empleats(Nom, Sou2, Dpt, I_2)$$

Òbviament, la inserció d'*empleats('jordi', 40000, 1, 2, 3)* violaria *ic1*. A més, la inserció del fet *empleats('jordi', 36000, 1, 2, 3)* no hauria de violar *ic1*, però en canvi la viola.

El problema és que, si tenim termes dependents temporalment, necessitem usar la selecció temporal completada per expressar les condicions de les restriccions d'integritat, en lloc d'usar la selecció habitual:

$$ic2 \leftarrow Q'[[1,3]]\sigma^{comp} (Sou1 \neq Sou2 \wedge I_2 \text{ overlaps } I_1) \\ empleats(Nom, Sou1, Dpt, I_1) \wedge empleats(Nom, Sou2, Dpt, I_2)$$

Les restriccions d'integritat que es defineixen amb predicats que contenen termes dependents temporalment les anomenarem *restriccions d'integritat dependents temporalment*.

Adicionalment, tenir termes dependents temporalment ens ofereix la possibilitat de definir nous tipus de restriccions d'integritat. Per exemple, la restricció d'integritat *ic3* es defineix utilitzant la funció *distancia* per garantir que per a intervals de distancia 2 el sou sigui més gran o igual que 20000:

$$ic3 \leftarrow Q'[[1,3]]\sigma^{comp} (distancia(I)=2 \wedge Sou < 20000) \\ empleats(Nom, Sou, Dpt, I)$$

Conceptualment, les restriccions d'integritat dependents temporalment s'han de comprovar a la base de dades completada. Per fer més eficient la comprovació, la idea és fer la comprovació de les restriccions d'integritat dependents temporalment utilitzant la informació de la base de dades completada, però sense arribar a generar-la tota. L'operació *tall* serà el mecanisme clau que ens donarà la informació rellevant necessària per fer la comprovació, sense haver de fer la generació de la base de dades completada sencera.

A més, si per derivar els valors dels subinterval·ls apliquem una simple funció basada en la distància, ens adonem que tots els termes independents temporalment òbviament tenen sempre el mateix valor i tots els termes

dependents temporalment tenen el mateix valor per a tots els subinterval de la mateixa distància. D'aquesta manera, per als termes independents temporalment una funció constant és suficient, però per als termes dependents temporalment no cal gaire més que obtenir els $n - I$ valors diferents per a cada interval de distància n , utilitzant un nou predicat: $subin(Pred, ArgNr, I, D, V)$, on D és el valor des d' I fins a $distancia(I) - I$, i V és el valor per a tots els subinterval de la mateixa distància. Addicionalment, el predicat $subin$ es pot usar a la definició de les restriccions d'integritat dependents temporalment. $Ic4$ és un exemple d'una nova definició d' $ic3$ utilitzant el predicat $subin$:

$$ic4 \leftarrow Q'[1,3] \sigma^{comp}(subin(empleats, 2, I, 2, V) \wedge V < 20000) \\ empleats(Nom, Sou, Dpt, I)$$

8. Magatzems de dades bitemporals

L'estudi desenvolupat en el capítol 4 per a dues dimensions de temps ens va fer pensar que la potència temporal obtinguda per a les bases de dades bitemporals podria ser aplicada a altres àrees més joves. Concretament, l'àrea de magatzems de dades és una àrea que ha tingut un ràpid creixement en els darrers anys sense gaire formalització teòrica a causa de la imminent necessitat d'aplicació per part de la indústria. A [Vas00] podem trobar un estudi recent sobre recerca en magatzems de dades, així com a [JLV+00] es descriuen els conceptes bàsics d'aquesta àrea. Analitzant aquests treballs, veiem que en l'àrea de magatzems de dades s'han hagut de desenvolupar ràpidament mecanismes d'emmagatzemament i gestió de dades d'empresa per donar resposta als processos de presa de decisions, amb uns clars orígens d'aplicació pràctica per resoldre problemes empresarials.

L'àrea de bases de dades temporals és una àrea més antiga i estable comparada amb els magatzems de dades i sobretot és una àrea on la base teòrica està sòlidament establerta. En bases de dades temporals, la darrera bibliografia apareguda fins ara la podem trobar a [WJW98], on comprovem que els orígens i l'evolució d'aquesta altra àrea de recerca han estat sempre clarament acadèmics. La idea d'aquest capítol és explicar com es poden aplicar conceptes de bases de dades bitemporals per millorar el funcionament dels magatzems de dades. D'altra banda, en intentar tractar conjuntament dues àrees de recerca, en principi completament diferents, ens hem trobat que de l'àrea de magatzems de dades també es poden extreure temes interessants que encara no s'han treballat en

bases de dades temporals. També els diferents enfocaments d'aquestes dues àrees han estat, sens dubte, un incentiu més per intentar relacionar-les.

8.1. Bases de dades bitemporals i magatzems de dades

L'afinitat entre els conceptes de bases de dades bitemporals i magatzems de dades no és evident, però, en canvi, sí que és clara la necessitat de referenciar el temps en la gestió de les decisions empresarials. A [IIS98] es defineix un *magatzem de dades* com una estructura que gestiona dades “orientades al tema”, “integrades”, “històriques” i “no volàtils”. Concretament, els dos darrers conceptes són els que mostren l'esperit temporal dels magatzems de dades. A [SA86] una *base de dades bitemporal* es defineix com una base de dades que pot gestionar temps vàlid i temps de transacció. El concepte de *dades històriques* vol dir que cada registre del magatzem de dades és sempre relatiu a un moment de temps. El temps vàlid és el temps en què un fet és cert a la realitat modelada. Per tant, tots dos conceptes volen reflectir en quin moment les dades són correctes i es corresponen amb la realitat. El concepte de *dades no volàtils* es refereix al fet que els canvis en el magatzem de dades han de ser expressats amb un estampillatge de temps. Aquest segon concepte es pot identificar amb el de temps de transacció, el temps en què un fet és emmagatzemat a la base de dades.

Ateses les correspondències entre temps vàlid i historicitat i temps de transacció i no volatilitat, podem definir un *magatzem de dades* com una base de dades bitemporal que gestiona dades “orientades al tema” i “integrades”. A la figura 8.1 podem veure la representació gràfica d'aquesta idea. La primera implicació d'aquesta definició és que el temps de transacció serà completament mantingut pel sistema i cap usuari no el podrà canviar. A més, el sistema ens haurà de proveir dels mecanismes necessaris per gestionar el temps vàlid. La importància de la necessitat d'aquesta concepció bitemporal també la podem trobar a [PJ98], on es demana que els magatzems de dades gestionin conceptes temporals avançats. Un altre exemple d'aproximació entre aquestes dues àrees el podem trobar a [YW00], on s'explica com es poden definir vistes temporals sobre relacions no temporals.

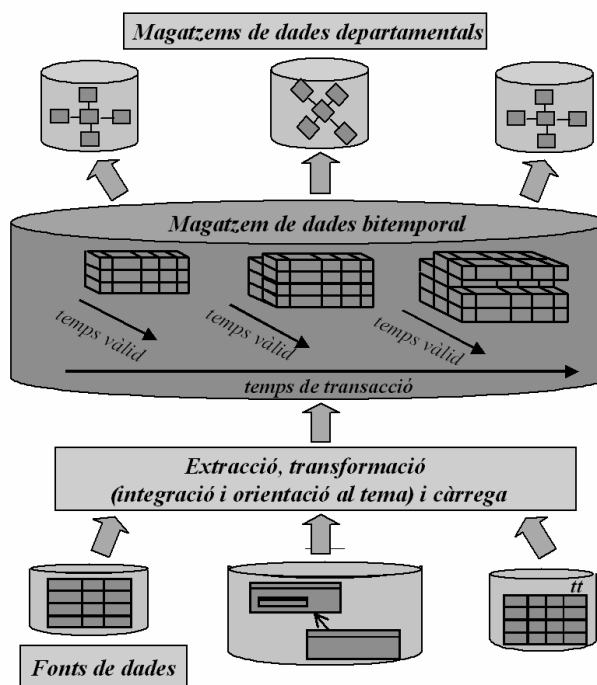


Figura 8.1. Una base de dades bitemporal com a magatzem de dades

A [AM03b], un resum de la versió estesa presentada a [MA02], s'expliquen, a més de com un magatzem de dades pot ser redefinit utilitzant una base de dades bitemporal, els principals punts en comú que s'han trobat entre aquestes dues àrees, agrupats de la manera següent:

- Conceptes temporals rellevants per als magatzems de dades

Tal com hem comentat abans, la naturalesa bitemporal d'un magatzem de dades comporta que aquest pugui gestionar les dimensions de temps vàlid i temps de transacció, àmpliament estudiades en bases de dades temporals. Respecte als magatzems de dades departamentals, com que només contindran una història parcial de les dades, caldrà convertir les dades bitemporals del magatzem de dades en dades en una sola dimensió de temps per als magatzems de dades departamentals. Per resoldre aquest problema es podrien aplicar els operadors fotogràfics presentats a [JSS92a]. Pel que fa a les fonts de dades, és important també remarcar que no totes les dades provinents de

les fonts ens arribaran amb un component bitemporal. Gestionar diferents objectes i atributs amb granularitats diferents es pot fer utilitzant treballs realitzats en aquesta àrea, com [BJW00]. A més, tots els conceptes utilitzats en bases de dades temporals ([JDB+98]): cronons, instants, intervals, conjunts d'instantos i conjunts d'intervals, es poden aplicar a magatzems de dades.

- Manipulació de dades temporals

La manipulació de dades és un problema important en magatzems de dades atesa la gran quantitat de dades que cal gestionar. Per resoldre aquest aspecte es pot considerar la utilització de tècniques de modificació temporals [JSS94], així com llenguatges temporals [Cho94], [Sno95], [BJS00], dependències funcionals temporals [Wij99], valors desconeguts per als temps de finalització [CDI+97], fusió [BSS96], etc.

- Gestió d'emmagatzemament temporal

Com que la nostra proposta és utilitzar una base de dades bitemporal com a magatzem de dades, arribats a aquest punt hauríem d'endinsar-nos en les opcions que ens ofereixen les bases de dades temporals d'estructures de representació d'emmagatzemament. A [JSS94] podem trobar les estructures més utilitzades en bases de dades temporals. A més, a la propera secció d'aquest capítol presentem una nova proposta que aprofita les particularitats dels magatzems de dades. També és important destacar respecte a aquest punt que en bases de dades temporals existeixen dos enfocaments [JS99]: *estratificat* i *integrat*, així com un ampli ventall de tècniques d'indexació temporal [ST99]. L'enfocament estratificat consisteix a desenvolupar una capa temporal sobre un altre SGBD no temporal. L'enfocament integrat proposa modificar o estendre els mòduls interns d'un SGBD de cara a poder gestionar dades temporals.

- Models de dades temporals orientats a objectes

Els models orientats a objectes ens proveeixen d'una riquesa semàntica que no només facilita la integració, sinó que, a més, ajuda l'analista a conèixer millor el significat de les dades. A més, fan més senzill l'assoliment de l'orientació al tema, si definim cada tema com un objecte diferent. Novament,

en aquest cas, la nostra proposta és utilitzar models de dades temporals orientats a objectes com [BFG97] i [SN97]. Una altra característica important dels models orientats a objectes és la definició de mètodes per objectes i classes. Un exemple del que serien mètodes temporals el podem trobar a [EK01].

8.2. Una estructura de dades bitemporal com a magatzem de dades

A [Dev97] s'explica la naturalesa històrica d'un magatzem de dades, distingint entre dos tipus de dades temporals. Les *dades transitòries* es poden representar amb una relació en temps vàlid, mentre que les *dades periòdiques* amb una relació bitemporal, que té com a estructura de representació *l'estructura basada en backlog* que es presenta a [JSS94]. Donat un esquema d'una relació bitemporal R que té els atributs A_1, \dots, A_n , la seva representació seria la següent:

$$R = (A_1, \dots, A_n, V_s, V_e, T, Op)$$

On V_s és el temps vàlid d'inici, V_e és el temps vàlid de finalització, T és el temps de transacció quan el tuple s'emmagatzema i Op és l'operació (inserció, esborrat o modificació) que es realitza.

A [KRR+98] es descriuen tres estructures diferents per a magatzems de dades en un entorn de control d'inventari d'una granja. El primer model és el *model de transacció*, que novament es pot representar amb l'estructura basada en *backlogs*. En el segon *model d'estat de distribució* es construeix un registre per a cada distribució de producte de la granja. El tercer model és el *model d'inventari*, que es pot representar amb *l'estructura d'estampillatge de tuples* explicada a [JSS94].

$$R = (A_1, \dots, A_n, V_s, V_e, T_s, T_e)$$

On V_s és el temps vàlid d'inici, V_e és el temps vàlid de finalització, T_s és el temps de transacció d'inici i T_e és el temps de transacció de finalització.

El problema de l'estructura d'estampillatge de tuples és que, encara que els canvis afectin només un atribut del tuple, tot el tuple en queda afectat. Per això a

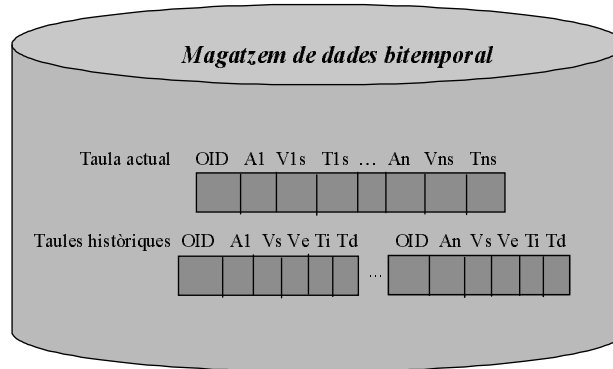
[JSS94] es presenta també una estructura que no està en primera forma normal, l'estructura d'estampillatge d'atributs.

$$R = (\{[A_1, [T_s, T_e], [V_s, V_e]], \dots\}, \dots, \{[A_n, [T_s, T_e], [V_s, V_e]], \dots\})$$

Amb aquesta representació un tuple es compon de n conjunts, on cada element d'un conjunt és un triplet compost per un valor d'un atribut, un interval de temps vàlid $[V_s, V_e]$ i un interval de temps de transacció $[T_s, T_e]$.

A [AM03a] es proposa una nova estructura d'emmagatzemament per a magatzems de dades, anomenada *actual/històrica*. Tal com es mostra a la figura 8.2, només hi haurà dos tipus de taules. La *taula actual* contindrà les dades actuals, on V_s serà el temps vàlid d'inici i T_s serà el temps de transacció d'inici. No caldrà emmagatzemar V_e , ja que sempre serà *Ara*, ni tampoc T_e , que sempre serà *FNC*. Les taules històriques capturen la història completa per a un atribut amb els atributs següents: V_s serà el temps vàlid d'inici, V_e serà el temps vàlid de finalització que ara serà conegut, T_i serà el temps de transacció en què la inserció és processada i T_d és el temps de transacció en què es processa l'esborrat.

Figura 8.2. Una taula per a les dades actuals i n taules per a les dades històriques



A la figura 8.3 podem veure un exemple del funcionament de l'estructura actual/històrica en el cas que ens arribi una operació d'inserció per a una relació que emmagatzema *assignatures* amb els atributs: *OID*, *Nom* i *Preu*. L'exemple es presenta per al cas de fonts de dades que proveeixen només temps de transacció. A [MA03] es pot veure el funcionament de l'estructura actual/històrica per a la resta de fonts de dades.

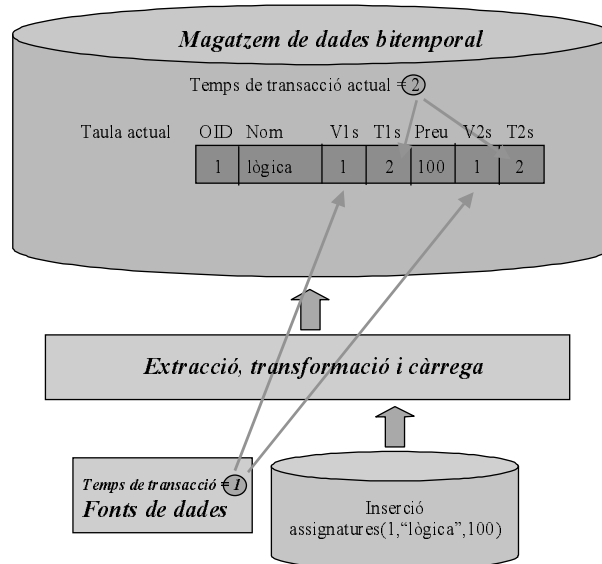


Figura 8.3. Inserció del curs de lògica a la taula actual

A la figura 8.3 veiem la inserció de l'assignatura de *lògica* amb un preu de 100 euros i un temps de transacció d'1. Per processar una operació d'inserció l'únic que hem de fer és afegir un tuple a la taula actual. El temps de transacció de la font es converteix en el temps vàlid d'inici del magatzem de dades i el temps de transacció d'inici del magatzem de dades l'obtenim directament del magatzem de dades.

A la figura 8.4 podem veure l'esborrat de l'assignatura de *lògica* amb un preu de 100 euros i un temps de transacció de 3. Per processar una operació d'esborrat, el que hem de fer és treure la informació de la taula actual i afegir-la a les taules històriques. Els temps vàlid i de transacció d'inici de les taules històriques són els que teniem a la taula actual. El temps de transacció de finalització de les taules històriques s'obté del temps de transacció del magatzem de dades i el temps vàlid de finalització de les taules històriques s'obté del temps de transacció de les fonts.

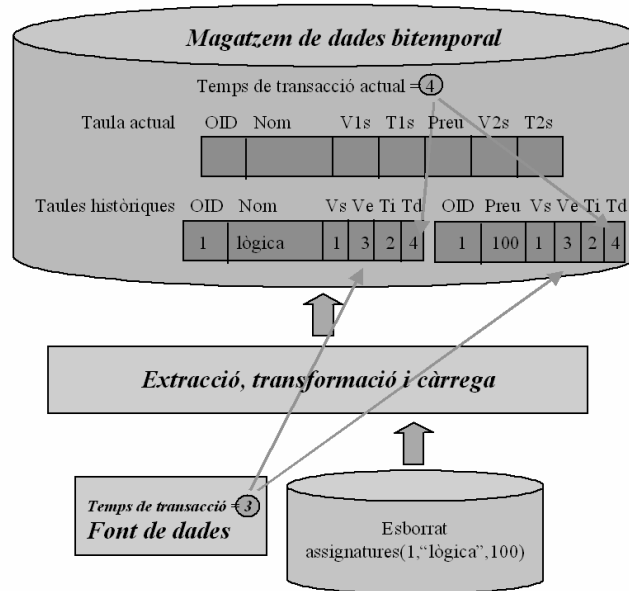


Figura 8.4. Inserció del curs de lògica a les taules històriques

9. Conclusions i recerca futura

En aquest capítol presentem les aportacions principals d'aquesta tesi, així com l'estat actual de la nostra recerca i de les seves futures directrius.

9.1. Conclusions

En aquesta tesi hem presentat un nou mètode per a la comprovació de les restriccions d'integritat en bd-dbts.

La primera idea en què es va basar el mètode proposat per aquesta tesi va ser introduïda a [MS94c], després d'haver fet un acurat estudi dels últims treballs en l'àrea de bases de dades temporals a [MS94b] i dels mètodes existents per a la comprovació de restriccions d'integritat en bd-dbts a [MS94a].

Els coneixements necessaris de bases de dades temporals els vam enriquir assistint a la primera reunió de treball internacional de bases de dades temporals [SJS95], on es van presentar els prototipus existents de bases de dades temporals [Böh95] i es van marcar les pautes de les necessitats de recerca en aquesta àrea [CJS+95], entre les quals es trobava precisament el tema de la nostra tesi. En l'àmbit de les bases de dades deductives, vam participar en els projectes ODISSEA [MSS92] i FOLRE [UTP+94], atès que per al nostre mètode estàvem a cavall de dues àrees de recerca, que havíem de conèixer en profunditat.

Sempre treballant en bases de dades deductives, el primer pas va ser trobar un mètode per a la comprovació de restriccions d'integritat incorporant només una dimensió temporal a [MS96a]. Un cop fet això, vam començar a incorporar, per primer cop, dues dimensions de temps a [MS96b]. L'aplicació de regles de transició amb resolució SLDNF per comprovar restriccions d'integritat en bd-dbts va resultar molt prometedora, però amb un nombre molt elevat de regles de transició, malgrat que ja havíem trobat algunes primeres simplificacions. A partir d'aquí, vam centrar la nostra feina a simplificar al màxim les regles de transició que teníem. El 1998 vam assistir a un tutorial sobre bases de dades temporals [CT98a] per assentar coneixements. I els primers resultats interessants van ser presentats a [Mar98]. Per arribar a obtenir encara més simplificacions de les regles de transició, vam estudiar la semàntica de les restriccions d'integritat temporals a [Mar00], que va donar lloc a les simplificacions introduïdes a [Mar01] i ja presentades en aquesta tesi.

Un cop repassat el procés de desenvolupament d'aquesta tesi, recordarem les principals contribucions:

- La utilització del llenguatge de lògica de predicats de primer ordre amb tractament explícit de punts de temps per a la manipulació de transaccions i la definició de restriccions d'integritat en bd-dbts.
- La definició del concepte de transacció coherent, que simplifica molts passos de la comprovació de restriccions d'integritat en bd-dbts.
- La formalització de l'aplicació de la fusió de rectangles bitemporals en bd-dbts.
- La incorporació d'uns nous predicats ι^* i δ^* , que afegeixen el temps de transacció automàticament a les insercions i els esborrats de les transaccions, de manera que les fan transparents a l'usuari.

- Una nova classificació de restriccions d'integritat temporals per a bd-dbts estudiant el seu comportament.

Restriccions d'integritat de Michael Böhlen	estàtiques intraestat intraelement dinàmiques interestat interelement			
	Restriccions d'integritat	Primer grup actuals	Segon grup en t. de transacció	Tercer grup en temps vàlid
Restriccions d'integritat per condicions	√	√	√	√
Restriccions d'integritat síncrones en t. vàlid	} √	√		(√)
Restriccions d'integritat síncrones en t. transacció			√	(√)
Restriccions d'integritat periòdiques en t. vàlid			√	√
Restriccions d'integritat periòdiques en t. trans		√		√
Restriccions d'integritat d.d.l. en t. vàlid			√	√
Restriccions d'integritat d.d.l. en t. transacció		√		√

Figura 9.1. Quadre resum de la classificació de restriccions d'integritat

- Una reducció dràstica de les regles de transició del mètode de comprovació de restriccions d'integritat per a bd-dbts.

Restriccions d'Integritat	Primer grup actuals	Segon grup en temps de transacció	Tercer grup en temps vàlid	Quart grup bitemporals
Cas 1: Predicats positius	Darrera regla	Totes les regles menys la darrera	Darrera regla	Totes les regles menys la darrera
Cas 2: Predicats positius i negats	Darrera regla	Totes les regles menys la darrera	Darrera regla	Totes les regles menys la darrera
Cas 3: Predicats negats	Darrera regla	Totes les regles menys la darrera	Darrera regla	Totes les regles menys la darrera

Figura 9.2. Quadre resum general de les simplificacions obtingudes per regles d'integritat

- La utilització de la resolució SLDNF per a la comprovació de restriccions d'integritat en bd-dbts.

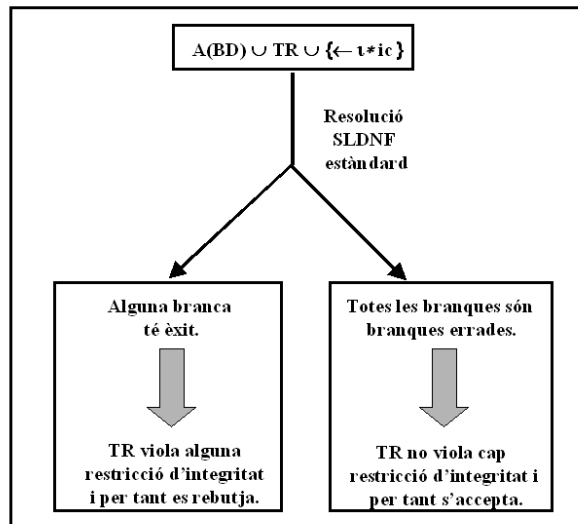


Figura 9.3. Resolució SLDNF per a la comprovació de restriccions d'integritat en bd-dbts

- L'estudi de la dimensió del temps de transacció ens ha donat la clau per reduir el nombre de regles de transició. De l'estudi desenvolupat en la dimensió de temps vàlid, cercant la possibilitat de poder definir restriccions d'integritat temporals més potents, hem explicat les característiques dels termes dels predicats dependents temporalment i hem introduït la idea teòrica de la base de dades completada, que proveeix informació per a tots els subintervalls de temps. A la base de dades completada, els forats d'informació es cobreixen i la informació per a tots els subintervalls d'un interval de temps determinat és derivada. Tenint termes dependents temporalment, hem motivat la necessitat de tenir unes operacions adequades per poder manipular-los. Aquestes operacions han estat definides per a termes independents i dependents temporalment. També hem analitzat les seves propietats i utilitat en la seva aplicació a consultes dependents temporalment i restriccions d'integritat dependents temporalment.

- En l'àrea de magatzems de dades, hem identificat les dues dimensions de temps: temps vàlid i temps de transacció i hem presentat la nostra estructura d'emmagatzemament bitemporal, anomenada actual/històrica.

9.2. Recerca futura

El mètode per a la comprovació de les restriccions d'integritat en bd-dbts que es presenta en aquesta tesi, pot ser encara ampliat, afegint-hi nous resultats de recerca. Per exemple, en el nostre mètode, encara que es planteja la resolució de restriccions d'integritat amb regles de deducció perquè es vegi clar el seu funcionament, no hem entrat a treballar a fons en aquest aspecte. S'haurien d'estudiar en profunditat les regles de deducció i, per tant, les regles d'esdeveniments, i veure si és possible obtenir simplificacions tan interessants com les obtingudes per a les regles de transició. També es podria mirar d'aplicar altres tècniques combinades amb la nostra, com per exemple la utilització de grafs proposada a [Ger96].

La versió del mètode presentada a [MS96b] va ser implementada directament en un sistema Prolog, i es va comprovar, entre altres coses, la necessitat d'obtenir més simplificacions de les regles de transició. La implementació consisteix en un prototipus d'aplicació del mètode que mostra el seu funcionament amb la utilització de transaccions introduïdes per l'usuari. Una possible feina de continuïtat, en l'aspecte de la implementació, seria fer l'entorn de l'usuari molt més amigable i incorporar les aportacions presentades en aquesta tesi.

Hi ha certs temes que s'han hagut de tocar mínimament per desenvolupar el mètode de comprovació i, per tant, els hem hagut de tractar, però sense aprofundir-hi gaire, ja que no eren l'objectiu principal d'aquesta tesi. És el cas, per exemple, de la manipulació de transaccions o la fusió de rectangles bitemporals, en els quals es podria treballar més a fons. Concretament, en el cas de la fusió es podria ampliar la feina desenvolupada per [BSS96] en bases de dades en temps vàlid.

Per desenvolupar aquest mètode de comprovació de restriccions d'integritat en bd-bts, hem hagut d'entrar en altres àrees de recerca pròximes, la qual cosa ens ha fet pensar que potser seria interessant estudiar amb més detall altres

àmbits, com ara el de les versions de bases de dades o el dels magatzems de dades (en el que ens hem començat a introduir en el tema 8 d'aquesta tesi) i veure com es poden aprofitar aquestes àrees de les aportacions d'aquesta tesi.

En els capítols 7 i 8 hem presentat uns primers resultats assentats de la nostra recerca actual, però aquests dos capítols formen clarament part de la nostra recerca futura. En aquests moments estem treballant en la implementació dels termes dependents i independents temporalment utilitzant el sistema temporal estratificat TIGER (<http://www.cs.aau.dk/~tigeradm/>). La idea és oferir els termes dependents i independents temporalment com una nova funcionalitat més del llenguatge ATSQL [BJS00]. Respecte als magatzems de dades bitemporals hem implementat l'estructura actual/històrica i les tres estructures més conegudes de representació de bases de dades bitemporals per poder obtenir una comparativa que demostrï que l'estructura actual/històrica és gairebé sempre millor que les altres i sens dubte, en el cas de magatzems de dades.

Finalment, res no ens agradaria més que haver treballat en un tema de recerca totalment aplicable a la pràctica. I és per això que pensem continuar al dia en aquesta àrea per poder participar, en el moment que això sigui possible, en el disseny d'un SGBD deductiu bitemporal integrat que incorpori el nostre mètode de comprovació de restriccions d'integritat.

Referències

- [All83] Allen, J.F. "Maintaining Knowledge about Temporal Intervals". *Communications of the ACM*. Vol. 26. Núm. 11. 1983, p. 832-843.
- [All84] Allen, J.F. "Towards a General Theory of Action and Time". *Artificial Intelligence*. Vol. 23. Núm. 2. 1984, p. 123-154.
- [ABW88] Apt, K.R.; Blair, H.A.; Walker, A. "Towards a Theory of Declarative Knowledge". A: [Min88], p. 89-148.
- [AM03a] Abelló, A.; Martín, C. "A Bitemporal Storage Structure for a Corporate Data Warehouse". *Proc. of the ICEIS'03*. Angers, 2003, p. 177-183.
- [AM03b] Abelló, A.; Martín, C. "The Data Warehouse: An Object-Oriented Temporal Database". *Proc. of JISBD'03*. Alacant, 2003, p. 675-68.
- [AS86] Ahn, I.; Snodgrass, R.T. "Performance Evaluation of a Temporal Database Management System". *Proc. of the SIGMOD'86*. Washington D.C., 1986, p. 96-107.
- [BAD+82] Bolour, A.; Anderson, T.L.; Dekeyser, L.J.; Wong, H.K.T. "The Role of Time in Information Processing: A Survey". *ACM SIGMOD Record*. Vol. 12. Núm. 3. 1982, p. 27-50.

- [BBJ98] Böhlen, M.H.; Busatto, R.; Jensen, C.S. "Point-Versus Interval-Based Temporal Data Models". *Proc. of the ICDE'98*. Orlando, 1998, p. 192-200.
- [BDM88] Bry, F.; Decker, H.; Manthey, R. "A uniform approach to constraint satisfaction and constraint satisfiability in deductive databases". *Proc. of the EDBT'88*. Venecia, 1988, p. 488-505.
- [BFG97] Bertino, E.; Ferrari, E.; Guerrini, G. "T_Quimera: A Temporal Object-Oriented Data Model". *Theory and Practice of Object Systems*. Vol. 3. Núm. 2. 1997, p. 103-125.
- [BJS00] Böhlen, M.H.; Jensen, C.S.; Snodgrass, R.T. "Temporal Statement Modifiers". *ACM Transactions on Database Systems*. Vol. 25. Núm. 4. 2000, p. 407-456.
- [BJW00] Bettini, C.; Jajodia, S.; Wang, S.X. *Time Granularities in Databases, Data Mining and Temporal Reasoning*. Springer, 2000.
- [Böh94a] Böhlen, M.H.. "Managing Temporal Knowledge in Deductive Databases". *Tesi doctoral, Swiss Federal Institute of Technology*. Zurich, 1994.
- [Böh94b] Böhlen, M.H. "Valid Time Integrity Constraints". *Technical Report 94-30*. Department of Computer Science. University of Arizona, 1994.
- [Böh95] Böhlen, M.H. "Temporal Database System Implementations". *ACM SIGMOD Record*. Vol. 24. Núm. 4. 1995, p. 53-60.
- [BR86] Bancilhom, F.; Ramakrishnan, R. "An Amateurs Introduction to Recursive Query Processing Strategies". *Proc. of the ACM SIGMOD Conf. on Management of Data*. Washington D.C., 1986, p. 16-52.
- [BSS96] Böhlen, M.H.; Snodgrass, R.T.; Soo, M.D. "Coalescing in Temporal Databases". *Proc. of the VLDB'96*. Bombay, 1996, p. 180-191.
- [CDI+97] Clifford, J.; Dyreson, C.; Isakowitz, T.; Jensen, C.S.; Snodgrass, R.T. "On the Semantics of 'Now' in Databases". *ACM Transactions on Database Systems*. Vol. 22. Núm. 2. 1997, p. 171-214.
- [CGT90] Ceri, S.; Gottlob, G.; Tanca, L. *Logic Programming and Databases*. Surveys in Computer Science. Springer Verlag, 1990.

-
- [Cho92] Chomicki, J. "History-less Checking of Dynamic Integrity Constraints". *Proc. of the ICDE'92*. Phoenix AZ, 1992, p. 557-564.
- [Cho94] Chomicki, J. "Temporal Query Languages: a Survey". *Proc. of the International Conference on Temporal Logic*. Bonn, 1994, p. 506-534.
- [Cho95] Chomicki, J. "Efficient Checking Encoding of Temporal Integrity Constraints Using Bounded History Encoding". *ACM Transactions on Database Systems*. Vol. 20. Núm. 2. 1995, p. 149-186.
- [CI88] Chomicki, J.; Imielinski, T. "Temporal Deductive Databases and Infinite Objects". *Proc. of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*. Austin, 1988, p. 61-73.
- [CJS+95] Clifford, J.; Jensen, C.S.; Snodgrass, R.T.; Böhlen, M.; Dewan, H.; Smchmidt, D. "Panel: The State-of-the-Art in Temporal Data Management: Perspectives from the Research and Financial Applications Communities". *Proc. of the International Workshop on Temporal Databases*. Zurich, 1995, p. 356-357.
- [CO92] Costal, M.D.; Olivé A. "A Method for Reasoning about Deductive Conceptual Models in Information Systems". *Proc. of the CAISE'92*. Manchester, 1992, p. 612-631.
- [CP00] Cowley, W.; Plexousakis, D. "Temporal Integrity Constraints with Indeterminacy". *Proc. of the VLDB'00*. El Caire, 2000, p. 441-450.
- [CS98] Chomicki, J.; Saake, G. (editors) "*Logics for Databases and Information Systems*". Kluwer, 1998.
- [CT95a] Clifford, J.; Tuzhilin, D. (editors) "*Recent Advances in Temporal Databases*". Springer, 1995.
- [CT95b] Chomicki, J.; Toman, D. "Implementing Temporal Integrity Constraints Using an Active DBMS". *Proc. of the ICDE'95*. Taipei, 1995, p. 566-582.
- [CT98a] Chomicki, J.; Toman, D. "Tutorial: Temporal Databases". *Proc. of the EDBT'98*. Valencia, 1998, p. V (foreword).
- [CT98b] Chomicki, J.; Toman, D. "Temporal Logic in Information Systems". A: [CS98]. p. 31-70.

- [CTB01] Chomicki, J.; Toman, D.; Böhlen, M. H. "Querying ATSQL Databases with Temporal Logic". *ACM Transactions on Database Systems*. Vol. 26. Núm. 2. 2001, p. 145-178.
- [CW83] Clifford, J.; Warren, D.S. "Formal Semantics for Time in Databases". *ACM Transactions on Database Systems*. Vol. 8. Núm. 2. 1983, p. 214-254.
- [Dat00] Date, C. J. "*An Introduction to Database Systems*". 7a ed. Addison-Wesley, 2000.
- [DDL03] Date, C. J.; Darwen H.; Lorentzos, N.A. "*Temporal Databases and the Relational Model*". Morgan Kaufmann Publishers, 2003.
- [Dec89] Decker, H. "The Range Form of Databases or: How to Avoid Floundering". *Proc. of the ÖGAI'89*. Igls, 1989, p. 114-123.
- [Dev97] Devlin, B. "Managing Time in the Data Warehouse". *InfoDB*. Vol. 11. Núm. 1. 1997, p. 7-11.
- [DEL+00] Dyreson, C.E.; Evans, W.S.; Lin H.; Snodgrass, R.T. "Efficiently Supporting Temporal Granularities". *IEEE Transactions on Knowledge and Data Engineering*. Vol. 12. Núm. 4. 2000. p. 568-587.
- [DFG+97] Doucet, A.; Fauvet, M.C.; Garçarski, S.; Jomier, G.; Monties, S. "Using Database Versions to Implement Temporal Integrity Constraints". *Proc. of the Int. Workshop on Constraint Databases*. Delphi, 1997, p. 219-233.
- [DLW84] Dadam, P.; Lum, V.; Werner, H.D. "Integration of Time Versions into a Relational Database System". *Proc. of the VLDB'84*. Singapur, 1984, p. 509-522.
- [DS98] Dyreson, C.E.; Snodgrass, R.T. "Supporting Valid-Time Indeterminacy". *ACM Transactions on Database Systems*. Vol. 23. Núm. 1. 1998, p. 1-57.
- [DW89] Das, S.K.; Williams, M.H. "A Path Finding Method for Constraint Checking in Deductive Databases". *Data & Knowledge Engineering*. Vol. 4. 1989, p. 223-244.

-
- [EJS98] Etzion, O.; Jajodia, S.; Sripada, S. (editors). "*Temporal Databases: Research and Practice*". Springer-Verlag, 1998.
- [EK01] Eder, J.; Koncilia, C. "Changes of Dimension Data in Temporal Data Warehouses". *Proc of the DaWaK'01*. Florència, 2001, p. 284-293.
- [FTU99] Farré, C.; Teniente, E.; Urpí, T. "The Constructive Method for Query Containment Checking". *Proc of the DEXA '99*. Florència, 1999, p. 583-593.
- [Ger96] Gertz, M. "Diagnosis and Repair of Constraint Violations in Database Systems". *Tesi doctoral, Institut für Informatik. Universität Hannover*. Alemanya, 1996.
- [GGG+98] Godfrey, P.; Grant, J.; Gryz, J.; Minker J. "Integrity Constraints: Semantics and Applications". A: [CS98]. p. 265-306.
- [GL95] Gertz, M.; Lipeck, U.W. "'Temporal' Integrity Constraints in Temporal Databases". *Proc. of the International Workshop on Temporal Databases*. Zurich, 1995, p. 77-92.
- [GM78] Gallaire, H.; Minker, J. "*Logic in Data Bases*". Plenum Press, 1978.
- [GM01] Grandi, F.; Mandreoli, F. "Effective Representation and Efficient Management of Indeterminate Dates". *Proc of the TIME'01, Cividale del Friuli*, 2001. p. 164-169.
- [IIS98] Inmon, W. H.; Imhoff, C.; Sousa, R. "*Corporate Information Factory*". 2a ed. John Wiley & Sons, 1998.
- [JCE+93] Jensen, C.S.; Clifford, J.; Elmasri, R.; Gadia, S.K.; Hayes, P.; Jajodia, A. "A Consensus Glossary of Temporal Database Concepts". *Technical Report R 93-2035*. Department of Mathematics and Computer, Institute for Electronic Systems, Aalborg University. 1993.
- [JCG+92] Jensen, C.S.; Clifford, J.; Gadia, S.K.; Segev, A.; Snodgrass, R.T. "A Glossary of Temporal Database Concepts". *ACM SIGMOD Record*. Vol. 21. Núm. 3. 1992, p. 35-43.

- [JDB+98] Jensen, C.S.; Dyreson, C.E. Böhlen, M.H.; Clifford, J; Elmasri, R.; Gadia, S.K.; Grandi, F.; Hayes, P.J.; Jajodia, S.; Käfer, W.; Kline, N.; Lorentzos, N.A.; Mitsopoulos, Y.G.; Montanari, A.; Nonen, D.A.; Peressi, E.; Pernici, B.; Roddick, J.F.; Sarda, N.L.; Scalas, M.R.; Segev, A.; Snodgrass, R.T.; Soo, M.D.; Tansel, A.U; Tiberio, P., Wiederhold, G. "The Consensus Glossary of Temporal Database Concepts - February 1998 Version". A: [EJS98]. p. 367-405.
- [JLV+00] Jarke, M.; Lenzerini, M.; Vassilious, Y.; Vassiliadis, P. "*Fundamentals of Data Warehousing*". Springer-Verlag, 2000.
- [JMS79] Jones, S.; Mason, P.; Stamper, R. "Legol 2.0: A Relational Specification Language for Complex Rules". *Information Systems*. Vol. 4. Núm. 4. 1979, p. 293-305.
- [JS94] Jensen, C.S.; Snodgrass R.T. "Temporal Specialization and Generalization". *IEEE Transactions on Knowledge and Data Engineering*. Vol. 6. Núm. 6. 1994, p. 3-11.
- [JS95] Jensen, C.S.; Snodgrass R.T. "Tutorial: Temporal Databases". *Proc. of the VLDB'95*. Zurich, 1995, p. 3-11.
- [JS96] Jensen, C.S.; Snodgrass R.T. "Semantics of Time-Varying Information". *Information Systems*. Vol. 21. Núm. 4. 1996, p. 311-352.
- [JS99] Jensen, C.S.; Snodgrass R.T. "Temporal Data Management". *IEEE Transactions on Knowledge and Data Engineering*. Vol. 11. Núm. 1. 1999, p. 36-44.
- [JSS92a] Jensen, C.S.; Soo, M.D.; Snodgrass R.T. "Unification of Temporal Data Models". *Technical Report TR 92-15*. Department of Computer Science. University of Arizona. 1992.
- [JSS92b] Jensen, C.S.; Soo, M.D.; Snodgrass R.T. "Extending Normal Forms to Temporal Relations". *Technical Report TR 92-17*. Department Computer Science. University of Arizona, 1992.
- [JSS93] Jensen, C.S.; Soo, M.D.; Snodgrass R.T. "Unification of Temporal Data Models". *Proc. of the ICDE'93*. Viena, 1993, p. 262-271.
- [JSS94] Jensen, C.S.; Soo, M.D.; Snodgrass R.T. "Unifying Temporal Data Models Via a Conceptual Model". *Information Systems*. Vol. 19. Núm. 7. 1994, p. 513-547.

-
- [Kim78] Kimball, K.A. "The DATA System". *Tesi doctoral, Graduate School of Arts and Sciences*. University of Pennsylvania. Pennsilvània, 1978.
- [KRR+98] Kimball, R.; Reeves, L.; Ross, M.; Thornthwaite, W. *"The Data Warehouse Lifecycle Toolkit"*. John Wiley & Sons, 1998.
- [Kli93] Kline, N. "An Update of the Temporal Database Bibliography". *ACM SIGMOD Record*. Vol. 22. Núm. 4. 1993, p. 66-80.
- [Kow78] Kowalski, R. "Logic for Data Description". A: [GM78], p. 77-103.
- [KS86] Kowalski, R.; Sergot, M. "A Logic-Based Calculus of Events" *New Generation Computing*. Vol. 4. Núm. 1. 1986, p. 67-95.
- [KSF+03] Koubarakis, M.; Sellis, T.K.; Frank A.U.; Grumbach, S.; Güting, R.H.; Jensen, C.S.; Lorentzos, N.A.; Manolopoulos, Y.; Nardelli, E.; Pernici, B.; Schek, H.; Scholl, M.; Theodoulidis, B.; Tryfona, N. *"Spatio-Temporal Databases: the CHOROCHRONOS Approach"*. Springer, 2003.
- [KSS87] Kowalski, R.; Sadri, F.; Soper, P. "Integrity Checking in Deductive Databases". *Proc. of the VLDB'87*. Brighton, 1987, p. 61-69.
- [LDE+84] Lum, V.; Dadam, P.; Erbe, R.; Guenauer, J.; Pistor, P.; Walch, G.; Werner, H.; Woodfill, J. "Designing DBMS Support for the Temporal Dimension". *Proc. of the ACM SIGMOD Conf. on Management of Data*. Boston, 1984, p. 115-130.
- [LDE+85] Lum, V.; Dadam, P.; Erbe, R.; Guenauer, J.; Pistor, P.; Walch, G.; Werner, H.; Woodfill, J. "Design of an Integrated DBMS to Support Advanced Applications". *Proc. of the FODO'85*. Kyoto, 1985, p. 31-49.
- [Llo87] Lloyd, J.W. *"Foundations on Logic Programming"*. 2a ed. Springer, 1987.
- [LM97] Lorentzos, N.A.; Mitsopoulos, Y.G. "SQL Extension for Interval Data". *IEEE Transactions on Knowledge and Data Engineering*. Vol. 9. Núm. 3. 1997, p. 480-499.
- [LT84] Lloyd, J.W.; Topor, R.W. "Making Prolog More Expressive". *Journal of Logic Programming*. Vol. 1. Núm. 3. 1984, p. 225-240.

- [Mar98] Martín, C. "The Events Method for Temporal Integrity Constraint Handling in Bitemporal Deductive Databases". *Report de recerca LSI-98-68-R*. Departament de Llenguatges i Sistemes Informàtics. Universitat Politècnica de Catalunya. 1998.
- [Mar00] Martín, C. "Restriccions d'Integritat Temporals en Bases de Dades Deductives Bitemporals". *Report de recerca LSI-00-80-R*. Departament de Llenguatges i Sistemes Informàtics. Universitat Politècnica de Catalunya. 2000.
- [Mar01] Martín, C. "Analyzing Temporal Integrity Constraints to Obtain the Minimum Number of Transition Rules". *Report de recerca LSI-01-52-R*. Departament de Llenguatges i Sistemes Informàtics. Universitat Politècnica de Catalunya. 2001.
- [MA02] Martín, C.; Abelló, A. "The Data Warehouse: A Temporal Database". *Report de recerca LSI-02-66-R*. Departament de Llenguatges i Sistemes Informàtics. Universitat Politècnica de Catalunya. 2002.
- [MA03] Martín, C.; Abelló, A. "A Temporal Study of Data Sources to Load a Corporate Data Warehouse". *Proc. of the DaWaK'03*. Praga, 2003. p. 109-118.
- [MBJ+90] Mylopoulos, J.; Borgida, A.; Jarke, M.; Koubarakis, M. "Telos: Representing Knowledge About Information Systems". *ACM Transactions on Information Systems*. Vol. 8. Núm. 4. 1990, p. 324-362.
- [McD82] McDermott, D.V. "A Temporal Logic for Reasoning about Processes and Plans" *Cognitive Science*. Vol. 6. 1982, p. 101-155.
- [McK86] McKenzie, E. "Bibliography: Temporal Databases". *ACM SIGMOD Record*. Vol. 15. Núm. 4. 1986, p. 40-52.
- [Min88] Minker, J. (editor). "*Foundations of Deductive Databases and Logic Programming*". Morgan Kaufmann. 1988.
- [MS94a] Martín, C.; Sistac, J. "Comprovació de Restriccions d'Integritat en Bases de Dades Deductives Bitemporals". *Report de treball LSI-94-9-T*. Departament de Llenguatges i Sistemes Informàtics. Universitat Politècnica de Catalunya. 1994.

-
- [MS94b] Martín, C.; Sistac, J. "Bases de Dades Bitemporals". *Report de recerca LSI-94-13-R*. Departament de Llenguatges i Sistemes Informàtics. Universitat Politècnica de Catalunya. 1994.
- [MS94c] Martín, C.; Sistac, J. "Integrity Constraints Checking in Historical Deductive Databases". *Proc. of the Fifth Int. Workshop on the Deductive Approach to Information Systems and Databases*. Aiguablava, 1994, p. 299-324.
- [MS96a] Martín, C.; Sistac, J. "A Method for Integrity Constraint Checking in Temporal Deductive Databases". *Proc. of the TIME'96*. Key West, 1996, p. 136-141.
- [MS96b] Martín, C.; Sistac, J. "Applying Transition Rules to Bitemporal Deductive Databases for Integrity Constraint Checking". *Proc. of the Int. Workshop on Logic in Databases*. San Miniato, 1996, p. 117-134.
- [MSS92] Mayol, E.; Sancho, M.R.; Sistac, J. "The ODISSEA project: an environment for the development of Information Systems". *Proc. of the Third Int. Workshop on the Deductive Approach to Information Systems and Databases*. Roses, 1992, p. 17-47.
- [Nic82] Nicolas, J.M. "Logic for Improving Integrity Checking in Deductive Databases". *Acta Informatica*. Vol. 18. Núm. 3. 1982, p. 227-253.
- [NS92] Niézette, M.; Stevenne, J.M. "An Efficient Symbolic Representation of Periodic Data". *Proc. of the CIKM'92*. Baltimore, 1992, p. 161-168.
- [NY78] Nicolas, J.M.; Yazdanian, K. "Integrity Checking in Deductive Databases". A: [GM78], p. 325-344.
- [Oli89] Olivé, A. "On the Design and Implementation of Information Systems from Deductive Conceptual Models". *Proc. of the VLDB'89*. Amsterdam, 1989, p. 3-11.
- [Oli91] Olivé, A. "Integrity Constraints Checking in Deductive Databases". *Proc. of the VLDB'91*. Barcelona, 1991, p. 513-523.
- [Pas97] Pastor, J. A. "Automatic Synthesis of Update Transaction Programs in Deductive Databases". *Tesi doctoral, Departament de Llenguatges i Sistemes Informàtics*. Universitat Politècnica de Catalunya. Barcelona, 1997.

- [PJ98] Pedersen, T.B.; Jensen, C.S. "Research Issues in Clinical Data Warehousing". *Proc. of the SSDBM'98*. Capri, 1998. p. 43-52.
- [Ple93] Plexousakis, D. "Integrity Constraint and Rule Maintenance in Temporal Deductive Knowledge Bases". *Proc. of the VLDB'93*. Dublín, 1993, p. 146-157.
- [Ple96] Plexousakis, D. "On the Efficient Maintenance of Temporal Integrity Constraints in Knowledge Bases". *Tesi doctoral, Department of Computer Science*. University of Toronto. Toronto, 1996.
- [QO93] Quer, C.; Olivé, A. "Object Interaction in Object-Oriented Deductive Conceptual Models". *Proc. of the CAiSE'93*. París, 1993, p. 374-396.
- [SA86] Snodgrass, R.T.; Ahn, I. "Temporal Databases". *IEEE Computer*. Vol. 19. Núm. 9. 1986, p. 35-42.
- [Sho88] Shoham, Y. "*Reasoning about Change: Time and Causation from the Standpoint of Artificial Intelligence*". MIT Press, 1988.
- [SK95] Sadri, F.; Kowalski, R. "Variants of the Event Calculus". *Proc. of the Int. Conf. on Logic Programming*. Tokyo, 1995, p. 67-81.
- [Sch77] Schueler, B. "*Update Reconsidered, in Architecture and Models in Data Base Management Systems*". North Holland Publishing Co. 1977.
- [SJS95] Segev, A.; Jensen, C.S.; Snodgrass, R.T. "Report on the 1995 International Workshop on Temporal Databases". *ACM SIGMOD Record*. Vol. 24. Núm. 4. 1995, p. 46-52.
- [SN97] Steiner, A.; Norrie, M.C. "A Temporal Extensions to a Generic Object Data Model". *Technical Report TR-15*. Time Center. 1997.
- [Sno86] Snodgrass, R.T. "Research Concerning Time in Databases". *ACM SIGMOD Record*. Vol. 15. Núm. 4. 1986, p. 19-52.
- [Sno90] Snodgrass, R.T. "Temporal Databases. Status and Research Directions". *ACM SIGMOD Record*. Vol. 19. Núm. 4. 1990, p. 83-89.

-
- [Sno93] Snodgrass, R.T. (editor) "*Proceedings of the International Workshop on an Infrastructure for Temporal Databases*". Arlington, 1993.
- [Sno95] Snodgrass, R.T. (editor, with multiple other contributions). "*The TSQL2 Temporal Query Language*". Kluwer Academic Publishers. 1995.
- [Sno00] Snodgrass, R.T. "*Developing Time-Oriented Database applications in SQL*". Morgan Kaufmann Publishers. 2000.
- [Soo91] Soo, M.D. "Bibliography on Temporal Databases". *ACM SIGMOD Record*. Vol. 20. Núm. 1. 1991, p. 14-23.
- [S094] Sancho, M.R; Olivé, A. "Deriving Transactions Specifications from Deductive Conceptual Models of Information Systems". *Proc. of the CAiSE'94*. Utrecht, 1994, p. 311-324.
- [Sri91] Sripada, S.M. "Temporal Reasoning in Deductive Databases". *Tesi doctoral, Imperial College of Science and Technology*. University of London. Londres, 1991.
- [Sri95] Sripada, S.M. "Efficient Implementation of the Event Calculus for Temporal Deductive Databases". *Proc. of the Int. Conf. on Logic Programming*. Tokyo, 1995, p. 99-113.
- [SS87] Segev, A.; Shoshani, A. "Logical Modeling of Temporal Data" *Proc. of the ACM SIGMOD Conf. on Management of Data*. San Francisco, 1987. p. 454-466.
- [SS88] Stam, R.B.; Snodgrass, R.T. "A Bibliography on Temporal Databases". *Data Engineering Bulletin*. Vol. 11. Núm. 4. 1988, p. 53-61.
- [ST99] Salzberg, B.; Tsotras, V.J. "A Comparison of Access Methods for Temporal Data". *ACM Computing Surveys*. Vol. 31. Núm. 2. 1999, p. 158-221.
- [TCG+93] Tansel, A.U.; Clifford, J.; Gadia, S.; Jajodia, S.; Segev, A.; Snodgrass, R. "*Temporal Databases: Theory, Design and Implementation*". Benjamin/Cummings. 1993.

- [Ten92] Teniente, E. "El Mètode dels Esdeveniments per a l'Actualització de Vistes en Baese de Dades Deductives". *Tesi doctoral, Departament de Llenguatges i Sistemes Informàtics*. Universitat Politècnica de Catalunya. Barcelona, 1992.
- [TK96] Tsostras, V.J.; Kumar, A. "Temporal Database Bibliography Update". *ACM SIGMOD Record*. Vol. 25. Núm. 1. 1996, p. 41-51.
- [TM99] Teniente E.; Mayol E. "Addressing Efficiency Issues During the Process of Integrity Maintenance". *Proc. of the DEXA'99*. Florència, 1999, p. 270-281.
- [T092] Teniente E.; Olivé A. "The Events Method for View Updating in Deductive Databases". *Proc. of the EDBT'92*. Viena, 1992, p. 245-260.
- [Tom97] Toman, D. "A Point-based Temporal Extensions of SQL". *Proc. of the DOOD '97*. Montreal, 1997, p. 103-121.
- [TS04] Terenziani, P.; Snodgrass, R.T. "Reconciling Point-Based and Interval-Based Semantics in Temporal Relational Databases: A Treatment of the Telic/Atelic Distinction". *IEEE Transactions on Knowledge and Data Engineering*. Vol. 16. Núm. 5. 2004, p. 540-551.
- [Ull88] Ullman, J.D. *Principles of Database and Knowledge-Base Systems, Volume I*. Computer Science Press, 1988.
- [UTP+94] Urpí, T.; Teniente, E.; Pastor, J.A.; Mayol, E.; Martín, C. "FOLRE: Towards a System for the Integrated Treatment of Updates and Rule Enforcement in Deductive Databases". *Proc. of the Sixth ERCIM Database Research Group Workshop on Deductive and Interoperable Databases*. Barcelona, 1994, p. 129-147.
- [Urp93] Urpí, A. "El Mètode dels Esdeveniments Interns per al Càlcul de Canvis en Bases de Dades Deductives". *Tesi doctoral, Departament de Llenguatges i Sistemes Informàtics*. Universitat Politècnica de Catalunya. Barcelona, 1993.
- [U092] Urpí T.; Olivé A. "A Method for Change Computation in Deductive Databases". *Proc of the VLDB '92*. Vancouver, 1992, p. 225-237.

- [Vas00] Vassiliadis, P. "Gulliver in the Land of Data Warehousing: Practical Experiences and Observations of a Researcher". *Proc. of the DMDW'00*. Estocolm, 2000, CEUR-WS.
- [VDD95] Van Belleghem, K.; Denecker, M.; De Schreye, D. "Combining Situation Calculus and Event Calculus". *Proc. of the Int. Conf. on Logic Programming*. Tokyo, 1995, p. 83-97.
- [WFW75] Wiederhold, G.; Fries, J.F.; Weyl, S. "Structured Organization of Clinical Data Bases". *Proc. of the AFIPS National Computer Conference*. Anaheim, 1975, p. 479-485.
- [Wij99] Wijzen, J. "Temporal FDs on Complex Objects". *ACM Transactions on Database Systems*. Vol. 24. Núm. 1. 1999, p. 126-176.
- [WJW98] Wu, Y.; Jajodia, S.; Wang, X.S. "Temporal Database Bibliography Update". A: [EJS98], p. 338-366.
- [Wüt91] Wüthrich, B. "Large Deductive Databases with Constraints". *Tesi doctoral, Swiss Federal Institute of Technology*. Zurich, 1991.
- [YW00] Yang, J.; Widom, J.; Snodgrass R.T. "Temporal View Self-Maintenance in a Warehousing Environment". *Proc. of the EDBT'00*. Konstanz, 2000, p. 395-412.

UNIVERSITAT POLITÈCNICA DE CATALUNYA
DEPARTAMENT DE LENGUATGES I SISTEMES INFORMÀTICS

**AN INTEGRITY CONSTRAINT CHECKING METHOD FOR
BITEMPORAL DEDUCTIVE DATABASES**

PH.D. THESIS SUMMARY REPORT FOR THE EUROPEAN DOCTORATE
MENTION

ADVISED BY DR. NÚRIA CASTELL ARIÑO

BARCELONA

2005

An Integrity Constraint Checking Method for Bitemporal Deductive Databases

Carme Martín Escofet

Departament de Llenguatges i Sistemes Informàtics,
Universitat Politècnica de Catalunya.
Jordi Girona Salgado 1-3. E-08034 Barcelona, Catalunya.
martin@lsi.upc.es

Ph.D. thesis advisor: Núria Castell Ariño

Abstract

This Ph.D. thesis mainly deals with temporal and deductive databases areas and *the aim is to obtain a method for integrity constraint checking in bitemporal deductive databases*, analyzing the temporal integrity constraint behavior. The existence of a transaction time dimension is especially useful to reduce the number of transition rules. In the valid time dimension, the temporal dependence of database predicate terms has been explored. Moreover, this Ph.D. thesis provides some research advances bringing together the areas of temporal databases and data warehouses. Chapter 1 is an introduction to the problem and the related work of this Ph.D. thesis main area. Chapter 2 shortly presents temporal databases, and chapter 3 deductive databases. Chapter 4 introduces the study of temporal deductive database transaction behavior, explaining the concept of coherent transaction. This concept is used to reject transactions without more information requirements. Chapter 5 defines a new taxonomy of temporal integrity constraints to achieve a better understanding of why constraints are required and a best way to define them using first-order logic. In order to meet these goals, we have studied temporal integrity constraint taxonomies existing on the temporal database area and other deeply related areas such as multiversion databases. These works have been adapted and further developed to cover the scope of bitemporal deductive databases. Chapter 6 proposes an integrity constraint checking method for bitemporal deductive databases that derives a set of transition rules for each temporal integrity constraint. Then, Standard Linear Deduction with Negation as Failure Resolution is used to verify that a transaction does not violate any constraint before the transaction is allowed to commit. The set of transition rules obtained is very large; therefore, a carefully analysis of the integrity constraints using the taxonomy introduced above drastically reduces the number of rules. This chapter includes some examples of this new reduced set of transition rules and application examples of our integrity constraint checking method in bitemporal deductive databases. In chapters 7 and 8, we present our current and further work. In chapter 7, we identify a new type of predicate term dependent of time. This new type of term, named temporally dependent term, is formally defined and deeply studied in order to find its properties and its applicability to queries and integrity constraints. In chapter 8, we explain how to combine temporal database and data warehouse areas. Finally, chapter 9 provides conclusions and further work.

Introduction

In this chapter, the state-of-the-art of temporal database research area is presented, following the [Sno90] idea of a chronological development. The different existing bibliographies on temporal databases are remembered: [BAD+82], [McK86], [Sno86], [SS88], [Sno90], [Soo91], [Kli93], [TK96] and [WJW98]. The historical explanation starts with one of the first papers on temporal databases [WFW75] and finishes with the CHOROCHRONOS project [KSF+03]. We can summarize the history saying that during the seventies the first valid time Data Base Management System (DBMS) and transaction time DBMS prototypes were implemented. During the eighties the first prototype of a bitemporal DBMS is implemented and started the work on temporal deductive databases. From the nineties on, the focus was on design and language standards, implementation improvements and clarification of the cooperation between temporal and deductive databases. Especially remarkable facts are the first glossary on temporal database concepts [JCG+92], the first book on temporal databases [TCG+93], the first book on the temporal database language TSQL2 [Sno95] and the creation of the TimeCenter (<http://www.cs.auc.dk/TimeCenter/>), an international center for the support of temporal database applications on traditional and emerging DBMS technologies.

In this survey, we notice a lack in temporal integrity constraint checking methods for bitemporal deductive databases that it is the main goal of our Ph.D. thesis. Our method is explained in chapter 6.

Ph.D. thesis objectives

The main goals of this Ph.D. thesis are:

- A new method proposal for integrity constraint checking in bitemporal deductive databases based on the use of Standard Linear Deduction with Negation as Failure (SLDNF) Resolution.
- The study of temporal deductive database transaction behavior, introducing the concept of coherent transaction.
- The formalization of a new taxonomy of temporal integrity constraints to achieve a better understanding of why constraints are required and a best way to define them using first-order logic.
- The idea of considering powerful temporal integrity constraints brings us to study the temporal dependence of the database predicate terms.
- The two temporal dimension analysis developed for our integrity constraint checking method motivates the relationship between bitemporal databases and other areas that recognize the need to incorporate two time dimensions. Specifically, we combine temporal database and data warehouse areas, due to the mutual benefits of the advances in each line. Moreover, a bitemporal storage structure for data warehouses is proposed.

Document organization

In order to better understand the integrity constraint checking method for bitemporal deductive databases presented in chapter 6, chapter 2 presents temporal databases and chapter 3 deductive databases. Chapter 4 introduces the study of temporal deductive database transaction behavior, explaining the concept of coherent transaction. This concept is used to reject transactions without more information requirements. Chapter 5 defines a new taxonomy of temporal integrity constraints to achieve a better understanding of why constraints are required and a best way to define them using first-order logic. In order to meet these goals, we have studied temporal integrity constraint taxonomies existing on the temporal database area and other deeply related areas such as multiversion databases. These works have been adapted and further developed to cover the scope of bitemporal deductive databases. Chapter 6 proposes an integrity constraint checking method for bitemporal deductive databases that derives a set of transition rules for each temporal integrity constraint. Then, SLDNF-resolution is used to verify that a transaction does not violate any constraint before the transaction is allowed to commit. The set of transition rules obtained is very large; therefore, a carefully analysis of the integrity constraints using the taxonomy introduced in chapter 5 drastically reduces the number of rules. This chapter includes some examples of this new reduced set of transition rules and application examples of our integrity constraint checking method in bitemporal deductive databases.

Moreover, in the last years our research that has carried on other directions, always connected with the temporal database area, have obtained first results. The idea of considering powerful temporal integrity constraints brings us to study the temporal dependence of the database predicate terms. In chapter 7, we identify a new type of predicate term dependent of time. This new type of term, named temporally dependent term, is formally defined and deeply studied in order to find its properties and its applicability to queries and integrity constraints. The two temporal dimension analysis developed for our integrity constraint checking method motivates the relationship between bitemporal databases and other areas that recognize the need to incorporate two time dimensions. In chapter 8, we explain how to combine temporal database and data warehouse areas, due to the mutual benefits of the advances in each line. Moreover, a bitemporal storage structure for data warehouses is proposed. Chapter 9 concludes the Ph.D. thesis listing the main contributions and explaining the state of our current and future research.

Conclusions and further work

In this chapter, we present the main contributions of this Ph.D. thesis and explain the state of our current and future research.

Conclusions

The aim of this Ph.D. thesis is a new integrity constraint checking method proposal for bitemporal deductive databases. As we explain in the introduction chapter, we notice a lack in integrity constraint checking methods for bitemporal deductive databases. The first approach of our method, based on the *Internal Events Method* [Oli91], has been introduced in [MS94c]. Before that, we study the temporal database area [MS94b], and the existing integrity constraint checking methods for bitemporal deductive databases [MS94a]. In [MS96a], we present our method for only a valid time dimension and in [MS96b] for a bitemporal dimension with very restrictive assumptions. In [Mar98], we introduce the idea of our method evolution using a case study of temporal integrity constraints. In [Mar00], we explain the temporal integrity constraint taxonomy used in this report. Our transition rules range over all the possible cases in which an update could violate a temporal integrity constraint. For this reason, we obtain a big set of transition rules. Therefore, in [Mar01] we apply the obtained results from temporal integrity constraint taxonomy [Mar00] to our transition rules to simplify or eliminate them.

The main **contributions** of this Ph.D. thesis are:

- The utilization of a first-order logic language to define and manipulate transactions and temporal integrity constraints.
- The definition of coherent transaction concept.
- The formalization of coalescing for bitemporal rectangles in bitemporal deductive databases.
- A new taxonomy of temporal integrity constraints for bitemporal deductive databases.
- The definition of two new predicates (ι^* and δ^*) that automatically incorporate transaction time.
- The new method proposal based on the use of SLDNF-resolution.
- An important simplification of transition rules based on the taxonomy of temporal integrity constraints for bitemporal deductive databases.

Transaction time dimension analysis is the main reason to reduce the number of transition rules. Concentrating in the valid time dimension, we have explained the special characteristics of temporally dependent predicate terms and we have introduced the idea of the completed database that provides information for all the valid time subintervals. In the completed database, the database information gaps are covered, all the subintervals contained in a time interval are generated, and the corresponding new information for these subintervals is deduced. Having

temporally dependent terms, we have motivated the need to have an appropriate language to manipulate them. We have defined a set of operations to handle completed databases with temporally dependent and independent terms and we have analyzed properties and usefulness of its application into temporally dependent queries and temporally integrity constraints.

In the corporate data warehouse, we have identified the two existing orthogonal temporal dimensions: valid time dimension and transaction time dimension. In this bitemporal data warehouse environment, we have used our bitemporal storage structure, named *Current/Historical*, to represent all the temporal data source knowledge obtained by the different kinds of data sources.

Further work

The method version presented in [MS96b] has been implemented in a *Prolog* system. After that, the high cost of transition rule generation motivates us to continue working on simplify transition rules before to develop a new prototype.

As further work, our method can be extended, for example with a deeply study of integrity constraints with derived predicates, that is only sketched in this PhD thesis, and start working with recursion. Moreover, other integrity constraint checking techniques, as graphs [Ger96], can be combined with our method.

In the chapters 7 and 8, we have presented the first results obtained from our current research, but both chapters are clearly part of our future research. At this moment, we are working in the definition of new temporally dependent statements to be incorporated into the temporal language ATSQL [BJS00]. These new statements will be implemented into the stratified temporal database system TIGER (<http://www.cs.aau.dk/~tigeradm/>) to support temporally dependent and independent information. About bitemporal data warehouses, we are implementing the *Current/Historical* structure and the three most used bitemporal representation structures. Currently, we are preparing a comparative to prove that *Current/Historical* structure is the best for a data warehouse environment.