UNIVERSITAT POLITÈCNICA DE CATALUNYA

Departament de llenguatges i sistemes informàtics

ACTUALITZACIÓ CONSISTENT DE BASES DE DADES DEDUCTIVES

Autor: Enric Mayol Sarroca
Director: Ernest Teniente i López

Barcelona, 2000

4. Definició del Mètode

El mètode presentat en aquest capítol, és un mètode definit per a l'actualització consistent de bases de dades deductives. En particular, aquest mètode està definit per a l'actualització de vistes i el manteniment de restriccions d'integritat en bases de dades deductives. En aquest sentit, donada una petició d'actualització que pot incloure tant peticions d'actualització de fets bàsics com de fets derivats, aquest mètode obté les diferents maneres d'actualitzar la base de dades extensional per a què es satisfaci la petició d'actualització i no es violi cap restricció d'integritat.

En aquest capítol, es presenta la descripció del mètode basada en exemples, la definició formal del mètode i la demostració de correctesa i completesa del mètode. Els aspectes relacionats amb l'eficiència del mètode proposat seran analitzats als capítols 6 i 7 d'aquesta tesi.

El mètode presentat en aquesta tesi és una extensió del Mètode dels Esdeveniments presentat a [Ten92, TO95]. La principal diferència entre els dos mètodes, a nivell de definició, rau en els tipus d'actualitzacions que pot tractar cada mètode. El Mètode dels Esdeveniments només considera dos tipus d'actualitzacions: la inserció i l'esborrat. En canvi, el nostre mètode permet tractar insercions, esborrats i modificacions com a actualitzacions bàsiques. La incorporació de la modificació com a un nou tipus d'actualització comporta un canvi en la semàntica de les insercions i esborrats, i per tant, requereix d'una redefinició d'aquestes actualitzacions i del propi mètode. Una comparació més detallada dels dos mètodes es troba descrita a la secció 5.4.

A l'igual que el Mètode dels Esdeveniments [Ten92, TO95], el nostre mètode es basa en la utilització de les regles de transició i les regles d'esdeveniments de la base de dades augmentada A(D). La base de dades augmentada que utilitza el nostre mètode és, òbviament, diferent a la utilitzada pel Mètode dels Esdeveniments. En el nostre cas utilitzem la base de dades augmentada definida al capítol 3 d'aquest document i que està basada en la definida a [Urp93].

Aquest capítol 4 està dividit en les seccions següents. A la secció 4.1 es realitza una descripció general del mètode i dels problemes que tracta. A la secció 4.2 es presenten alguns exemples en els que es mostra el funcionament i les principals característiques del mètode. A la secció 4.3 es presenta la definició formal del mètode i, finalment, a la secció 4.4 es presenta la demostració de correctesa i completesa del mètode.

4.1 Descripció del mètode

El propòsit del mètode que presentem en aquesta tesi és l'actualització consistent de base de dades deductives. Davant d'una petició d'actualització U, el mètode obté totes les possibles formes d'actualitzar la base de dades extensional per a satisfer la petició d'actualització, sense violar cap restricció d'integritat. Si alguna restricció d'integritat es veu afectada per la petició d'actualització, el propi mètode proposa actualitzacions addicionals per a evitar aquestes violacions i seguir satisfent la petició inicial d'actualització U. Sempre que, per a reparar una restricció d'integritat o per a satisfer la petició d'actualització U, calgui actualitzar algun fet derivat, el propi mètode tradueix aquesta petició d'actualització en actualitzacions sobre fets bàsics.

Tota actualització de la base de dades deductiva es representa mitjançant un esdeveniment. Els esdeveniments bàsics representen insercions, esborrats o modificacions dels fets de la base de dades extensional. Els esdeveniments derivats són actualitzacions sobre predicats derivats (o vistes) que cal traduir en esdeveniments bàsics, és a dir, en actualitzacions sobre els fets de la base de dades extensional. Pels predicats d'inconsistència, al considerar que en l'estat antic de la base de dades no es viola cap restricció d'integritat, només tenim definit l'esdeveniment d'inserció que indica una violació de la corresponent restricció d'integritat i que cal evitar induir a l'actualitzar la base de dades.

En el nostre mètode, la transició des de l'estat antic de la base de dades cap al nou estat de la base de dades (actualitzat) és deguda a l'ocurrència d'un conjunt d'actualitzacions de fets bàsics de la base de dades extensional. A aquest conjunt d'actualitzacions ens hi referirem com a Transacció. Així doncs, una transacció es representarà com un conjunt d'esdeveniments bàsics instanciats que es poden aplicar a la base de dades, és a dir, que la base de dades satisfà els prerequisits de cada esdeveniment i que no són mútuament exclusius entre ells, tal com estableixen les definicions 3.2 i 3.3 del capítol anterior.

Definició 4.1: Una *transacció* és un conjunt d'esdeveniments bàsics positius i instanciats que no són mútuament exclusius i pels que es satisfan els seus prerequisits.

En el nostre mètode, una petició d'actualització U consisteix en un conjunt d'esdeveniments instanciats positius que representen l'actualització que es vol aplicar a la base de dades deductiva. Aquesta petició és múltiple i mixta, en el sentit que pot contenir més d'un esdeveniment i tant poden ser esdeveniments d'inserció, d'esborrat com de modificació. Això és degut a que aquests tres tipus esdeveniments són gestionats de la mateixa manera pel nostre mètode. A més a més, aquests esdeveniments poden ser esdeveniments bàsics o esdeveniments derivats. Per altra banda, en la petició d'actualització U, també s'hi poden incloure esdeveniments negats, els quals representen actualitzacions que es vol evitar induir amb l'actualització de la base de dades.

Exemple 4.1: Suposem una petició d'actualització que inclou els esdeveniments següents: $\iota P(\underline{a},1)$, $\mu R(\underline{b},2,3)$ i $\neg\delta Q(\underline{a},2)$. Cada un d'aquest esdeveniments es corresponen a una petició per a inserir el fet $P(\underline{a},1)$, per a modificar el fet $R(\underline{b},2)$ pel nou valor $R(\underline{b},3)$ i, per a evitar induir l'esborrat del fet $Q(\underline{a},2)$ a l'actualitzar la base de dades extensional, respectivament.

□

Definició 4.2: Una *petició d'actualització U* es representa per una conjunció d'esdeveniments bàsics i derivats, positius i negatius. Els esdeveniments positius es corresponen a actualitzacions a aplicar, i els esdeveniments negatius a actualitzacions que cal evitar.

Exemple 4.2: La petició d'actualització associada a l'exemple 4.2 es representa, en el nostre mètode, per la conjunció següent: U = ιP(a,1) ∧ μR(b,2,3) ∧ ¬δQ(a,2).

□

La possibilitat d'incloure esdeveniments negats a una petició d'actualització és especialment útil en el cas de voler evitar violar alguna de les restriccions d'integritat de la base de dades. Tenint en compte que, quan una transacció T viola alguna restricció d'integritat de la base de dades s'indueix l'esdeveniment ιIc. Llavors, per tal de prevenir la violació de totes les restriccions d'integritat i, així, assegurar la consistència del nou estat de la base de dades cal estendre tota petició d'actualització U amb l'esdeveniment negat ¬ιIc.

Donada una petició d'actualització U, el nostre mètode obté totes les solucions que satisfan la petició inicial U i no violen cap restricció d'integritat. Cada una de les solucions obtingudes es correspon a una transacció. En cas d'existir més d'una solució a una petició U, cal escollir-ne només una per a aplicar a la base de dades. En el nostre treball no proposem cap tècnica específica per a determinar quina de les solucions és la més adequada, sinó que aquesta decisió es delega a l'usuari. En el nostre enfocament requerim que les solucions obtingudes siguin mínimes, és a dir, que donada una solució S no existeixi una altre solució S1 que sigui un subconjunt de S. En determinats casos, el nostre mètode pot obtenir solucions que no siguin mínimes, per tant, es requerirà un procés final de purgat d'aquestes solucions.

Definició 4.3: Siguin U una petició d'actualització i T una transacció. Direm que T és una *solució* de U si i sols si T satisfà la petició U i no viola cap restricció d'integritat. A més a més, direm que T és una *solució mínima* de U si T és solució de U i no existeix cap subconjunt de T que sigui solució de U.

Exemple 4.3: Consideri's la base de dades de l'exemple 3.1 amb la EDB = ∅. Siguin la petició d'actualització U = ιEmp(Pere, UPC). La transacció T = {ιTreb(Pere, UPC), ιCont(Pere, UPC), ιEdat(Pere), ιProp(Pere, AB)}és una solució a U, ja que satisfà U i no viola cap restricció d'integritat. En canvi, la transacció T no és una solució mínima de U, ja que la transacció T' = T - {ιProp(Pere, AB)} és també solució de U. En canvi, T' sí que és solució mínima de la petició U.

□

En el nostre mètode, les regles d'esdeveniment i de transició de la base de dades augmentada A(D) s'interpreten de forma descendent (*downward*, en anglès), és a dir, s'utilitzen per a propagar les actualitzacions dels predicats derivats en actualitzacions de predicats bàsics. Concretament, aquestes regles s'utilitzen per a traduir les peticions d'actualització de fets derivats en actualitzacions de fets bàsics i per a reparar restriccions d'integritat proposant actualitzacions addicionals que evitin induir l'esdeveniment ιIc.

El procés per a obtenir una solució de la petició d'actualització U, està basat en el mecanisme de resolució SLDNF. Una transacció T satisfà la petició d'actualització U i no viola cap restricció d'integritat si, utilitzant la resolució SLDNF, l'objectiu $\{\leftarrow U \wedge \neg\, \iota Ic\}$ té èxit considerant el conjunt d'entrada $A(D) \cup T$. Així doncs, la transacció T s'obté fent que alguna derivació fracassada d' $A(D) \cup \{\leftarrow U \wedge \neg\, \iota Ic\}$ tingui èxit. Això, s'aconsegueix incloent en el conjunt d'entrada T una instància de cada esdeveniment bàsic positiu que ocorre en l'objectiu de la branca fracassada.

Si s'assoleix la clàusula buida en el procés anterior, aleshores els esdeveniments del conjunt T es consideren una solució de la petició d'actualització U. Les diferents maneres d'assolir la clàusula buida, fent tenir èxit a les derivacions fracassades, es corresponen a les diferents solucions T_i de la petició d'actualització U.

En el cas de no obtenir cap solució, la petició d'actualització U no es podrà satisfer actualitzant únicament la base de dades extensional. En aquest cas, per a poder satisfer la petició U i no violar cap restricció d'integritat s'haurien de realitzar canvis en la base de dades intensional modificant les regles deductives i/o les restriccions d'integritat.

4.2 Exemples

Amb els exemples que es mostren en aquesta secció es pot veure el funcionament general del nostre mètode. Els exemples que es presenten són quatre. El primer d'ells mostra com es pot traduir una actualització de vistes en actualitzacions de fets bàsics. El segon exemple mostra com es pot reparar una restricció d'integritat al ser violada. En el tercer exemple la petició d'actualització és múltiple. En el quart exemple, es mostra com al reparar una restricció d'integritat es poden violar altres restriccions que ja han estat comprovades prèviament.

La base de dades deductiva que farem servir és la descrita a l'exemple 3.1. Però de totes maneres, a continuació mostrem únicament les regles de la base de dades augmentada rellevants als exemples que es mostraran en aquesta secció.

(F1)	Treb(Núria, Za)	(F2)	Cont(Núria, Za)	(F3)	Edat(Núria)
(F4)	Treb(Sílvia, SJD)	(F5)	Treb(Toni, AIC)	(F6)	Cont(Toni, AIC)
(F7)	Edat(Toni)	(F8)	Baixa(Toni)	(F9)	Treb(Mercè, EDM)

(R1) $Emp(p,c) \leftarrow Treb(p,c) \wedge Cont(p,c)$

(R2) $Actiu(p) \leftarrow Emp(p,c) \wedge \neg Baixa(p)$

(R3) $Contractat(p) \leftarrow Cont(p,c)$

(R8) $Ic1(p,c) \leftarrow Emp(p,c) \wedge \neg Edat(p)$

(R9) $Ic2(p,c,s) \leftarrow Sou(p,c,s) \wedge \neg Treb(p,c)$

(R10) $Ic3(p,n) \leftarrow Numss(p,n) \wedge \neg Contractat(p)$

(I1) $\iota Emp(\underline{p},c) \leftarrow Treb(\underline{p},c) \wedge \neg\delta Treb(\underline{p},c) \wedge \neg Aux1(\underline{p},c) \wedge \neg Aux9(\underline{p}) \wedge \iota Cont(\underline{p},c)$

(D1) $\delta Emp(\underline{p},c) \leftarrow Treb(\underline{p},c) \wedge \delta Treb(\underline{p},c) \wedge Cont(\underline{p},c)$

(D2) $\delta Emp(\underline{p},c) \leftarrow Treb(\underline{p},c) \wedge \mu Treb(\underline{p},c,c1) \wedge c1{\neq}c \wedge Cont(\underline{p},c) \wedge \neg\mu Cont(\underline{p},c,c1)$

(D3) $\delta Emp(\underline{p},c) \leftarrow Treb(\underline{p},c) \wedge Cont(\underline{p},c) \wedge \delta Cont(\underline{p},c)$

(D4) $\delta Emp(\underline{p},c) \leftarrow Treb(\underline{p},c) \wedge Cont(\underline{p},c) \wedge \mu Cont(\underline{p},c,c1) \wedge c1{\neq}c \wedge \neg\mu Treb(\underline{p},c,c1)$

(M1) $\mu Emp(\underline{p},c1,c2) \leftarrow Treb(\underline{p},c1) \wedge \mu Treb(\underline{p},c1,c2) \wedge Cont(\underline{p},c1) \wedge \mu Cont(\underline{p},c1,c2) \wedge c1{\neq}c2$

(I9) $\iota Actiu(\underline{p}) \leftarrow Emp(\underline{p},c) \wedge \neg\delta Emp(\underline{p},c) \wedge Baixa(\underline{p}) \wedge \delta Baixa(\underline{p})$

(I13) $\iota Contractat(\underline{p}) \leftarrow \neg Aux9(\underline{p}) \wedge \iota Cont(\underline{p},c)$

(C01) $\iota Ic \leftarrow \iota Ic1(\underline{p},c)$ (C02) $\iota Ic \leftarrow \iota Ic2(\underline{p},c,s)$ (C03) $\iota Ic \leftarrow \iota Ic3(\underline{p},n)$

(C2) $\iota Ic1(\underline{p},c) \leftarrow \neg Aux11(\underline{p}) \wedge \iota Emp(\underline{p},c) \wedge \neg Edat(\underline{p}) \wedge \neg\iota Edat(\underline{p})$

(C6) $\iota Ic2(\underline{p},c,s) \leftarrow \neg Aux13(\underline{p}) \wedge \iota Sou(\underline{p},c,s) \wedge \neg Treb(\underline{p},c) \wedge \neg\iota Treb(\underline{p},c) \wedge \neg Aux8(\underline{p},c)$

(C7) $\iota Ic2(\underline{p},c,s) \leftarrow \neg Aux13(\underline{p}) \wedge \iota Sou(\underline{p},c,s) \wedge Treb(\underline{p},c) \wedge \delta Treb(\underline{p},c)$

(C13) $\iota Ic3(\underline{p},n) \leftarrow \neg Aux14(\underline{p}) \wedge \iota Numss(\underline{p},n) \wedge \neg Contractat(\underline{p}) \wedge \neg\iota Contractat(\underline{p})$

(A8) $Aux8(\underline{p},c) \leftarrow \mu Treb(\underline{p},c1,c)$ (A9) $Aux9(\underline{p}) \leftarrow Cont(\underline{p},c)$

(A11) $Aux11(\underline{p}) \leftarrow Emp(\underline{p},c)$ (A13) $Aux13(\underline{p}) \leftarrow Sou(\underline{p},c,s)$

(A14) $Aux14(\underline{p}) \leftarrow Numss(\underline{p},n)$

En cada un dels exemples, es mostren les branques més rellevats de l'arbre de resolució associat. En cada pas, el literal seleccionat està ressaltat amb negreta i, indicarem amb una etiqueta encerclada la regla del mètode que s'ha utilitzat. Aquestes regles estan definides en la secció 4.3 on es formalitza el mètode. A més a més, a cada pas també s'indicarà entre parèntesis l'identificador de la regla de la base de dades augmentada que s'hagi utilitzat (si és el cas).

4.2.1 Actualització d'un predicat derivat (sense restriccions d'integritat)

En aquest primer exemple, i per tal de reduir la seva complexitat, suposarem que no es viola cap restricció d'integritat, i per tant, no comprovarem la seva consistència.

En realitat, en aquesta secció 4.2.1 mostrarem dos exemples. Els dos consistiran en l'actualització d'un predicat derivat. En el primer d'ells sols obtindrem una solució, en canvi en el segon n'obtindrem diverses.

Exemple a): Suposem que la petició d'actualització U consisteix en la modificació del fet que la Núria és una empleada de l'empresa Za i passa a ser-ho de l'empresa Al. La petició d'actualització serà doncs U = μEmp(<u>Núria</u>, Za, Al)[1]. A la Figura 4.1.a es mostra l'arbre de resolució associat a aquesta petició d'actualització.

[1] Observi's que en el cas de voler assegurar la no violació de les restriccions d'integritat, la petició U es correspondria a U = μEmp(<u>Núria</u>, Za, Al) $\wedge \neg\iota$Ic

$\leftarrow$ **μEmp(Núria, Za, Al)**

A.1 | 1 (M1)

$\leftarrow$ **Treb(Núria,Za)** $\wedge$ μTreb(Núria,Za,Al) $\wedge$
Cont(Núria, Za) $\wedge$ μCont(Núria, Za, Al) $\wedge$ Za $\neq$ Al

A.1 | 2 (F1)

$\leftarrow$ **μTreb(Núria,Za,Al)** $\wedge$ Cont(Núria, Za) $\wedge$
μCont(Núria, Za, Al) $\wedge$ Za $\neq$ Al

A.2 | 3 T = {μTreb(Núria,Za,Al)}

$\leftarrow$ **Cont(Núria, Za)** $\wedge$ μCont(Núria, Za, Al) $\wedge$ Za $\neq$ Al

A.1 | 4 (F2)

$\leftarrow$ **μCont(Núria, Za, Al)** $\wedge$ Za $\neq$ Al

A.2 | 5 T = T $\cup$ {μCont(Núria,Za,Al)}

$\leftarrow$ **Za $\neq$ Al**

A.1 | 6

[ ]

Figura 4.1.a Arbre de derivació de l'exemple a)

El passos 1 i 2 es corresponen a passos de resolució SLDNF on la base de dades augmentada A(D) actua com a conjunt d'entrada. En el primer pas, el literal seleccionat es resol amb la regla d'esdeveniment M1, mentre que en el pas 2 es considera el fet de la base de dades extensional Treb(Núria, Za).

En el tercer pas, el literal seleccionat és un esdeveniment bàsic positiu i totalment instanciat. Així doncs, per tal d'assolir l'èxit en aquesta branca, s'inclou aquest esdeveniment al conjunt T i es continua amb la resta de literals.

El pas 4 és similar al pas 2, i el pas 5 similar al 3. Per tant, en el pas 4 es resol com un pas de resolució SLDNF amb la A(D) com a conjunt d'entrada, i en el pas 5 s'inclou l'esdeveniment seleccionat al conjunt T.

Finalment, en el pas 6 s'avalua una expressió de comparació. El resultat d'aquesta expressió és cert i s'assoleix la clàusula buida ([]).

Un cop assolida la clàusula buida, la branca ha assolit l'èxit i, per tant, obtenim una solució a la petició d'actualització U = μEmp(Núria, Za, Al). Aquesta solució es correspon als esdeveniments bàsics acumulats al conjunt T = {μTreb(Núria, Za, Al), μCont(Núria, Za, Al)}. Aquesta solució consisteix en la modificació de l'empresa on treballa i està contractada la Núria: passant de treballar i estar contractada de l'empresa Za a l'empresa Al. Es pot comprovar fàcilment que aquesta solució és mínima i que és l'única forma de satisfer la petició inicial U actualitzant solament la base de dades extensional.

□

Exemple b): Suposem que la petició d'actualització consisteix en l'esborrat del fet que la Núria és una empleada de l'empresa Za. La petició d'actualització es correspon a l'esdeveniment U = δEmp(Núria, Za). Cada una de les solucions que satisfan aquesta petició s'obtenen fent que una derivació fracassada d'A(D) ∪ {← U} tinguin èxit. A la Figura 4.1.b es mostra part de l'arbre de resolució associat a aquesta petició d'actualització.

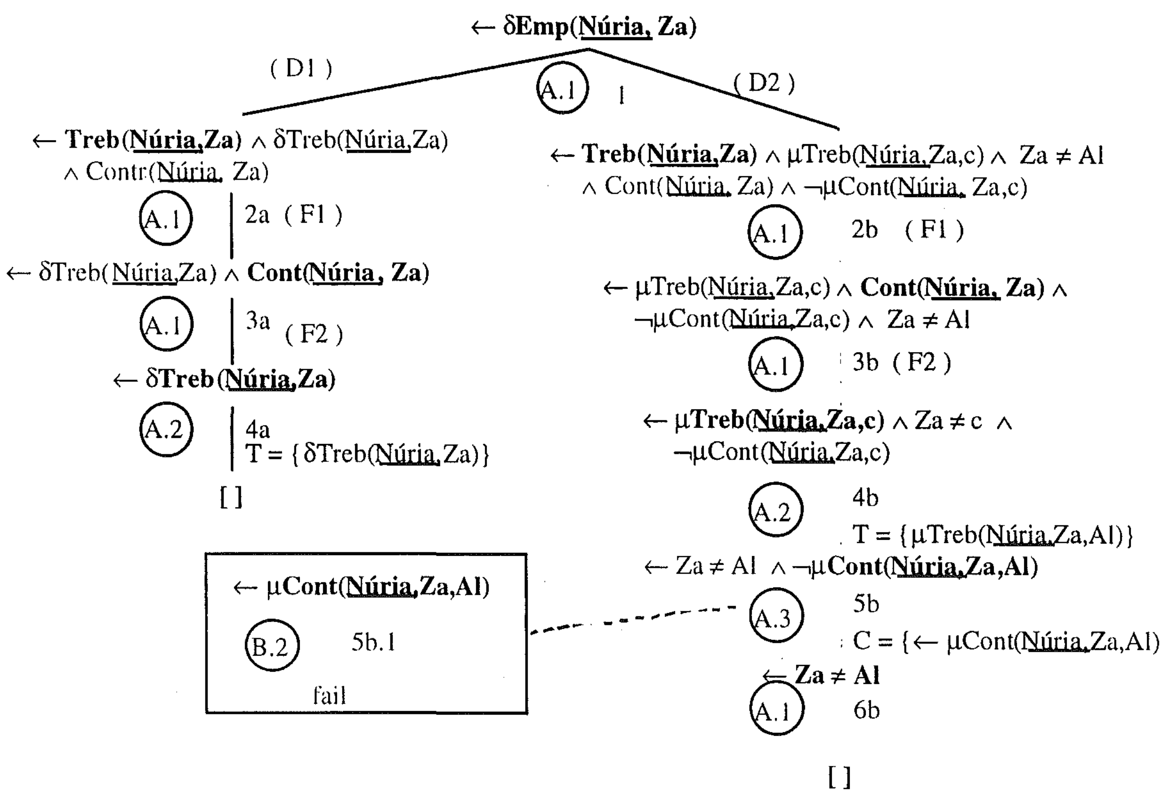

Figura 4.1.b Arbre de derivació de l'exemple b)

El pas 1 de l'arbre de derivació d'aquest exemple es correspon a un pas de resolució SLDNF on les regles d'esdeveniment d'esborrat del predicat Emp de l'A(D) (D1, D2, D3 i D4) actuen com a conjunt d'entrada. En l'arbre de la figura 4.1.b solament es mostren dues de les quatre branques que es generen en aquest pas, concretament les associades a les regles D1 i D2. Les altres dues branques (D3 i D4) tenen un comportament similar a les branques D1 i D2, respectivament.

En la branca de l'esquerra de l'arbre, en els passos 2a i 3a els literals seleccionats es corresponen a predicats bàsics, i per tant, aquests passos són passos de resolució SLDNF amb el conjunt d'entrada A(D). En el pas 4a, el literal seleccionat és un esdeveniment bàsic totalment instanciat que s'afegeix al conjunt T i d'aquesta manera s'assoleix la clàusula buida. Aquesta branca de l'arbre ha assolit l'èxit, i per tant, permet obtenir la solució corresponent a la transacció T_1={δTreb(Núria, Za)}.

Pel que respecta a la branca dreta de l'arbre de la figura 4.1.b, els passos 2b i 3b també es corresponen a passos de resolució SLDNF amb el conjunt d'entrada A(D). En el pas 4b, el literal seleccionat és un esdeveniment bàsic parcialment instanciat. Abans de incloure'l al conjunt T, cal assignar un valor a la variable c. Per a fer-ho, i per a obtenir totes les possibles solucions a la petició U, cal considerar tots els valors del domini de la variable c. En aquest exemple, considerarem un valor per defecte o que l'usuari proporciona un únic valor, per exemple, c = Al.

En el pas 5b, el literal seleccionat està negat. Caldrà doncs assegurar la seva consistència respecte als esdeveniments que pertanyen en aquest moment al conjunt T. Per a tal efecte, es genera un arbre subsidiari amb l'objectiu $\leftarrow$ μCont(Núria, Za, Al). En el pas 5b.1, es comprova que aquest esdeveniment no pertany a T, de forma que l'arbre falla de forma finita, i per tant, el pas 5b es resol satisfactòriament. Cal tenir en compte que aquesta comprovació de consistència s'ha realitzat tenint en compte únicament els esdeveniments que pertanyen al conjunt T en aquest moment. En el cas que es realitzessin noves incorporacions al conjunt T, hem de seguir assegurant aquesta consistència, en aquest cas, que l'esdeveniment μCont(Núria, Za, Al) no s'inclourà posteriorment al conjunt T. Per a assegurar-ho, utilitzem un conjunt addicional anomenat Conjunt de Condicions C, en el que s'inclouen aquells objectius (condicions) que han fracassat un algun moment del procés de derivació i pels que hem d'assegurar el seu fracàs durant la resta del procés de derivació. Per tant, en el pas 5b.1 hem inclòs la condició $\leftarrow$ μCont(Núria, Za, Al) al conjunt C. Observi's que cada cop que s'inclogui un nou esdeveniment al conjunt T caldrà comprovar que no viola cap de les condicions del conjunt C.

En el pas 6b, s'avalua un expressió de comparació i s'assoleix la clàusula buida. Per tant, obtenim una nova solució a la petició d'actualització U composada per l'esdeveniment del conjunt T_2 = {μTreb(Núria, Za, Al)}.

Aquesta branca de l'arbre de derivació permet obtenir, apart de la solució T_2, tantes solucions com possibles valors existeixen al domini de la variable c diferents a Za ($c \neq Za$).

Les branques de l'arbre associades a les regles d'esdeveniment d'esborrat D9 i D10, que no es mostren a la figura 4.2.b també permeten obtenir solucions a la petició inicial U. Concretament, considerant la regla D3 obtenim la solució T_3 = {δCont(Núria, Za)}, si en canvi considerem la regla D4, la solució obtinguda és T_4 = {μCont(Núria, Za, Al)}.

Així doncs, les diferents formes com es pot esborrar el fet Emp(Núria, Za) són les següents:

T_1 = {δTreb(Núria, Za)}

T_2 = {μTreb(Núria, Za, Al)}

T_3 = {δCont(Núria, Za)}

T_4 = {μCont(Núria, Za, Al)}

Observi's que totes aquestes solucions són mínimes i que no existeix cap altra manera d'induir l'esborrat del fet Emp(Núria, Za) de la base de dades actualitzant únicament la base de dades extensional.

□

4.2.2 Reparació d'una restricció d'integritat

En aquest exemple es mostra com una actualització d'un fet bàsic viola una restricció d'integritat i com aquesta restricció és reparada.

Suposem una petició d'actualització consistent en l'assignació d'un sou de 1500 Euros a la Sílvia per l'empresa HB. En aquest cas, la petició d'actualització es correspon a la conjunció U = ιSou(Sílvia,HB,1500) $\wedge$ $\neg\iota$Ic.

El literal $\neg\iota$Ic s'ha afegit a la petició inicial d'actualització per tal d'evitar que es violi cap restricció d'integritat durant la transició. Com es pot observar, considerant la definició de la restricció Ic2 i degut a que el fet Treb(Sílvia,HB) no és cert a la base de dades, l'actualització U indueix una violació de la restricció i, per a reparar-la, caldrà afegir algun esdeveniment addicional al conjunt T que ho eviti.

A la Figura 4.2 es mostra part de l'arbre de derivació obtingut per aquest exemple.

En el pas 1 d'aquest arbre, el literal seleccionat és un esdeveniment bàsic instanciat i, per tant, s'inclou al conjunt T un cop comprovat que es compleixen els seus prerequisits, és a dir, que no existeix cap fet Sou(Sílvia,c) a la base de dades.

En el pas 2, el literal seleccionat és $\neg\iota$Ic. En aquest pas, hem d'assegurar la consistència d'aquest literal respecte al conjunts T i A(D), és a dir, que no es viola cap restricció d'integritat. Per a assegurar-ho, es genera un arbre subsidiari de consistència en el que totes les branques han de fallar de forma finita. Observi's que en aquest arbre subsidiari, es comprova que no es viola cap restricció d'integritat, i en cas de que se'n violi alguna, s'afegeixen nous esdeveniments al conjunt T per tal de reparar-les. Aquest arbre es representa a la figura 4.2 emmarcat en la capsa principal.

En l'arbre de derivació subsidiari, els passos 2.1, 2.2a i 2.2b es corresponen a passos de resolució SLDNF amb el conjunt d'entrada A(D). En l'arbre de la figura 4.2.2 sols es mostren la branca associada a la segona restricció d'integritat (regla d'esdeveniment C02) i les branques associades a les regles C6 (branca 'a' (esquerra)) i C7 (branca 'b' (dreta)). Per una banda, la primera i tercera restricció d'integritat no es violen, i per l'altra, la resta de branques associades a les regles (C5, C8, C9, C10 i C11) es comporten de forma similar a la branca de la dreta en la figura (branca 'b').

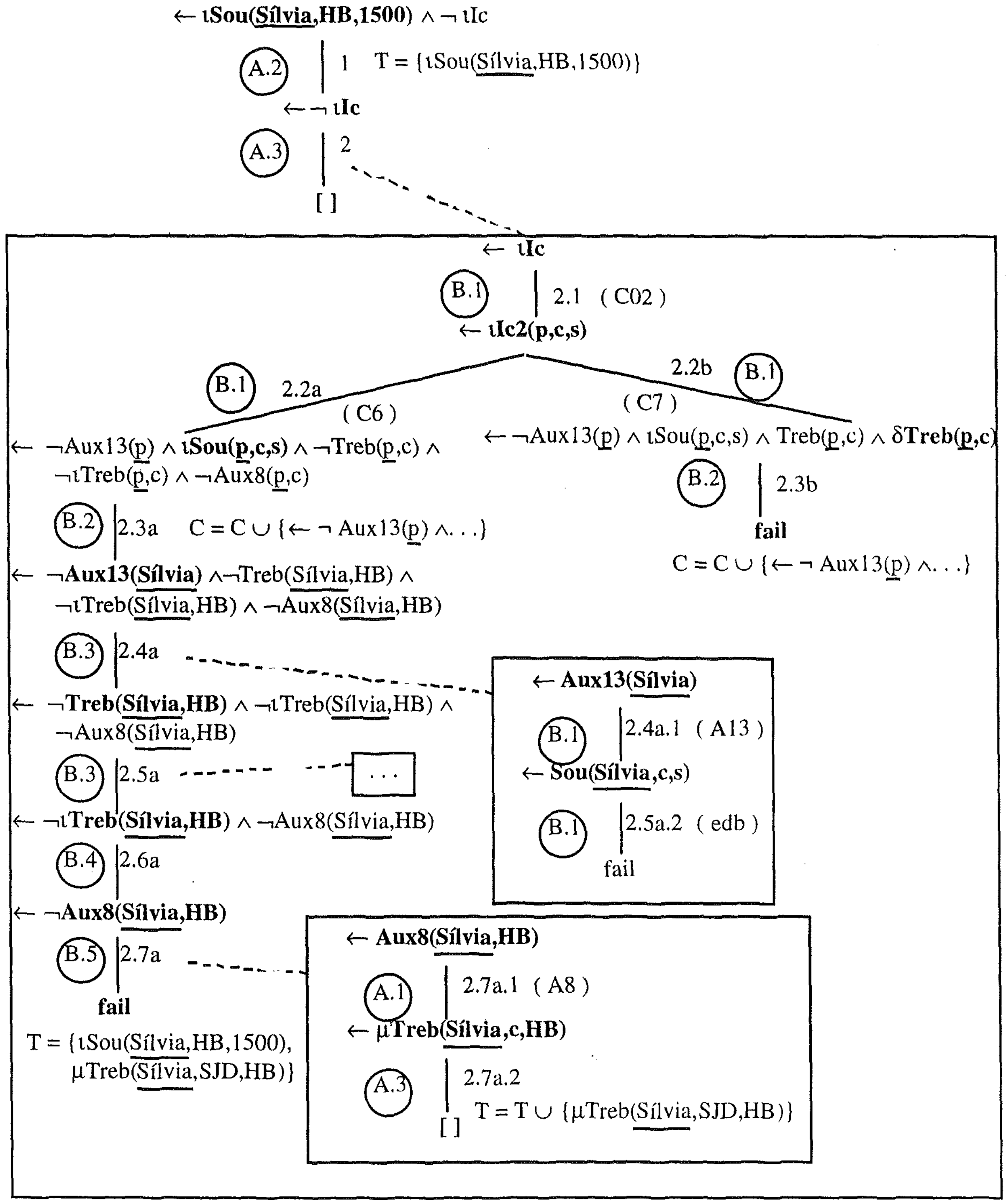

Figura 4.2 Arbre de derivació de l'exemple 4.2.2

En la branca 'b', el pas 2.3b és correspon a un pas de SLDNF però en aquest cas, en conjunt d'entrada és el conjunt T. Al no existir cap esdeveniment en aquest conjunt T que s'unifiqui amb el literal seleccionat, la branca falla. Per evitar que en altres branques de l'arbre s'incorporin esdeveniments que pugin fer tenir èxit a aquesta branca, cal afegir aquest objectiu al conjunt C.

En la branca esquerra ('a') de l'arbre, el pas 2.3a és un pas SLDNF amb el conjunt d'entrada T, on hi ha un esdeveniment que es pot unificar amb el literal ιSou(p,c,s). Al igual que en el cas anterior, afegim l'objectiu al conjunt C per evitar que altres esdeveniments unificables amb el literal s'incloguin a T i facin tenir èxit a aquesta branca.

En el pas 2.4a es requereix demostrar la falsedat del literal Aux13(Sílvia) amb un arbre subsidiari. En aquest arbre es considera el contingut de la base de dades extensional per demostrar que no existeix cap fet Sou(Sílvia,c,s). En pas 2.5a és similar al pas 2.4a.

El literal seleccionat en el pas 2.6a és un esdeveniment bàsic negat. Una possible manera de fer fallar aquesta branca 'a' consisteix en comprovar que l'esdeveniment ιTreb(Sílvia,HB) no pertany al conjunt T i continuar considerant la resta de literals.

En el darrer pas (2.7a), per tal de fallar aquesta branca, cal forçar el fet Aux8(Silvia,HB) amb un arbre subsidiari que assoleixi la clàusula buida. El primer pas d'aquest arbre subsidiari (2.7a.1) es correspon a un pas de resolució SLDNF amb el conjunt d'entrada A(D). En el segon pas (2.7a.2) el literal seleccionat és un esdeveniment bàsic no instanciat. Per assolir la clàusula buida, cal incloure aquest esdeveniment a l conjunt T. Però prèviament, cal comprovar que existeix un fet Treb(Sílvia,c) a la base de dades amb c≠HB i poder així, assignar un valor a la variable c (c=SJD); comprovar que aquest esdeveniment no es mútuament exclusiu amb els del conjunt T; i finalment, comprovar que aquest nou esdeveniment μTreb(Sílvia,SJD,HB) que s'inclou al conjunt T no permet satisfer cap de les condicions del conjunt C que havien fracassat amb anterioritat. Fetes aquestes comprovacions, s'inclou l'esdeveniment al conjunt T, s'assoleix la clàusula buida en aquest arbre subsidiari, i per tant, s'aconsegueix fer fallar la branca 'a' en el pas 2.7a.

La resta de branques de l'arbre associat a l'objectiu ← ιIc també fallen, per tant, en el pas 2 de l'arbre principal s'assoleix la clàusula buida ([]) obtenint una solució a la petició inicial d'actualització U. Aquesta solució està composada pels esdeveniments del conjunt T = {ιSou(Sílvia,HB,1500), μTreb(Sílvia,SJD,HB)}.

Es pot comprovar fàcilment, que en el pas 2.6a podria existir una forma alternativa de fer fallar la branca. Aquesta alternativa està representada a la següent Figura 4.2bis.

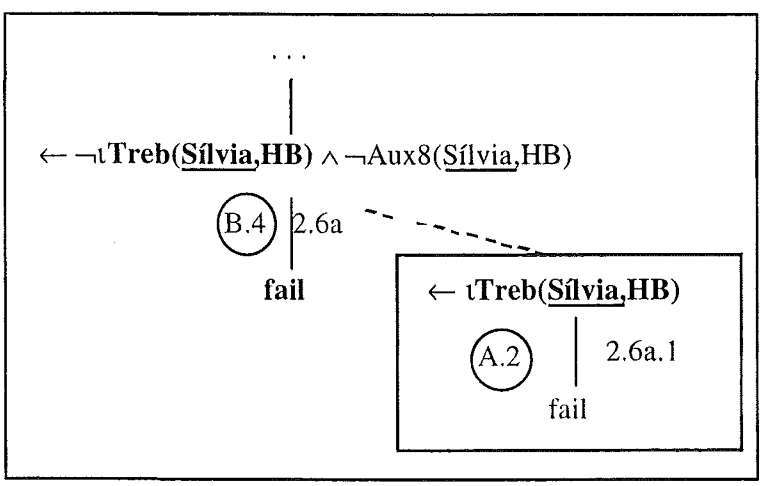

Figura 4.2bis. Alternativa al pas 2.6a de la figura anterior.

Aquesta alternativa al pas 2.6a consisteix en intentar forçar la ocurrència de l'esdeveniment ιTreb(Sílvia,HB). Per a fer-ho cal un arbre subsidiari que tingui èxit, el qual es representa emmarcat en la figura 4.2bis. Observi's que en aquest cas, no és possible obtenir l'èxit en aquest arbre. En el pas 2.6a.1 el literal seleccionat és un esdeveniment bàsic instanciat que caldria incloure al conjunt T. Però en aquest cas, això no és possible, ja que no es satisfan els prerequisits d'aquest l'esdeveniment: $\iota P(\underline{k}, x) \rightarrow \neg \exists\, y\; P(\underline{k}, y)$. Observi's que, en aquest cas, existeix un fet a la base de dades extensional Treb(Sílvia,SJD) que ho impedeix.

En conclusió, en aquest exemple hem obtingut una única solució a la petició d'actualització inicial U = ιSou(Sílvia,HB,1500) ∧ ¬ιIc corresponent al conjunt T={ιSou(Sílvia,HB,1500), μTreb(Sílvia,SJD,HB)}. Observi's que el primer esdeveniment del conjunt T permet satisfer la petició, en canvi, el segon esdeveniment és necessari per a reparar una restricció d'integritat que esdevé violada. Es pot comprovar fàcilment que aquesta solució és mínima i que no existeix cap altra solució que compleixi aquestes mateixes condicions.

4.2.3 Petició d'actualització múltiple

En aquest exemple, la petició d'actualització és múltiple i mixta, és a dir, inclou més d'un esdeveniment i aquests són de diferent tipus. Un d'ells es correspon a una inserció d'un fet derivat i l'altre a una modificació d'un fet bàsic. Concretament la petició d'actualització correspon a la inserció d'en Toni com a una persona activa, i a la modificació del seu contracte: U = ιActiu(Toni) ∧ μCont(Toni,AIC,SS) ∧ ¬ιIc.

Observi's que en aquesta petició, per a satisfer l'esdeveniment ιActiu(Toni) cal que en Toni sigui un empleat d'alguna companyia, en canvi al modificar el seu contracte (μCont(Toni,AIC,SS)) en Toni deixarà de ser empleat. Aquesta situació caldrà tenir-la en compte i generar solucions que l'evitin de forma que es puguin satisfer a la vegada els dos esdeveniments de la petició i, a més a més, no violar cap restricció d'integritat.

En la figura 4.3 es mostra part de l'arbre de resolució associat a la petició U = ιActiu(Toni) ∧ μCont(Toni,AIC,SS) ∧ ¬ιIc.

Els passos 1, 2, 3, 4 i 5 es corresponen a passos de resolució SLDNF amb A(D) com a conjunt d'entrada. En el pas 6 el literal seleccionat és un esdeveniment bàsic instanciat pel que es compleixen els seus prerequisits i al tenir els conjunts T i C buits, s'inclou directament al conjunt T.

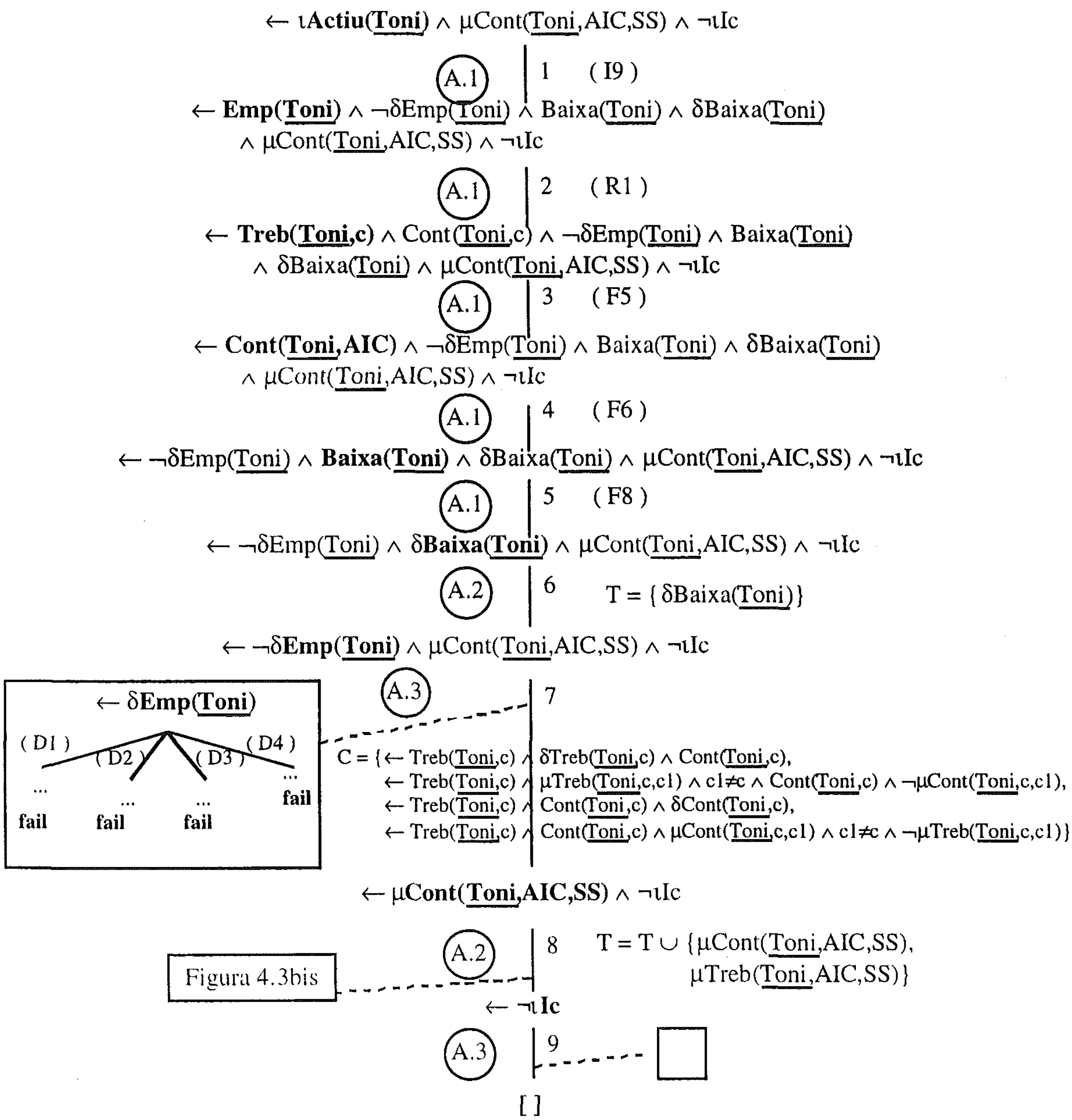

Figura 4.3. Arbre de derivació de l'exemple 4.2.3

En el pas 7, el literal seleccionat és negatiu, per tant, cal demostrar la seva falsedat amb un arbre de derivació subsidiari. Aquest arbre no es mostra en detall en la figura 4.3. Com a conseqüència d'aquesta demostració, al conjunt de condicions C s'hi han afegit quatre condicions que han fracassat i que no es podran satisfer durant tot el procés de derivació. Aquestes condicions es corresponen al cos de les regles d'esdeveniment d'esborrat de δEmp(Toni) identificades per D1, D2, D3 i D4.

En el pas 8, es selecciona un esdeveniment bàsic instanciat que cal incloure al conjunt T. Abans de incloure'l a T s'ha comprovat que existeix el fet Cont(Toni,ACI) a la base de dades extensional, que ACI ≠ SS i que aquest esdeveniment no és mútuament exclusiu amb cap altre del conjunt T. A més a més, amb aquesta nova incorporació al conjunt T, cal assegurar que no hi ha cap condició del conjunt C que esdevingui certa. Aquesta comprovació es fa en l'arbre subsidiari que es mostra a la figura 4.3bis.

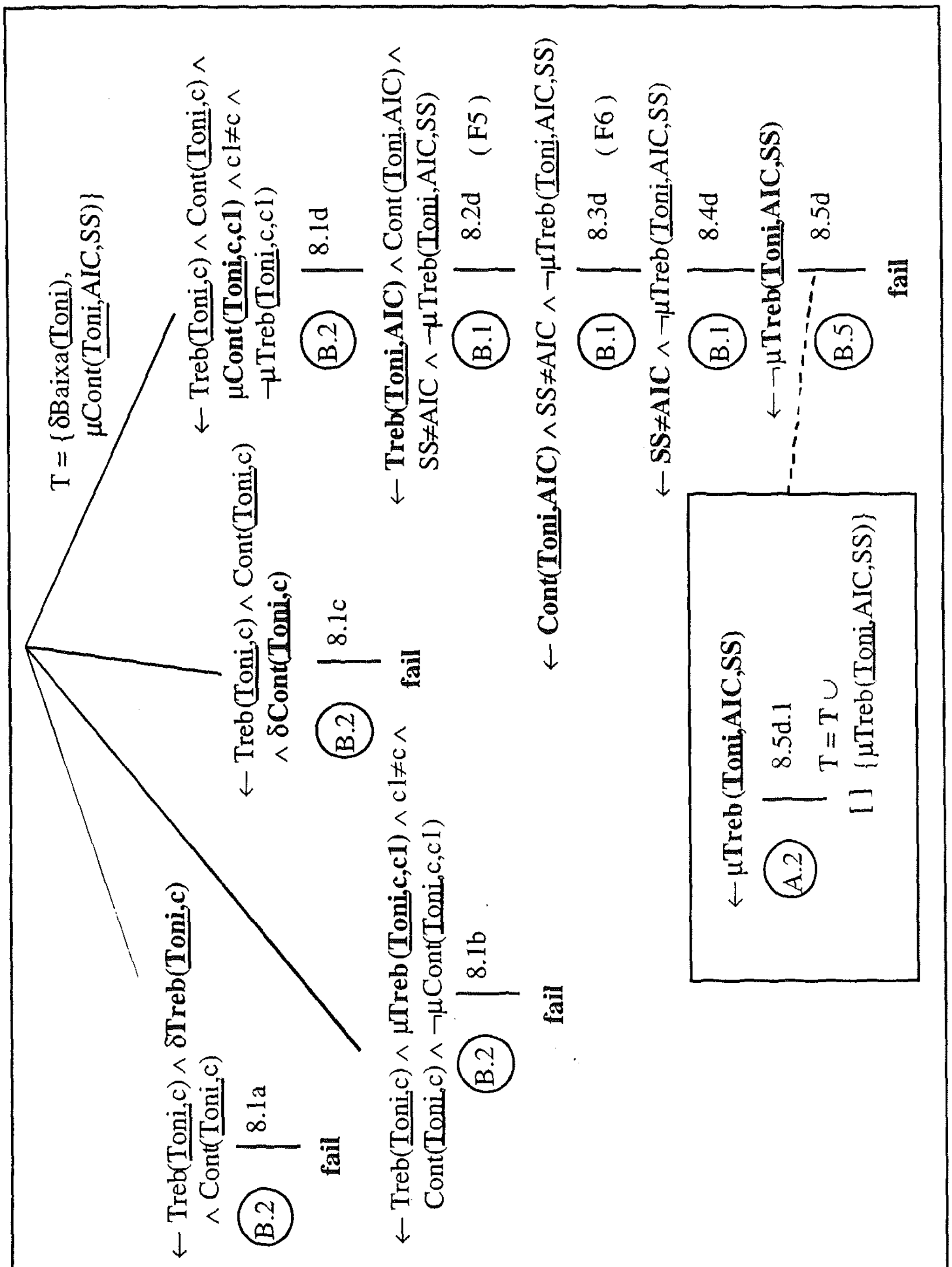

Figura 4.3bis. Arbre de derivació subsidiari del pas 8 (Figura 4.3)

En l'arbre mostrat en la figura 4.3bis, existeix una branca per cada una de les condicions que pertanyen el conjunt C. Hem d'assegurar que totes elles fracassen tenint en compte la incorporació en el pas 8 del nou esdeveniment al conjunt T={δBaixa(Toni), µCont(Toni,AIC,SS)}.

En les tres primeres branques de l'esquerra (passos 8.1a, 8.1b, 8.1c) s'assoleix directament el fracàs, ja que són passos SLDNF on T actua com a conjunt d'entrada i no existeix cap esdeveniment a T unificable amb els literals seleccionats en cada cas. En canvi, en el pas 8.1d s'obté un resolvent que cal fer fallar. Els passos 8.2d i 8.3d es corresponen a passos de resolució SLDNF amb el conjunt d'entrada A(D). En el pas 8.4d s'avalua una expressió de comparació. Finalment, en el pas 8.5d el literal seleccionat és un esdeveniment bàsic instanciat i negat. L'única forma de fer fallar aquesta branca consistirà en forçar aquest esdeveniment mitjançant un arbre subsidiari que assoleixi la clàusula buida. Com es pot veure, en l'únic pas d'aquest arbre (8.5d.1), l'esdeveniment bàsic està instanciat i cal afegir-lo al conjunt T. Es comprova que existeixi el fet Treb(Toni,AIC) a la base de dades; que aquest esdeveniment no sigui mútuament exclusiu amb els del conjunt T; i que no es satisfaci cap condició del conjunt C (aquesta comprovació no es mostra a la figura). Aleshores, s'afegeix l'esdeveniment μTreb(Toni,AIC,SS) al conjunt T per a mantenir totes les condicions del conjunt C no satisfetes, i per tant, poder incloure en el pas 8 de l'arbre principal de la figura 4.3 l'esdeveniment μCont(Toni,AIC,SS) al conjunt T.

Observi's que la inclusió de l'esdeveniment μTreb(Toni,AIC,SS) al conjunt T és necessària per evitar que l'esdeveniment μCont(Toni,AIC,SS) indueixi l'esborrat de Toni com a empleat de la companyia AIC. En realitat, l'ocurrència dels dos esdeveniments permet induir una modificació de la companyia en que està contractat en Toni, permetent així satisfer la petició d'actualització U al complet.

En el pas 9 de l'arbre principal de la figura 4.3, es comprova que no hi ha cap restricció d'integritat que sigui violada per la transacció T. Aquesta comprovació es fa en un arbre subsidiari que no es mostra a la figura. En aquest cas, al no violar-se cap restricció d'integritat, s'assoleix la clàusula buida.

Així doncs, la transacció T={δBaixa(Toni), μCont(Toni,AIC,SS), μTreb(Toni,AIC,SS)} es correspon a l'única solució mínima a la petició inicial d'actualització U.

4.2.4 Reparació d'una restricció que violen altres restriccions ja comprovades

En molts casos, a l'intentar satisfer una petició d'actualització es proposen esdeveniments que violen alguna restricció d'integritat, en aquesta casos, cal proposar nous esdeveniments que reparin aquesta violació. Aquestes reparacions, a la vegada poden induir noves violacions d'altres restriccions d'integritat. Per tant, cada cop que es repara una restricció d'integritat, cal comprovar de nou totes les restriccions d'integritat definides a la base de dades per tal d'assegurar que la base de dades restarà consistent un cop actualitzada.

En aquest exemple, es mostra com al reparar una restricció d'integritat es poden induir noves violacions d'altres restriccions. A més a més, en aquest exemple també es mostra com el conjunt de condicions C és especialment útil per tal d'assegurar que tota reparació d'una restricció d'integritat no viola cap restricció d'integritat que ja ha estat comprovada (i reparada) prèviament.

Sigui la petició d'actualització U = ιNumss(Mercè,800) ∧ ¬ιIc. A la figura 4.4 es mostra part de l'arbre de derivació que s'obté associat a aquesta petició d'actualització.

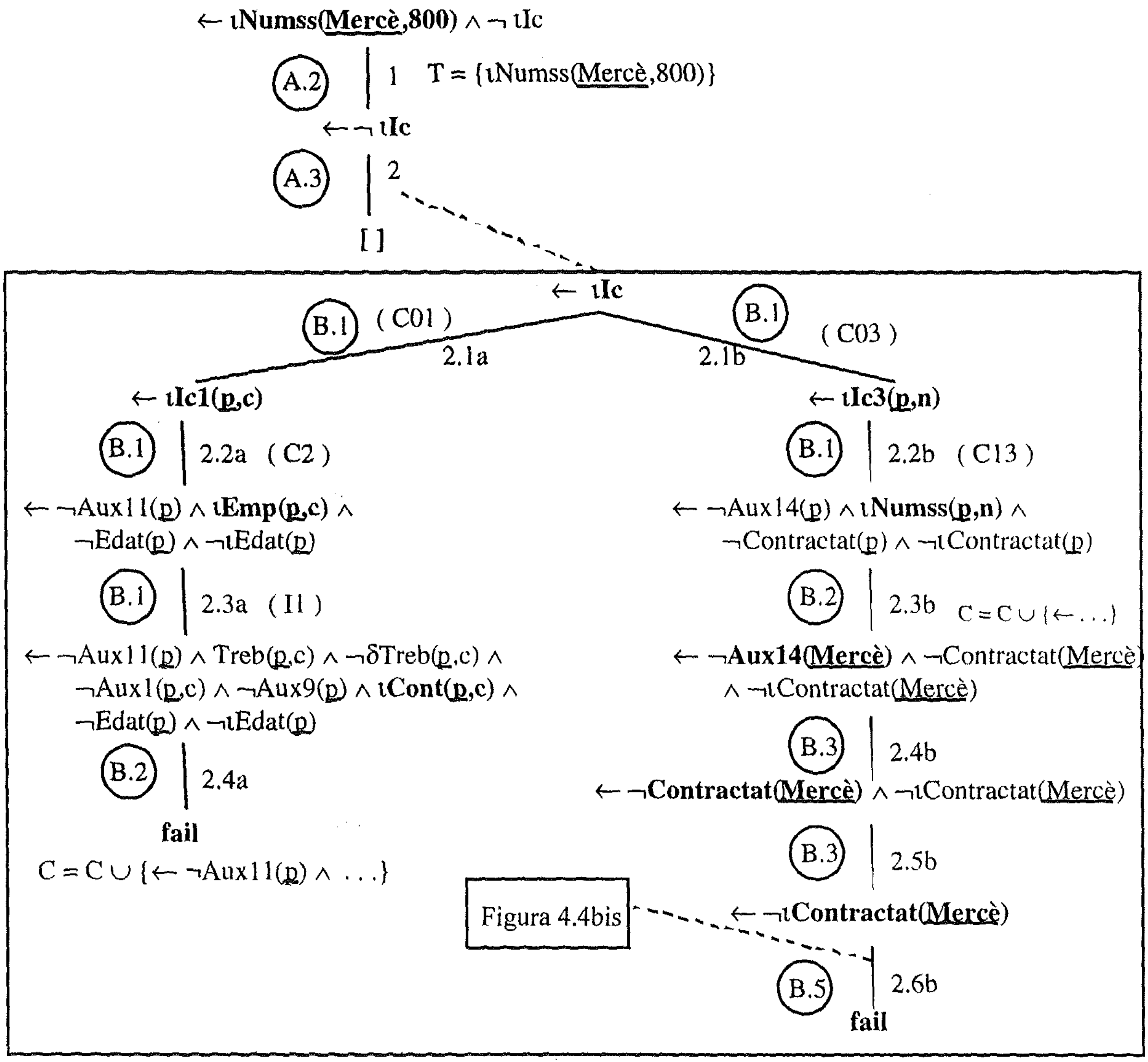

Figura 4.4 Arbre de derivació de l'exemple 4.2.4

A l'arbre principal de la figura 4.4, el literal seleccionat al pas 1 es correspon a un esdeveniment bàsic que cal incloure al conjunt T prèvia comprovació de que no existeix cap fet Numss(Mercè,c) a la base de dades extensional. Aleshores, en el següent pas cal assegurar que aquest esdeveniment no viola cap restricció d'integritat en l'arbre subsidiari emmarcat en aquesta mateixa figura.

Suposem que totes les regles d'esdeveniment de ιIc, ιIc1(p,c), ιIc2(p,c,s) i ιIc3(p,n) es consideren en el mateix ordre en que estan definides a la A(D), és a dir, des de C1 fins a C15. En l'arbre de la figura 4.4, solament es mostren dues de les branques de tot l'arbre de derivació, les branques associades a les regles C2 de la primera restricció d'integritat i la regla C13 de la tercera restricció. Tota la resta de branques fracassen de forma finita, i a més a més, s'inclouen com a condicions al conjunt C. Així doncs, assegurarem que un cop fracassada cada branca associada a una regla Ci, l'objectiu associat s'emmagatzema al conjunt de condicions i no podrà satisfer-se amb posterioritat.

A la branca esquerra de l'arbre, els passos 2.1a, 2.2a i 2.3a es corresponen a passos SLDNF amb el conjunt d'entrada A(D). En el pas 2.4a, també es correspon a un pas de SLDNF però amb el conjunt d'entrada T. A més a més, en aquest pas, s'inclou l'objectiu al conjunt de condicions C i s'assoleix el fracàs de la branca esquerra de l'arbre.

A la branca dreta d'aquest arbre, els passos 2.1b, 2.2b i 2.3b, es corresponen a passos SLDNF amb el conjunt d'entrada A(D). El pas 2.3b també es correspon a un pas de SLDNF amb el conjunt d'entrada T, obtenint un nou objectiu a fer fracassar. A més a més, en aquest pas, s'inclou l'objectiu inicial al conjunt de condicions C. En el passos 2.4b i 2.5b, es demostra la consistència dels literals seleccionats en un arbre de derivació subsidiari que no es mostra a la figura. Finalment, en el pas 2.6, l'única manera de fer fracassar aquesta branca consisteix en forçar l'esdeveniment ιContractat(<u>Mercè</u>) en l'arbre de derivació subsidiari que es mostra a la figura 4.4bis. Observi's que el forçar aquest esdeveniment derivat es correspon a la reparació de la tercera restricció d'integritat, la qual ha esdevingut violada per l'esdeveniment ιNumss(<u>Mercè</u>,800).

Un cop obtingut el fracàs d'aquesta branca, i juntament amb el fracàs de totes les branques d'aquest arbre subsidiari (figura 4.4), hem assegurat que no es viola cap restricció d'integritat i que el contingut del conjunt T es correspon a una solució a la petició d'actualització U.

La figura 4.4b mostra part de l'arbre associat a la generació d'una reparació de la restricció d'integritat Ic3(<u>p</u>,n). La petició de reparació consisteix en satisfer l'objectiu ←ιContractat(<u>Mercè</u>), el qual es correspon a una petició d'actualització de vista que cal traduir a actualitzacions de fets bàsics.

El pas 1 és un pas de resolució SLDNF amb el conjunt d'entrada A(D). En el pas 2, es demostra la no existència de cap fet Cont(<u>Mercè</u>,c) en un arbre subsidiari no mostrat a la figura.

En el pas 3, el literal seleccionat és un esdeveniment bàsic que cal incloure al conjunt T per tal d'assolir la clàusula buida i, per tant, reparar satisfactòriament la restricció Ic3(<u>p</u>,n). Però prèviament, cal realitzar certes comprovacions: 1) cal assignar un valor del domini a la variable c. En el nostre exemple considerarem el valor c=EDM. Els altres valors del domini conduirien a solucions alternatives. 2) Cal comprovar que no existeix cap fet Cont(<u>Mercè</u>,c) a la base de dades extensional, el qual ha estat assegurat en el pas anterior. 3) Cal comprovar que aquest nou esdeveniment no és mútuament exclusiu amb els esdeveniments que ja pertanyen al conjunt T, el qual es satisfà. 4) Finalment cal assegurar que aquest nou esdeveniment no permet satisfer cap de les condicions del conjunt C. Aquesta darrera comprovació es mostra en part a l'arbre subsidiari d'aquesta mateixa figura 4.4bis.

Un cop realitzades les quatre comprovacions del pas 3 de l'arbre de la figura 4.4bis, s'inclou l'esdeveniment al conjunt T i s'assoleix la clàusula buida, per tant, s'ha reparat satisfactòriament la restricció Ic3(p,n).

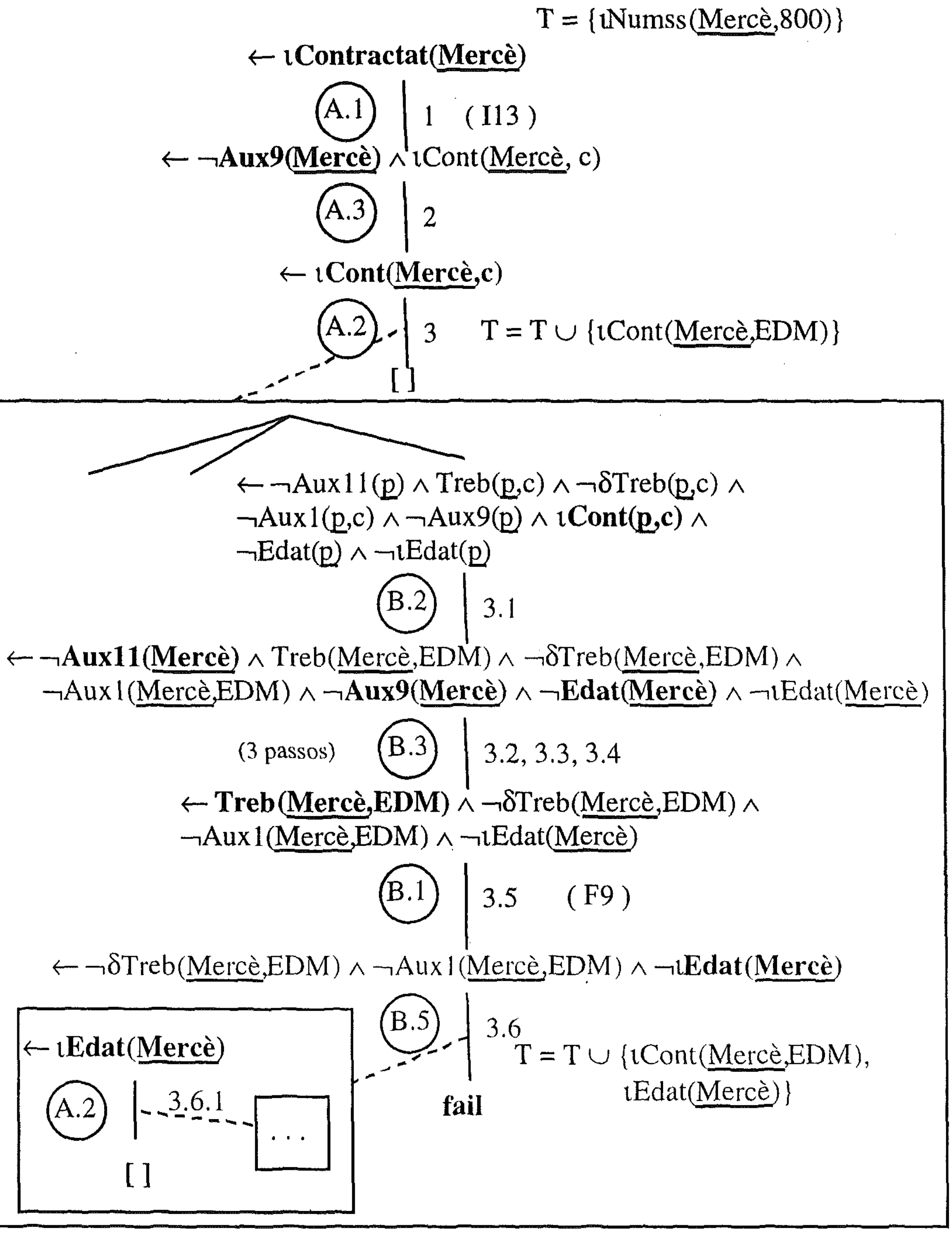

Figura 4.4bis Arbre associat a la reparació de la tercera restricció d'integritat

En l'arbre subsidiari de la figura, existeix una branca per a cada objectiu del conjunt de condicions C. En l'arbre sols es mostra una d'aquestes branques, concretament, la associada a condició generada en la comprovació de la regla d'esdeveniment C2 de la restricció d'integritat Ic1(p,c) de la figura 4.4. Les branques no mostrades a la figura es corresponen a branques que fracassen de forma finita sense requerir cap nova incorporació al conjunt T.

El pas 3.1 es correspon a un pas de resolució SLDNF amb el conjunt d'entrada T. Els passos 3.2, 3.3 i 3.4 són tres passos similars que hem representat en un únic pas en la figura. En aquests passos, es comprova en els respectius arbres subsidiaris la consistència del fet dels literals seleccionats. El pas 3.5 es correspon a un pas de resolució SLDNF amb el conjunt d'entrada A(D). En el darrer pas (3.6) i per a assolir el fracàs d'aquesta branca, cal forçar un d'aquests esdeveniments δTreb(Mercè,EDM) o μTreb(Mercè,EDM,c1) o ιEdat(Mercè) en un arbre subsidiari. Cada una d'aquestes alternatives ens conduirà a una solució diferent. En la figura es mostra la tercera alternativa. En el pas 3.6.1, s'inclou l'esdeveniment a T i es comprova que es compleixin els prerequisits de l'esdeveniment, que no sigui mútuament exclusiu amb els de T i que no hi hagi cap condició del conjunt C que esdevingui satisfeta.

Observi's que en aquest exemple, la reparació de la restricció d'integritat Ic3(p,n) (ιCont(Mercè,EDM)) ha violat la primera restricció d'integritat Ic1(p,c). Però aquesta restricció ja havia estat comprovada prèviament. El fet d'incloure l'objectiu del pas 2.4a de la figura 4.4 en el conjunt de condicions C, i el fet de comprovar aquestes condicions a cada nova incorporació d'un esdeveniment al conjunt T, permet, en aquest exemple, assegurar que la violació de la restricció d'integritat Ic1(p,c) és detectada i reparada satisfactòriament, encara que la restricció violada (Ic1(p,c)) ja havia estat comprovada prèviament a la restricció que ha induït la violació (Ic3(p,n)).

En aquest exemple, i seguint les derivacions mostrades a les figures 4.4 i 4.4bis, hem obtingut una solució T_1 = {ιNumss(Mercè,800), ιCont(Mercè,EDM), ιEdat(Mercè)}. Però per aquest mateix exemple existeixen tres solucions alternatives: T_2 i T_3 s'obtenen considerant un literal diferent al seleccionat al pas 3.6 de la figura 4.4bis, i la solució T_4 s'obté considerant un valor del domini diferent a EDM en el pas 3 de la mateixa figura.

T_2 = {ιNumss(Mercè,800), ιCont(Mercè,EDM), δTreb(Mercè,EDM)}

T_3 = {ιNumss(Mercè,800), ιCont(Mercè,EDM), μTreb(Mercè,EDM,MAC)}

T_4 = {ιNumss(Mercè,800), ιCont(Mercè,MAM)}

Es pot comprovar que totes aquestes solucions són mínimes, que satisfan la petició d'actualització inicial i que no violen cap restricció d'integritat.

4.3 Formalització del mètode

Un cop hem il·lustrat el funcionament del nostre mètode amb diversos exemples, en aquesta secció definirem formalment com, utilitzant el procediment de resolució SLDNF, podem obtenir totes les solucions que satisfan una petició d'actualització sense violar cap restricció d'integritat.

En els exemples anteriors, s'ha pogut comprovar que el mecanisme d'obtenció de les solucions parteix d'un objectiu inicial composat per la conjunció de la petició d'actualització U i per l'esdeveniment negat ¬ιIc.

A partir d'aquest objectiu inicial, el funcionament del mètode consisteix bàsicament en una alternança entre dues activitats: el satisfer la petició d'actualització afegint esdeveniments bàsics al conjunt T i, per altra banda, el comprovar que les actualitzacions de vista induïdes no són contradictòries amb l'actualització demanada i que no es viola cap restricció d'integritat. Aquestes activitats es realitzen, respectivament, durant les derivacions constructiva i de consistència definides a continuació.

Sigui U una petició d'actualització. Una transacció T és una solució a la petició d'actualització U si existeix una *derivació constructiva* des de $(\{\leftarrow U \wedge \neg \iota Ic\}\ \varnothing\ \varnothing)$ fins a $([]\ T\ C)$. Els esdeveniments continguts al conjunt T es corresponen a les actualitzacions (insercions, esborrats i modificacions) que cal aplicar a la base de dades extensional per a satisfer la petició d'actualització U i no violar cap restricció d'integritat.

Durant la derivació constructiva, al seleccionar un literal negat $\neg L_j$ cal garantir-ne la seva consistència respecte als conjunts T i C que es tenen en el moment de fer aquesta selecció. La consistència d'aquest literal es garanteix si es pot demostrar que existeix una *derivació de consistència* des de $(\{\leftarrow L_j\}\ T\ C)$ fins a $(\{\}\ T'\ C')$. L'existència d'aquesta derivació de consistència ens assegura que els esdeveniments del conjunt T' no satisfan el literal L_j. En el cas particular de les restriccions d'integritat ($L_j = \iota Ic$), l'existència de la derivació de consistència des de $(\{\leftarrow \iota Ic\}\ T\ C)$ fins a $(\{\}\ T'\ C')$ ens assegura que els esdeveniments del conjunt T' no violen cap restricció d'integritat.

Abans de definir formalment les derivacions constructiva i de consistència cal tenir en compte les convencions següents:

- El conjunt T conté els esdeveniments bàsics instanciats requerits per a fer assolir l'èxit d'una certa derivació fracassada.
- El conjunt C conté un conjunt de condicions o objectius que han de fracassar en tot el procés de derivació. Aquests objectius han assolit el fracàs en un determinat moment de la derivació i cal assegurar el seu fracàs durant la resta de derivació.
- Denotem G_i a l'objectiu de la forma $\leftarrow L_1 \wedge \ldots \wedge L_k$ i denotem per $G_i \backslash L_j$ a objectiu resultant d'eliminar-ne el literal L_j: $G_i \backslash L_j = \leftarrow L_1 \wedge \ldots \wedge L_{j-1} \wedge L_{j+1} \wedge \ldots \wedge L_k$. Observi's que, si $G_i = \leftarrow L_j$ llavors $G_i \backslash L_j = []$.
- En la derivació de consistència, denotem per F_i al conjunt d'objectius que han d'assolir el fracàs $F_i = H_i \cup F_i'$ on H_i es correspon a l'objectiu $(\leftarrow L_1 \wedge \ldots \wedge L_k)$ que es tracta en aquesta branca de l'arbre de derivació, i F_i' es correspon als objectius pendents de considerar.

Derivació Constructiva

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

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

tal que per cada $i \geq 1$, si G_i té la forma $\leftarrow L_1 \wedge \ldots \wedge L_k$ i on $R(G_i) = L_j$, aleshores (G_{i+1} T_{i+1} C_{i+1}) s'obté segons una de les regles següents:

A1) Si L_j és positiu i no és un esdeveniment bàsic, aleshores $G_{i+1}= S$, $T_{i+1}= T_i$ i $C_{i+1}= C_i$, on S es correspon a: si L_j és un predicat avaluable instanciat que avalua a cert, aleshores S $= G_i \backslash L_j$; si no és un predicat avaluable, aleshores S és el resolvent d'alguna clàusula en A(D) amb G_i en el literal seleccionat L_j.

A2) Si L_j és un esdeveniment bàsic positiu i $\exists\sigma$[2] (una substitució) tal que

A21) $L_j\sigma \in T_i$, aleshores $G_{i+1}= G_i \backslash L_j\sigma$, $T_{i+1} = T_i$ i $C_{i+1} = C_i$.

A22) $L_j\sigma \notin T_i$ i

a) $L_j = \iota P(\underline{k},x)$ i $\neg\, \exists y\ \iota P(\underline{k\sigma},y) \in T_i$ i $\neg\, \exists y$ tal que el fet $P(\underline{k\sigma},y)$ sigui cert a D

b) $L_j = \delta P(\underline{k},x)$ i $\neg\, \exists x'\ \mu P(\underline{k\sigma},x\sigma,x') \in T_i$ i el fet $P(\underline{k},x)\sigma$ és cert a D

c) $L_j = \mu P(\underline{k},x,x')$, $\delta P(\underline{k},x)\sigma \notin T_i$, $\neg\, \exists y'\ \mu P(\underline{k\sigma},x\sigma,y') \in T_i$, $(x \neq x')\sigma$ i el fet $P(\underline{k},x)\sigma$ és cert a D.

Si $C_i = \{\leftarrow Q_1, \ldots, \leftarrow Q_k, \ldots, \leftarrow Q_n\}$ i existeixen derivacions de consistència

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

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

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

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

A3) Si L_j és un literal negat i existeix una derivació de consistència des de ($\{\leftarrow\neg L_j\}$ T_i C_i) fins a ($\{\}$ T' C'), aleshores $G_{i+1}= G_i \backslash L_j$, $T_{i+1}=$ T' i $C_{i+1}=$ C'.

El pas corresponent a la regla A1) és un pas de resolució SLDNF on les clàusules d'A(D) actuen com a clàusules d'entrada.

En el pas corresponent a la regla A2) és té en compte el contingut del conjunt T_i. En el cas de A21), aquest pas és un pas de resolució SLDNF on el conjunt T_i actua com a conjunt d'entrada. En el cas A22) s'afegeix un nou esdeveniment bàsic al conjunt T sempre que es compleixin tres condicions: que aquest esdeveniment no sigui mútuament exclusiu amb els esdeveniment

[2] En el cas que L_j està totalment instanciat, la substitució σ es correspon a la identitat.

existents a T_i, que es satisfacin els prerequisits de la base de dades necessaris per aplicar aquest esdeveniment i, finalment, que es pugui assegurar la consistència de les condicions del conjunt C_i. L'avaluació de la consistència de les condicions del conjunt C_i pot provocar noves incorporacions als conjunts T_i i C_i.

En aquest cas A22), si l'esdeveniment bàsic no està totalment instanciat caldrà assignar valors concrets als seus arguments abans de comprovar la consistència respecte al conjunt C_i i abans de incloure'l al conjunt T_i. Per a fer-ho, cal tenir en compte tots els valors del domini de cada un dels arguments no instanciats.

En el pas corresponent a la regla A3) es comprova la consistència del literal seleccionat respecte T_i i C_i. Aquesta comprovació també pot comportar noves incorporacions a aquests conjunts.

Derivació de Consistència

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

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

tal que per cada $i \geq 1$, si F_i té la forma $\{H_i\} \cup F'_i$ i on $R(H_i) = L_j$ *per algun* $j=1\ldots k$, aleshores (F_{i+1} T_{i+1} C_{i+1}) s'obté segons una de les regles següents:

B1) Si L_j és positiu i no és un esdeveniment bàsic, aleshores $F_{i+1} = S' \cup F'_i$ $T_{i+1} = T_i$ i $C_{i+1} = C_i$, on S' es correspon a: si L_j és un predicat avaluable instanciat que avalua a cert i k>1, aleshores $S' = \{H_i \backslash L_j\}$, però si avalua a fals, aleshores $S' = \emptyset$; si L_j no és un predicat avaluable, aleshores S' és el conjunt de tots els resolvents de les clàusules en A(D) amb H_i en el literal seleccionat L_j si $[] \notin S'$.

B2) Si L_j és un esdeveniment bàsic positiu, S' és el conjunt de tots els resolvents de les clàusules en T_i amb H_i en el literal seleccionat L_j. Si $[] \notin S'$, aleshores $F_{i+1} = S' \cup F'_i$, $T_{i+1} = T_i$.

Si $S' = \emptyset$ o L_j no està totalment instanciat aleshores $C_{i+1} = C_i \cup \{H_i\}$, sinó $C_{i+1} = C_i$.

B3) Si L_j és un literal negatiu, $\neg L_j$ no és un esdeveniment bàsic, aleshores si k>1 i existeix una derivació de consistència des de ($\{\leftarrow \neg L_j\}$ T_i C_i) fins a ({ } T' C'), aleshores $F_{i+1} = \{H_i \backslash L_j\} \cup F'_i$, $T_{i+1} = T'$ i $C_{i+1} = C'$.

B4) Si L_j és un esdeveniment bàsic negatiu i si $\neg L_j \notin T_i$ i $k > 1$, aleshores $F_{i+1} = \{H_i \backslash L_j\} \cup F'_i$, $T_{i+1} = T_i$ i $C_{i+1} = C_i$.

B5) Si L_j és un literal negatiu, llavors si existeix una derivació constructiva des de ($\leftarrow \neg L_j$ T_i C_i) fins a ([] T' C'), aleshores $F_{i+1} = F'_i$, $T_{i+1} = T'$ i $C_{i+1} = C'$.

El pas corresponent a la regla B1) és un pas de resolució SLDNF on les clàusules d'A(D) actuen com a clàusules d'entrada.

En el pas corresponent a la regla B2), si el literal està totalment instanciat, aquest pas es correspon a un pas de resolució SLDNF on el conjunt T_i actua com a conjunt d'entrada. En cas de no estar totalment instanciat caldrà, a més a més, incloure l'objectiu H_i al conjunt C_i per assegurar que noves inclusions a T_i no provoquin l'èxit d'aquest objectiu.

Amb la regla B3) el fracàs de la branca actual s'aconsegueix mitjançant el fracàs de la resta de l'objectiu ($H_i \backslash L_j$) si podem demostrar que l'objectiu $\neg L_j$ fracassa amb una derivació de consistència

La regla B4) és un cas particular de la regla B3) en que el literal seleccionat es correspon a un esdeveniment bàsic negat. En aquest cas la consistència de l'objectiu $\neg L_j$ es demostra comprovant únicament que l'esdeveniment no pertany a T_i.

La regla B5) aconsegueix el fracàs de la branca actual forçant l'èxit del literal $\neg L_j$ amb una derivació constructiva.

Les derivacions de consistència no depenen de l'ordre en que la regla de càlcul selecciona els literals ja que, en general, cal explorar totes les possibilitats per a fer fracassar l'objectiu $H_i = \leftarrow L_1 \wedge \ldots \wedge L_k$. Cal tenir en compte, que cadascuna pot conduir a una solució diferent.

4.4 Demostració de correctesa i completesa

En aquesta secció demostrarem que el mètode definit en aquesta secció és correcte i és complet. En aquest sentit, demostrarem que donada una petició d'actualització u, tota solució obtinguda pel nostre mètode satisfà aquesta petició i no viola cap restricció d'integritat. A més a més, demostrarem que el mètode proposat és capaç d'obtenir totes les possibles formes (solucions) d'actualitzar la base de dades extensional de forma que satisfaci la petició u sense violar cap restricció d'integritat.

4.4.1 Correctesa del mètode

En aquesta secció, presentem la demostració de correctesa del nostre mètode. En aquest sentit, volem demostrar que tota solució a una petició d'actualització u obtinguda pel nostre mètode satisfà aquesta petició i no viola cap restricció d'integritat.

Per a realitzar aquesta demostració, en primer lloc, definirem els conceptes de derivació constructiva i derivació de consistència de nivell k[1]. Abans d'enunciar i demostrar el teorema

---

[1] El concepte de nivell que fem servir en aquesta demostració és diferent al concepte de rang d'una derivació SLDNF definit per Lloyd a [Llo87].

que estableix la correctesa del nostre mètode, introduirem i demostrarem un lema necessari per a demostrar el teorema de correctesa.

Definició 4.4: Sigui G un objectiu qualsevol, T i T' dos conjunts de traducció i, C i C' dos conjunts de condició. Una *derivació de consistència de nivell 0* des de ({G} T C) fins a ({ } T' C') és una derivació de consistència que no crida a cap derivació constructiva ni a cap derivació de consistència.

Definició 4.5: Sigui G un objectiu qualsevol, T i T' dos conjunts de traducció i, C i C' dos conjunts de condició. Una *derivació constructiva de nivell 0* des de (G T C) fins a ([] T' C') és una derivació constructiva que no crida a cap derivació de consistència, o en tot cas, sols crida a derivacions de consistència de nivell 0.

Definició 4.6: Sigui G un objectiu qualsevol, T i T' dos conjunts de traducció i, C i C' dos conjunts de condició. Una *derivació de consistència de nivell k+1* des de ({G} T C) fins a ({ } T' C') és una derivació de consistència que crida a alguna derivació (constructiva o de consistència) de nivell k.

Definició 4.7: Sigui G un objectiu qualsevol, T i T' dos conjunts de traducció i, C i C' dos conjunts de condició. Una *derivació constructiva de nivell k+1* des de (G T C) fins a ([] T' C') és una derivació constructiva que crida a alguna derivació de consistència de nivell k.

Sigui u una petició d'actualització. El Lema 1 estableix que existeix una refutació SLDNF d'A(D) $\cup$ T $\cup$ {$\leftarrow$ u $\wedge \neg$ιIc} per a cada traducció T obtinguda pel nostre mètode.

Lema 1: Sigui D una base de dades deductiva, A(D) la base de dades augmentada associada, u una petició d'actualització i T una traducció mínima tal que existeix una derivació constructiva des de ($\leftarrow$ u $\wedge \neg$ιIc $\emptyset$ $\emptyset$) fins a ([] T C). Llavors, existeix una refutació SLDNF d'A(D) $\cup$ T $\cup$ {$\leftarrow$ u $\wedge \neg$ιIc}.

<u>*Demostració*</u>: Hem de demostrar que les regles que defineixen una derivació constructiva i les de consistència subsidiàries, es corresponen a passos de resolució SLDNF, on les clàusules de A(D) $\cup$ T actuen com a clàusules d'entrada. Aquesta demostració la realitzarem per inducció respecte el nivell k de les derivacions.

Sigui G un objectiu, T i T' dos conjunts de traducció i, C i C' dos conjunts de condició. En primer lloc, demostrarem que una derivació de consistència es correspon a un arbre de cerca SLDNF que fracassa de forma finita. Aquest resultat serà utilitzat amb posterioritat per demostrar que una derivació constructiva es correspon a una refutació SLDNF.

$\boxed{k = 0}$

a) Sigui CS una *derivació de consistència de nivell 0* des de ({G} T C) fins a ({} T' C'). Llavors, l'arbre de derivació SLDNF d'A(D) ∪ T' ∪ {G} fracassa de forma finita[2]. Aquesta derivació es correspon al fracàs de l'objectiu G.

- El pas B1 es correspon a un pas de resolució SLDNF amb el conjunt d'entrada A(D).
- Els passos B2 i B4 es corresponen a passos de resolució SLDNF amb el conjunt d'entrada T=T'.
- Les regles B3 i B5 no són aplicables a aquest cas (k=0).

b) Sigui CT una *derivació constructiva de nivell 0* des de (G T C) fins a ([] T' C'). Llavors, existeix una refutació SLDNF d'A(D) ∪ T' ∪ {G}. Distingim dos casos:

1. No es crida a cap derivació de consistència.

- El pas A1 es correspon a un pas de resolució SLDNF amb el conjunt d'entrada A(D).
- El pas A2 es correspon a un pas de resolució SLDNF amb el conjunt d'entrada T':

  En el cas d'aplicar A21, aquest pas es correspon a un pas de resolució SLDNF amb el conjunt d'entrada T'=T.

  En el cas d'aplicar A22, un esdeveniment bàsic totalment instanciat L_j pot incloure's al conjunt T (T' = T∪{$L_j\sigma$}) i no es requereix cridar a cap derivació de consistència (C=∅). Aleshores, aquest pas també es correspon a un pas de resolució SLDNF amb el conjunt d'entrada T'.
- La regla A3 no es aplicable en aquest cas.

2. Es crida a alguna derivació de consistència de nivell 0.

- El pas A1 correspon a un pas de resolució SLDNF amb el conjunt d'entrada A(D).
- El pas A2 es correspon a un pas de resolució SLDNF amb el conjunt d'entrada T':

  En el cas d'aplicar A21, aquest pas es correspon a un pas de resolució SLDNF amb el conjunt d'entrada T'=T.

  En el cas d'aplicar A22, un esdeveniment bàsic totalment instanciat Lj pot incloure's al conjunt T, i per tant, cal cridar a una derivació de consistència de nivell 0 des de (C T∪{Ljσ} C) fins a ({} T' C'). En cas d'existir aquesta derivació, obtindrem un nou objectiu en la derivació constructiva. Aquesta derivació de consistència es correspon a

[2] Si G es correspon a un conjunt de n>1 objectius, l'arrel de l'arbre serà un objectiu implícit ←F amb n branques descendents, una per a cada objectiu H_i de G.

l'aplicació de la regla de negació per fracàs finit de l'objectiu C respecte a la nova clàusula d'entrada $Lj\sigma$. El fracàs de C respecte al conjunt T no es veu alterat i, per tant, l'aplicació de la regla A22 es correspon a un pas de resolució SLDNF amb el conjunt d'entrada T'.

- El pas en el que s'aplica la regla A3 es comprova l'existència de la derivació de consistència de nivell 0 des de ($\leftarrow\neg L_j$ T C) fins a ({ } T' C'). És a dir, es comprova la falsedat del literal $\neg L_j$. Aquesta derivació es correspon a l'aplicació de la regla de negació per fracàs finit. Així doncs, la regla A3 es correspon a un pas de resolució SLDNF.

$\boxed{k = k}$

Donat que hem provat el Lema 1 pel cas bàsic (k=0), ara suposarem que aquest enunciat també és cert per a derivacions de nivell k.

$\boxed{k = k+1}$

Ara provarem el Lema 1 per a derivacions de nivell k+1.

a) Sigui CS una *derivació de consistència de nivell k+1* des de ({G} T C) fins a ({ } T' C'). Llavors, l'arbre de cerca SLDNF d'A(D) $\cup$ T' $\cup$ {G} fracassa de forma finita[2].

- El pas B1 es correspon a un pas a SLDNF amb el conjunt d'entrada A(D).
- El pas B2 es correspon a un pas de resolució SLDNF amb el conjunt d'entrada T'. El fracàs de l'objectiu G està assegurat per:
  - els esdeveniments que ja pertanyen al conjunt T
  - aquells esdeveniments bàsics que s'inclouran al conjunt T després d'aquest pas (T'-T). Aquests esdeveniments s'inclouran en una derivació constructiva de nivell k (regla A2). Com que l'objectiu G és inclòs en el conjunt de condició C, llavors, aquest objectiu no es satisfarà, assegurat per la derivació de consistència subsidiària a l'aplicació de la regla A2.
- El pas B3 es correspon a un pas de resolució SLDNF amb el conjunt d'entrada A(D) $\cup$ T'. Per inducció, l'arbre de cerca SLDNF d'A(D) $\cup$ T' $\cup$ {$\leftarrow\neg L_j$} fracassa de forma finita ja que aquest pas crida a una derivació de consistència de nivell k.
- El pas B4 es correspon a un pas de resolució SLDNF amb el conjunt d'entrada T=T'.

---

[2] Si G es correspon a un conjunt de n>1 objectius, l'arrel de l'arbre serà un objectiu implícit $\leftarrow$F amb n branques descendents, una per a cada objectiu H_i de G.

- El pas B5 es correspon a un pas de resolució SLDNF amb el conjunt d'entrada A(D) $\cup$ T'. Per inducció, existeix una refutació d'A(D) $\cup$ T' $\cup$ $\{\leftarrow\neg L_j\}$ ja que es crida a una derivació constructiva de nivell k.

b) Sigui CT una *derivació constructiva de nivell k+1* des de (G T C) fins a ([] T' C'). Llavors, existeix una refutació SLDNF d'A(D) $\cup$ T' $\cup$ {G}.

- El pas A1 es correspon a un pas de resolució SLDNF amb el conjunt d'entrada A(D).
- El pas A2 es correspon a un pas de resolució SLDNF amb el conjunt d'entrada T':

  En el cas d'aplicar A21, aquest pas es correspon a un pas de resolució SLDNF amb el conjunt d'entrada T=T'.

  En el cas d'aplicar A22, un esdeveniment bàsic totalment instanciat L_j pot incloure's al conjunt T i, per tant, cal cridar a una derivació de consistència de nivell k+1 des de (C T$\cup\{L_j\sigma\}$ C) fins a ({ } T' C'). En cas d'existir aquesta derivació, obtindrem un nou objectiu en la derivació constructiva. Com ja hem provat en el cas c), una derivació de consistència de nivell k+1 o inferior, es correspon a aplicar la regla de negació per fracàs finit de l'objectiu C respecte a la nova clàusula d'entrada $L_j\sigma$. El fracàs de C respecte al conjunt T no es veu alterat i, per tant, l'aplicació de la regla A22 es correspon a un pas de resolució SLDNF amb el conjunt d'entrada T'.
- El pas en el que s'aplica la regla A3 es comprova l'existència de la derivació de consistència de nivell 0 des de ($\leftarrow\neg L_j$ T C) fins a ({ } T' C'). És a dir, es comprova la falsedat del literal $\neg L_j$. Aquesta derivació es correspon a l'aplicació de la regla de negació per fracàs finit. Així doncs, la regla A3 es correspon a un pas de resolució SLDNF.

$\square$

Teorema 1: (Correctesa del mètode)

Sigui D una base de dades deductiva, A(D) la base de dades augmentada associada, i u una petició d'actualització, tal que u no és conseqüència lògica del complement comp(A(D)). Sigui T una traducció obtinguda pel nostre mètode. Llavors, u $\wedge\neg$ıIc és conseqüència lògica de comp(A(D) $\cup$ T).

<u>*Demostració*</u>:

El lema 1 estableix que existeix una refutació SLDNF d'A(D) $\cup$ T $\cup$ {$\leftarrow$ u $\wedge\neg$ıIc} si existeix una derivació constructiva des de ($\leftarrow$ u $\wedge\neg$ıIc $\varnothing$ $\varnothing$) fins a ([] T C).

A més a més, tenint en compte que la resolució SLDNF és correcte, l'existència de la refutació SLDNF d'A(D) $\cup$ T $\cup$ {$\leftarrow$ u $\wedge\neg$ıIc } ens assegura que u $\wedge\neg$ıIc és conseqüència lògica de comp(A(D) $\cup$ T).

Com que tota traducció T obtinguda pel nostre mètode, s'obté mitjançant una derivació constructiva des de ($\leftarrow$ u $\wedge$ $\neg$ıIc $\varnothing$ $\varnothing$) fins a ([] T C) llavors, segons el Lema 1 i la correctesa de la resolució SLDNF, podem assegurar que u $\wedge$ $\neg$ıIc és conseqüència lògica de comp(A(D) $\cup$ T). És a dir, tota traducció T obtinguda pel nostre mètode és correcte ja que satisfà la petició inicial d'actualització i no viola cap restricció d'integritat.

□

4.4.2 Completesa del mètode

En aquesta secció, presentem la demostració de completesa del nostre mètode. En aquest sentit, volem demostrar que el nostre mètode és capaç d'obtenir totes les possibles solucions a una petició d'actualització u. És a dir, totes les possibles formes d'actualitzar la base de dades extensional de forma que en el nou estat de la base de dades, se satisfaci la petició d'actualització i no es violi cap restricció d'integritat.

Sigui D una base de dades deductiva, A(D) la base de dades augmentada associada, u una petició d'actualització i T una traducció mínima tal que la base de dades un cop actualitzada satisfà u. És a dir, que u se segueix d'A(D) $\cup$ T i, per tant, ha d'existir una refutació SLDNF d'A(D) $\cup$ T $\cup$ {$\leftarrow$ u}.

Així doncs, per cada possible T per la que usant resolució SLDNF tinguem una refutació d'A(D) $\cup$ T $\cup$ {$\leftarrow$ u}, demostrarem (teorema 3) que existeix una derivació constructiva des de ($\leftarrow$u $\varnothing$ $\varnothing$) fins a ([] T C).

Aquesta demostració està basada en la completesa de la resolució SLDNF. Com ja hem esmentat en el capítol 3 d'aquesta tesi, la base de dades D és estratificada i la seva A(D) associada és consistent per crida. Així doncs, la resolució SLDNF es completa en aquest tipus de base de dades i, per tant, per a qualsevol d'aquestes traduccions T existeix una refutació SLDNF.

Per a realitzar aquesta demostració, utilitzarem el concepte de rang d'una refutació SLDNF i de rang d'un arbre SLDNF que fracassa de forma finita tal com es defineix a [Llo87]. Establirem i demostrarem tres lemes i un teorema previs a l'enunciat i demostració del teorema que estableix la completesa del mètode proposat.

Lema 2: Sigui D una base de dades deductiva; A(D) la base de dades augmentada associada; G i H dos objectius; T, T' i T" tres conjunts de traducció; C, C' i C" tres conjunts de condició. Aleshores, aquests dos resultats són certs:

a) Si existeix una refutació SLDNF de rang n d'A(D) $\cup$ T $\cup$ {G}, aleshores:

- per tot T' tal que T' $\subseteq$ T i
- per cada conjunt C' tal que, per tot $C_i \in$ C' l'arbre SLDNF d'A(D) $\cup$ T $\cup$ {C_i} fracassa de forma finita i té rang n-1

existeix una derivació constructiva des de (G T' C') fins a ([] T'' C'') on:

- T'' $\subseteq$ T i
- per cada condició $C_i \in$ C'', l'arbre SLDNF d'A(D) $\cup$ T'' $\cup$ $\{C_i\}$ fracassa de forma finita i té rang n-1.

b) Si existeix un arbre SLDNF de rang n d'A(D) $\cup$ T $\cup$ {H} que fracassa de forma finita, aleshores:

- per tot T' tal que T' $\subseteq$ T i
- per cada conjunt C' tal que, per tot $C_i \in$ C' l'arbre SLDNF d'A(D) $\cup$ T $\cup$ $\{C_i\}$ fracassa de forma finita i té rang n-1

existeix una derivació de consistència des de ({H} T' C') fins a ({ } T'' C'') on:

- T'' $\subseteq$ T i
- per cada condició $C_i \in$ C''- C', l'arbre SLDNF d'A(D) $\cup$ T $\cup$ $\{C_i\}$ fracassa de forma finita i té rang n-1.

<u>*Demostració*</u>: aquesta demostració la realitzarem per inducció sobre el rang n de les refutacions i/o arbres de cerca SLDNF.

A cada pas de resolució SLDNF d'una refutació li associarem un pas de la derivació constructiva i, provarem que en cada pas intermedi (G_i T_i C_i) es compleixen les dues condicions del lema. Inicialment tenim que G_i=G, T_i=T' i C_i=C'. En el darrer pas obtindrem la clàusula buida ([] T'' C'').

A cada pas de resolució d'un arbre de cerca SLDNF fracassat de forma finita li associarem un pas de la derivació de consistència i, provarem que en cada pas intermedi (F_i T_i C_i) es compleixen les dues condicions del lema. Cal tenir en compte que F_i es correspon als objectius dels nodes de l'arbre de cerca $F_i = \{H_i\} \cup F'_i$. Inicialment tenim F_i = {H}, T_i=T' i C_i=C'. En el darrer pas de cada branca de l'arbre s'assoleix la clàusula { }.

$\boxed{n = 0}$

a) En aquest cas, segons la definició de refutació SLDNF de rang 0, l'objectiu G sols pot contenir literals positius i no es requereix cap arbre de cerca auxiliar fracassat de forma finita. Aleshores, en aquest cas, hem de demostrar que existeix una derivació constructiva des de (G T' $\emptyset$) fins a ([] T'' $\emptyset$) on T'' $\subseteq$ T.

Sigui L_j el literal seleccionat en un pas de la refutació. Tenint en compte el tipus d'aquest literal, anem a veure quin resultat obtenim en la derivació SLDNF:

- <u>L_j és un literal positiu</u>

- *L_j no és un esdeveniment bàsic*: Obtenim (S T_i $\emptyset$). On hem aplicat un pas de resolució SLDNF amb el conjunt d'entrada A(D). (A1)
- *L_j és un esdeveniment bàsic*: Obtenim ($G_i \backslash L_j\sigma$ $T_i \cup \{L_j\sigma\}$ $\emptyset$). L'esdeveniment $L_j\sigma$ ja pertany al conjunt de traducció T degut a que hem assumit que existia una refutació d'A(D) $\cup$ T $\cup$ {G}. Aquest pas es correspon a un pas de resolució SLDNF on el conjunt T actua com a conjunt d'entrada i on $T_i \cup \{L_j\sigma\} \subseteq T$. (A2)

▪ <u>L_j és un literal negatiu</u>

Aquest cas no és aplicable per n=0. (A3)

Així doncs, tots els casos estan contemplats i la derivació acaba amb la clàusula buida [].

b) Com en el cas anterior, segons la definició d'arbre de cerca SLDNF de rang 0 fracassa de forma finita, l'objectiu F_i sols pot contenir literals positius i no es requereix cap arbre de cerca fracassat de forma finita ni cap refutació SLDNF auxiliar. Aleshores, en aquest cas, hem de demostrar que existeix una derivació de consistència des de (F_i T' C') fins a ({ } T'' C'') on els arbres de cerca SLDNF associats a cada $C_i \in$ C'' fracassen de forma finita.

Sigui H_i l'objectiu $\leftarrow L_1 \wedge \ldots \wedge L_k$, i el literal seleccionat en cada pas L_j. Tenint en compte el tipus d'aquest literal, anem a veure quin resultat obtenim en la derivació SLDNF:

▪ <u>L_j és un literal positiu</u>

- *L_j no és un esdeveniment bàsic*: Obtenim ($S' \cup F_i'$ T_i C_i) on $[] \notin S'$. Aquest pas es correspon a un pas de resolució SLDNF amb el conjunt d'entrada A(D). (B1)
- *L_j és un esdeveniment bàsic*: (B2)

  I. Si $L_j \in T_i$ i està totalment instanciat, obtenim ($S' \cup F_i'$ T_i C_i).

     En aquest cas, $S' = \{H_i \backslash L_j\}$ on $[] \notin S'$ i es correspon a un pas de resolució SLDNF amb el conjunt d'entrada T_i.

  II. Si $L_j \notin T_i$ i està totalment instanciat, obtenim (F_i' T_i $C_i \cup \{H_i\}$).

     En aquest cas, $S' = \emptyset$. L'arbre de cerca SLDNF d'A(D) $\cup$ T $\cup$ $\{H_i\}$ té rang 0 i fracassa de forma finita: per als esdeveniments que pertanyen a T_i fracassa de forma finita, ja que es tracta de la branca actual; per als esdeveniments que s'inclouran posteriorment a T, també fracassa, ja que ho assegurem al incloure H_i al conjunt de condició i quan s'incloguin aquests esdeveniments a T, el pas A2 comprovarà el fracàs de H_i.

  III. Si $L_j \in T_i$; no està instanciat, però es pot unificar amb una substitució σ i $[] \notin S'$, aleshores obtenim ($S' \cup F_i'$ T_i $C_i \cup \{H_i\}$).

En aquest cas, $S'=\{H_i \backslash L_j\sigma\}$ on $[] \notin S'$. Llavors, com en el cas II, l'arbre de cerca SLDNF d'$A(D) \cup T \cup \{H_i\}$ té rang 0 i fracassa de forma finita.

IV. Si $L_j \notin T_i$ i no està instanciat, obtenim $(F_i' \; T_i \; C_i \cup \{H_i\})$.

En aquest cas, $S'=\varnothing$. Llavors, com en el cas II, l'arbre de cerca SLDNF d'$A(D) \cup T \cup \{H_i\}$ té rang 0 i fracassa de forma finita.

- L_j és un literal negatiu

  Aquest cas no és aplicable per n=0. (B3, B4, B5)

Així doncs, tots els casos estan contemplats i la derivació acaba amb fracàs ({ }).

$\boxed{n = n\text{-}1}$

Suposem que els dos resultats del lema són certs per a refutacions SLNDF i arbres de cerca de rang n-1 fracassats de forma finita. És a dir, existeix una derivació constructiva des de (G T' C') fins a ([] T'' C'') associada a la refutació SLDNF de rang n-1 d'$A(D) \cup T \cup \{G\}$. De la mateixa manera, existeix una derivació de consistència des de ({H} T' C') fins a ({ } T'' C'') associada a l'arbre de cerca de rang n-1 d'$A(D) \cup T \cup \{H\}$ que fracassa de forma finita.

$\boxed{n = n}$

Ara provarem que aquesta mateixos resultats són certs per a refutacions i arbres de cerca de rang n fracassats de forma finita.

a) Segons la definició de refutació SLDNF de rang n, l'objectiu G tan pot contenir literals positius com negatius. Aleshores, en aquesta cas, hem de demostrar que existeix una derivació constructiva des de (G T' C') fins a ([] T'' C'') on $T'' \subseteq T$ i on l'arbre de cerca SLDNF per cada $C \in C''$ fracassa de forma finita.

Sigui L_j el literal seleccionat en un pas de la refutació. Tenint en compte el tipus d'aquest literal, anem a veure quin resultat obtenim en la derivació SLDNF:

- L_j és un literal positiu

  – *L_j no és un esdeveniment bàsic*: Obtenim $(S \; T_i \; C_i)$ on hem aplicat un pas de resolució SLDNF amb A(D) com a conjunt d'entrada. (A1)

  – *L_j és un esdeveniment bàsic*: (A2)

    I. Si $L_j\sigma \in T_i$, obtenim $(G_i \backslash L_j\sigma \; T_i \; C_i)$.

    Aquest cas correspon a un pas de resolució SLDNF amb el conjunt d'entrada T_i.

    II. Si $L_j\sigma \notin T_i$, obtenim $(G_i \backslash L_j\sigma \; T' \; C')$

En aquest cas, es comprova l'existència d'una derivació de consistència (C $T_i \cup \{L_j\sigma\}$ C_i) per cada $C \in C_i$.

Com que hem suposat que l'arbre de cerca SLDNF d'A(D) $\cup$ T $\cup$ {C} per cada $C \in C_i$ fracassa de forma finita i té rang n-1, aleshores, per inducció, existeix una derivació de consistència des de (C $T_i \cup \{L_j\sigma\}$ C_i) fins a ({} T' C') amb T' $\subseteq$ T i els arbres de cerca SLDNF A(D) $\cup$ T $\cup$ {C} per tota $C \in C' - C_i$ tenen rang n-1 i fracassen de forma finita.

- <u>L_j és un literal negatiu</u> (A3)

  El següent objectiu serà ($G_i \backslash L_j$ T' C') si existeix la derivació de consistència des de ({$\leftarrow \neg L_j$} T_i C_i) fins a ({} T' C'). Com que l'arbre de cerca SLDNF d'A(D) $\cup$ T $\cup$ {$\leftarrow \neg L_j$} té rang n-1 i fracassa de forma finita, aleshores, per inducció, existeix aquesta derivació de consistència.

Així doncs, tots els casos estan contemplats i la derivació acaba amb la clàusula buida [].

b) Suposem que existeix un arbre de cerca SLDNF de rang n d'A(D) $\cup$ T $\cup$ {H} que fracassa de forma finita. Hem de demostrar que existeix una derivació de consistència des de ({H} T' C') fins a ({} T" C") on T" $\subseteq$ T i on els arbres de cerca SLDNF per cada $C \in C$" fracassen de forma finita.

Sigui H_i l'objectiu $\leftarrow L_1 \wedge \ldots \wedge L_k$, i L_j el literal seleccionat en cada pas de l'arbre. Tenint en compte el tipus d'aquest literal, anem a veure quin resultat obtenim en la derivació SLDNF:

- <u>L_j és un literal positiu</u>

  – *L_j no és un esdeveniment bàsic*: Obtenim ($S' \cup F_i'$ T_i C_i) on $[] \notin S'$. Aquest pas es correspon a un pas de resolució SLDNF amb el conjunt d'entrada A(D). (B1)

  – *L_j és un esdeveniment bàsic*: (B2)

  I. Si $L_j \in T_i$ i està totalment instanciat, obtenim ($S' \cup F_i'$ T_i C_i).

  En aquest cas, $S' = \{H_i \backslash L_j\}$ on $[] \notin S'$ i es correspon a un pas de resolució SLDNF amb el conjunt d'entrada T_i.

  II. Si $L_j \notin T_i$ i està totalment instanciat, obtenim (F_i' T_i $C_i \cup \{H_i\}$).

  En aquest cas, $S' = \varnothing$. L'arbre de cerca SLDNF d'A(D) $\cup$ T $\cup$ {H_i} té rang n i fracassa de forma finita: per als esdeveniments que pertanyen a T_i fracassa de forma finita, ja que es tracta de la branca actual; per als esdeveniments que s'inclouran posteriorment a T, també fracassa, ja que ho assegurem al incloure H_i al conjunt de condició i quan s'incloguin aquests esdeveniments a T, el pas A2 comprovarà el fracàs de H_i.

III. Si $L_j \in T_i$; no està instanciat, però es pot unificar amb una substitució σ i $[] \notin S'$, aleshores obtenim $(S' \cup F_i'\ T_i\ C_i \cup \{H_i\})$.

En aquest cas, $S'=\{H_i \backslash L_j\sigma\}$ on $[] \notin S'$. Llavors, com en el cas II, l'arbre de cerca SLDNF d'$A(D) \cup T \cup \{H_i\}$ té rang n i fracassa de forma finita.

IV. Si $L_j \notin T_i$ i no està instanciat, obtenim $(F_i'\ T_i\ C_i \cup \{H_i\})$.

En aquest cas, $S'=\varnothing$. Llavors, com en el cas II, l'arbre de cerca SLDNF d'$A(D) \cup T \cup \{H_i\}$ té rang n i fracassa de forma finita.

- L_j és un literal negatiu

  – *L_j no és un esdeveniment bàsic*:

I. En cas d'existir una derivació de consistència des de $(\{\leftarrow\neg L_j\}\ T_i\ C_i)$ fins a $(\{\}\ T'\ C')$ i k>1 llavors, obtenim $(\{H_i \backslash L_j\} \cup F_i'\ T'\ C')$. (B3)

La definició d'arbre de cerca SLDNF de rang n fracassat de forma finita estableix que existeix un l'arbre subsidiari d'$A(D) \cup T \cup \{\leftarrow\neg L_j\}$ té rang n-1 i fracassa de forma finita. Aleshores, per inducció, podem demostrar que existeix la derivació de consistència anterior.

II. En cas d'existir una derivació constructiva des de $(\leftarrow\neg L_j\ T_i\ C_i)$ fins a $([]\ T'\ C')$ llavors, obtenim $(F_i'\ T'\ C')$. (B5)

La definició d'arbre de cerca SLDNF de rang n fracassat de forma finita estableix que existeix la refutació d'$A(D) \cup T \cup \{\leftarrow\neg L_j\}$ i que té rang n-1. Aleshores, per inducció, podem demostrar que existeix la derivació de consistència anterior.

  – *L_j és un esdeveniment bàsic*:

I. Si $\neg L_j \notin T$ i k>1 llavors, obtenim $(\{H_i \backslash L_j\} \cup F_i'\ T_i\ C_i)$. (B4)

Aquest pas es correspon a un pas de resolució SLDNF amb el conjunt d'entrada T_i.

II. Si $\neg L_j \in T$ o ($\neg L_j \notin T$ amb k=1) llavors, obtenim $(F_i'\ T_i\ C_i)$. (B5)

L'existència de la derivació constructiva des de $(\leftarrow\neg L_j\ T_i\ C_i)$ fins a $([]\ T_i\ C_i)$ està assegurada, ja que aquest pas es correspon a un pas de resolució SLDNF amb el conjunt d'entrada T_i.

Així doncs, tots els casos estan contemplats i la derivació acaba amb fracàs ({ }).

□

Lema 3: Sigui D una base de dades deductiva; A(D) la base de dades augmentada associada; u una petició d'actualització i T una traducció mínima. Totes les refutacions SLDNF (principal i auxiliars) que apareixen en l'espai de cerca SLDNF d'$A(D) \cup T \cup \{\leftarrow U\}$ són assolides per la derivació constructiva de $(\{\leftarrow u\}\ \emptyset\ \emptyset)$ fins a ([] T C).

<u>*Demostració*</u>: Farem aquesta demostració per reducció a l'absurd. Suposem que hi ha una refutació SLDNF d'$A(D) \cup T \cup \{\leftarrow G_s\}$ que no és assolida per la derivació constructiva des de $(\{\leftarrow u\}\ \emptyset\ \emptyset)$ fins a ([] T C).

- Aquesta refutació no pot ser la principal ja que, per la correctesa del mètode, a la derivació constructiva des de $(\{\leftarrow u\}\ \emptyset\ \emptyset)$ fins a ([] T C) li correspon la refutació SLDNF d'$A(D) \cup T \cup \{\leftarrow u\}$.
- Aleshores, tan sols pot ser una refutació SLDNF auxiliar. Aquesta refutació serà cridada des d'algun pas de resolució d'un arbre de cerca SLDNF d'$A(D) \cup T \cup \{H_k\}$ que fracassa de forma finita. Pel lema 2, ha d'existir una derivació de consistència des de $(\{H_k\}\ T_k\ C_k)$ fins a ({ } T' C') a la que li correspongui aquest arbre de cerca SLDNF.

Si aquesta derivació de consistència no assoleix l'objectiu G_s és degut a que en algun pas de la derivació, l'aplicació d'una regla del mètode (B1, .., B5) al literal seleccionat L_j ha fet fracassar la branca actual. Les regles que fan fracassar una branca de la derivació de consistència són les següents:

(B1) La branca actual fracassa si $S'=\emptyset$. Però aquest pas és un pas de resolució SLDNF que no requereix cap derivació auxiliar.

(B2) La branca actual fracassa si l'esdeveniment bàsic $L_j \notin T_k$. Aquest pas es correspon a un pas de resolució SLDNF amb el conjunt d'entrada T_k que fracassa. Si suposem que l'esdeveniment L_j s'inclourà posteriorment al conjunt de traducció $T\text{-}T_k$ en alguna derivació constructiva llavors, caldrà assegurar el fracàs de l'objectiu $\leftarrow H_k$ amb una derivació de consistència. Si aquesta derivació de consistència també fracassa llavors, els conjunts de traducció T i $T\text{-}\{L_j\}$ correspondrien a dues traduccions de la petició u. Aquest fet contradiu el supòsit de que T és solució mínima i, per tant, aquesta derivació constructiva per afegir L_j a $T\text{-}T_k$ no pot ser cridada.

(B3) L'aplicació d'aquesta regla no fa fracassar la branca actual.

(B4) L'aplicació d'aquesta regla no fa fracassar la branca actual.

(B5) Si existeix una derivació constructiva des de $(\leftarrow\neg L_j\ T_k\ C_k)$ fins a ([] T' C'), per la correctesa del mètode, existeix una refutació SLDNF d'$A(D) \cup T \cup \{\leftarrow\neg L_j\}$, per tant, no pot ser la d'$A(D) \cup T \cup \{\leftarrow G_s\}$.

Analitzats tots els casos, es conclou que no pot existir una refutació SLDNF d'A(D) $\cup$ T $\cup$ {$\leftarrow G_s$} que no sigui assolida per una derivació constructiva (principal o auxiliar) des de ({$\leftarrow$u} $\emptyset$ $\emptyset$) to ([] T C). Aleshores, totes les refutacions SLDNF que apareixen a l'espai de cerca SLDNF d'A(D) $\cup$ T $\cup$ {$\leftarrow$ u} són assolides per la derivació constructiva des de ({$\leftarrow$u} $\emptyset$ $\emptyset$) fins a ([] T C), quedant demostrat aquest lema 3.

□

Lema 4: Sigui D una base de dades deductiva; A(D) la base de dades augmentada associada; u una petició d'actualització i T una traducció mínima. Si existeix una refutació SLDNF d'A(D) $\cup$ T $\cup$ {$\leftarrow$ u} llavors, cada esdeveniment bàsic t $\in$ T és utilitzat en la refutació principal o en una refutació auxiliar d'A(D) $\cup$ T $\cup$ {$\leftarrow$ u}.

Demostració: Suposem que un esdeveniment t és utilitzat únicament en passos de resolució en arbres de cerca SLDNF que fracassen de forma finita. En aquest cas, els arbres de cerca també fracassen amb el conjunt de transacció T-{t}, i per tant, aquest conjunt és també una traducció. Aquest fet és contradictori amb que T és una traducció mínima i, per tant, tots els esdeveniment del conjunt T han de ser utilitzats en alguna refutació (principal o auxiliar) d'A(D) $\cup$ T $\cup$ {$\leftarrow$ u}.

□

Teorema 2: Sigui D una base de dades deductiva; A(D) la base de dades augmentada associada; u una petició d'actualització i T una traducció mínima. Si existeix una refutació SLDNF d'A(D) $\cup$ T $\cup$ {$\leftarrow$ u} llavors, existeix una derivació constructiva des de ({$\leftarrow$u } $\emptyset$ $\emptyset$) fins a ([] T C).

Demostració: El lema 2 ens assegura que si existeix una refutació d'A(D) $\cup$ T $\cup$ {$\leftarrow$ u} llavors, existeix una derivació constructiva des de ({$\leftarrow$u} $\emptyset$ $\emptyset$) fins a ([] T' C') amb T' $\subseteq$ T. Pel lema 3, aquesta derivació constructiva assoleix la refutació principal i totes les auxiliars de l'espai de cerca d'A(D) $\cup$ T $\cup$ {$\leftarrow$ u}. Pel lema 4, tots els esdeveniments de T' són utilitzats en aquestes refutacions. Com que T és una traducció mínima i T' $\subseteq$ T llavors T' ha d'incloure tots els esdeveniments de T, és a dir, que T'=T. Aleshores, queda demostrat que existeix la derivació constructiva des de ({$\leftarrow$u} $\emptyset$ $\emptyset$) fins a ([] T C).

□

Teorema 3: (Completesa del mètode)

Sigui D una base de dades deductiva; A(D) la base de dades augmentada associada; u una petició d'actualització i T una traducció mínima. Suposem que la resolució SLDNF és completa per a l'A(D) $\cup$ T i l'objectiu {$\leftarrow$ u}. Llavors, per a qualsevol conjunt de traducció T tal que u és conseqüència lògica de comp(A(D) $\cup$ T), existeix una derivació constructiva des de ({$\leftarrow$u} $\emptyset$ $\emptyset$) fins a ([] T C).

Demostració: A partir de la completesa de la resolució SLDNF, si u és conseqüència lògica de comp(A(D) $\cup$ T) llavors, existeix una refutació SLDNF d'A(D) $\cup$ T $\cup$ {$\leftarrow$ u}. Pel teorema 2, podem assegurar per tant que existeix una derivació constructiva des de ({$\leftarrow$u} $\emptyset$ $\emptyset$) fins a ([] T C). Així doncs, per a qualsevol T tal que u és conseqüència lògica de comp(A(D) $\cup$ T), existeix una derivació constructiva des de ({$\leftarrow$u} $\emptyset$ $\emptyset$) fins a ([] T C), i per tant, és obtinguda pel nostre mètode.

□

Dels teoremes 1 i 3 es poden deduir dues conclusions importants: Sigui u $\wedge \neg$ıIc la petició inicial d'actualització. La correctesa del mètode (teorema 1) ens assegura que si existeix una derivació constructiva des de ($\leftarrow$ u $\wedge \neg$ıIc $\emptyset$ $\emptyset$) fins a ([] T C), aleshores la base de dades modificada segons T satisfà la petició inicial u i no viola cap restricció d'integritat. Per altra banda, la completesa del mètode (teorema 3) ens assegura que si la derivació constructiva anterior fracassa de forma finita, aleshores no és possible satisfer la petició inicial d'actualització canviant únicament la base de dades extensional.