

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

7. Comparació amb altres treballs a nivell d'eficiència

Entre tots els treballs analitzats al capítol 5 d'aquest document, solament els treballs de [CFPT94, FP97] i [GL93, Ger94] consideren, directament o indirectament, el problema de l'eficiència en el procés de manteniment de restriccions d'integritat. Però, cap dels treballs analitzats presenta cap tècnica específica per a la millora de la eficiència en el procés de traducció de vistes.

Per a comparar el nostre mètode respecte a altres treballs a nivell d'eficiència, a la secció 7.1 comparem el nostre treball amb els treballs presentats a [CFPT94, FP97] i a [GL93, Ger94] pel manteniment de restriccions d'integritat. A la secció 7.2 s'analitza la millora de l'eficiència que s'aconsegueix amb la definició de l'arquitectura del mètode del capítol 6. Aquesta anàlisi es realitza comparant l'eficiència del nostre mètode respecte a l'eficiència del Mètode dels Esdeveniments [TO95]. Una alternativa equivalent a aquesta segona comparació consisteix en fer l'anàlisi de millora de l'eficiència respecte a la formalització del nostre propi mètode del capítol 4, però hem decidit comparar amb el Mètode dels Esdeveniments per tal de fer explícita l'aportació en eficiència del nostre mètode respecte al nostre precursor.

7.1 Treballs de Manteniment de Restriccions d'Integritat

En aquesta secció presentem les principals conclusions obtingudes després de fer una comparació detallada del nostre mètode amb els mètodes [CFPT94, FP97] i [GL93, Ger94], tenint en compte únicament aspectes d'eficiència en el procés de manteniment de restriccions d'integritat.

Els dos mètodes analitzats segueixen un enfocament actiu, és a dir, es basen en la generació i execució de regles actives pel manteniment de les restriccions d'integritat. Aquestes regles actives es defineixen en temps de compilació, i en temps d'execució s'executen per a comprovar i reparar les possibles violacions d'una restricció d'integritat. El propòsit comú d'aquests treballs és el de definir l'ordre en que cal mantenir les restriccions d'integritat. Aquest ordre ve determinat per l'ordre en que cal executar les regles actives que s'encarreguen de comprovar i reparar cada restricció d'integritat.

Per a definir aquest ordre, aquests mètodes analitzen les dependències que hi ha entre les regles actives que reparen les restriccions d'integritat i que, a la vegada, poden produir noves violacions d'aquestes restriccions. Aquestes dependències entre les regles actives es representen en un graf que servirà per a definir l'ordre d'execució d'aquestes regles i, per tant, l'ordre de tractament de les restriccions d'integritat.

En els dos treballs [CFPT94, Ger94] un graf està definit per un conjunt de nodes i un conjunt d'arestes entre aquests nodes. Cada node es correspon a una restricció d'integritat,

mentre que una aresta entre dos nodes es correspon a una regla activa. Així doncs, existeix una aresta R_k entre un node Ic_i i un node Ic_j per a indicar que la regla activa R_k al reparar la restricció Ic_i pot violar la restricció Ic_j .

Una primera diferència entre aquests mètodes i el presentat en aquest document rau en el moment en què es determina l'ordre de tractament de les restriccions d'integritat. En els mètodes [CFPT94, Ger94] l'ordre d'execució de les regles actives es determina en temps de compilació utilitzant la informació representada en el graf. En temps d'execució, aquest graf no es té en compte ja que l'ordre d'execució de les regles ja està totalment determinat. En canvi, en el nostre mètode, el Graf de Precedències¹ representa les precedències entre condicions i és en temps d'execució, durant el procés de manteniment de la consistència quan es determina l'ordre real de tractament d'aquestes condicions.

Però la principal diferència entre aquests mètodes i el presentat en aquesta tesi rau en la precisió de la informació representada en el graf de precedències. En els dos treballs [CFPT94, Ger94], cada node del graf es correspon a una de les restriccions d'integritat definides a la base de dades i, cada una de les arestes representades es correspon a una regla activa. En el nostre mètode, al considerar les condicions de les restriccions d'integritat en lloc de les pròpies restriccions i, al considerar les actualitzacions concretes (reparadors) en comptes de les regles que proposen aquestes actualitzacions, fa que el nostre Graf de Precedències sigui molt més precís en la identificació i definició de precedències entre condicions, i per tant, entre les restriccions d'integritat.

Precisió en la definició de les precedències

El Graf de Precedències del nostre mètode defineix de forma molt més precisa les precedències entre restriccions. En el Graf de Precedències, una aresta entre els nodes C_i i C_j està etiquetada amb un esdeveniment E per a indicar que l'actualització associada a E pot reparar la condició C_i i, a la vegada, pot violar la condició C_j .

Cada aresta del Graf de Precedències indica quina és l'actualització o esdeveniment causant de la precedència. En canvi, en els grafs generats per [CFPT94] i [Ger94], les arestes indiquen quina és la regla activa responsable de la precedència. Aleshores, en els casos en que una regla activa realitzi més d'una actualització, no es coneix directament quina d'elles és la causant real de la precedència. Per altra banda, en els nodes del nostre Graf de Precedència es representen Condicions de Restriccions d'Integritat, en lloc de la pròpia restricció d'integritat. Així doncs, podem diferenciar les diferents formes com cada restricció d'integritat pot ésser violada i, per tant, podem definir més precisament les precedències entre condicions.

¹ Ens referirem al Graf de Precedències en comptes del Graf de Precedències Actiu, ja que considerarem que aquest Graf sols conté Condicions de Restriccions d'Integritat.

El fet de representar les Condicions de Restricció d'Integritat en lloc de la pròpia definició de la restricció provoca que el nombre de nodes que componen el Graf de Precedències sigui molt més gran al nombre de nodes dels grafs definits per [CFPT94, Ger94]. En canvi, aquest major nombre de nodes permet que en el nostre Graf de Precedències hi apareguin menys cicles que en els grafs proposats per [CFPT94, Ger94].

Exemple 7.1: Siguin Ic1 i Ic2 dues restriccions d'integritat definides per predicats bàsics en forma de denegació. Per cada un dels mètodes [CFPT94, Ger94] es mostren les regles actives generades i els grafs associats.

$$Ic1(x, y, z) \leftarrow S(x, y) \wedge T(y, z) \wedge \neg R(x, z)$$

$$Ic2(x, y, z) \leftarrow S(x, y) \wedge R(z, y)$$

[CFPT94]

$$R1: S(x, y) \wedge T(y, z) \wedge \neg R(x, z) \rightarrow \text{delete}(S(x, y))$$

$$R2: S(x, y) \wedge T(y, z) \wedge \neg R(x, z) \rightarrow \text{delete}(T(y, z))$$

$$R3: S(x, y) \wedge T(y, z) \wedge \neg R(x, z) \rightarrow \text{insert}(R(x, z))$$

$$R4: S(x, y) \wedge R(z, y) \rightarrow \text{delete}(S(x, y))$$

$$R5: S(x, y) \wedge R(z, y) \rightarrow \text{delete}(R(x, z))$$

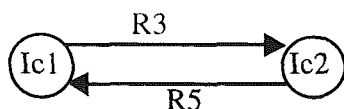


Fig.7.1 Graf de precedències de l'exemple 7.1 [CFPT94]

[Ger94]

$$R11: \Delta^{\text{insert}T}_{Ic1}(y, z) \wedge (S(x, y) \wedge \neg R(x, z)) \triangleright \neg S(x, y)$$

$$R12: \Delta^{\text{delete}R}_{Ic1}(x, z) \wedge (S(x, y) \wedge T(y, z)) \triangleright \neg S(x, y)$$

$$R21: \Delta^{\text{insert}S}_{Ic1}(x, y) \wedge (T(y, z) \wedge \neg R(x, z)) \triangleright \neg T(y, z)$$

$$R22: \Delta^{\text{delete}R}_{Ic1}(x, z) \wedge (S(x, y) \wedge T(y, z)) \triangleright \neg T(y, z)$$

$$R31: \Delta^{\text{insert}S}_{Ic1}(x, y) \wedge (T(y, z) \wedge \neg R(x, z)) \triangleright R(x, z)$$

$$R32: \Delta^{\text{insert}T}_{Ic1}(y, z) \wedge (S(x, y) \wedge \neg R(x, z)) \triangleright R(x, z)$$

$$R41: \Delta^{\text{insert}R}_{Ic2}(z, y) \wedge (S(x, y)) \triangleright \neg S(x, y)$$

$$R51: \Delta^{\text{insert}S}_{Ic2}(x, y) \wedge (R(z, y)) \triangleright \neg R(y, z)$$

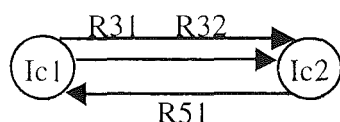


Fig.7.2 Graf de precedències de l'exemple 7.1 [Ger94]

En el nostre mètode, les condicions associades a les restriccions d'integritat i el Graf de Precedències és el següent:

- C1 $\leftarrow S(x, y) \wedge \neg \delta S(x, y) \wedge T(y, z) \wedge \neg \delta T(y, z) \wedge R(x, z) \wedge \delta R(x, z)$
- C2 $\leftarrow S(x, y) \wedge \neg \delta S(x, y) \wedge \neg T(y, z) \wedge \iota T(y, z) \wedge \neg R(x, z) \wedge \neg \iota R(x, z)$
- C3 $\leftarrow S(x, y) \wedge \neg \delta S(x, y) \wedge \neg T(y, z) \wedge \iota T(y, z) \wedge R(x, z) \wedge \delta R(x, z)$
- C4 $\leftarrow \neg S(x, y) \wedge \iota S(x, y) \wedge T(y, z) \wedge \neg \delta T(y, z) \wedge \neg R(x, z) \wedge \neg \iota R(x, z)$
- C5 $\leftarrow \neg S(x, y) \wedge \iota S(x, y) \wedge T(y, z) \wedge \neg \delta T(y, z) \wedge R(x, z) \wedge \delta R(x, z)$
- C6 $\leftarrow \neg S(x, y) \wedge \iota S(x, y) \wedge \neg T(y, z) \wedge \iota T(y, z) \wedge \neg R(x, z) \wedge \neg \iota R(x, z)$
- C7 $\leftarrow \neg S(x, y) \wedge \iota S(x, y) \wedge \neg T(y, z) \wedge \iota T(y, z) \wedge R(x, z) \wedge \delta R(x, z)$
- C8 $\leftarrow S(x, y) \wedge \neg \delta S(x, y) \wedge \neg R(z, y) \wedge \iota R(z, y)$
- C9 $\leftarrow \neg S(x, y) \wedge \iota S(x, y) \wedge R(z, y) \wedge \neg \delta R(z, y)$
- C10 $\leftarrow \neg S(x, y) \wedge \iota S(x, y) \wedge \neg R(z, y) \wedge \iota R(z, y)$

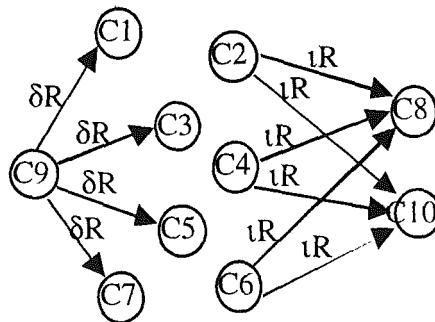


Fig. 7.3 Graf de Precedències entre condicions

Com es pot comprovar en el Graf de Precedències obtingut en el nostre cas, hi ha més nodes que en els casos anteriors però, en canvi, no hi ha cicles entre les condicions. Una de les avantatges de la major precisió en la definició del Graf de Precedències és que el nombre de cicles entre condicions és molt menor en el nostre Graf de Precedències que en els grafes de [CFPT94, Ger94].

A les seccions 7.1.1 i 7.1.2 s'analitzen algunes diferències específiques entre el mètode proposat en aquesta tesi i els mètodes [CFPT94, FP97] i [GL93, Ger94] respectivament.

7.1.1 Mètode de S.Ceri, P.Fraternali, S.Paraboschi i L.Tanca [CFPT94, FP97]

Aquest treball, des d'un punt de vista d'eficiència, té com a objectiu assegurar que l'execució del conjunt de regles actives definides pel manteniment de restriccions d'integritat finalitzi. Per assolir aquest objectiu, defineixen en temps de compilació un ordre total entre aquestes regles. Una de les conseqüències de la definició d'aquest ordre entre regles és que obtenen indirectament un ordre de comprovació i reparació de les restriccions d'integritat. Una restricció d'integritat únicament es comprova quan ja s'han obtingut tots els seus violadors

potencials, i per tant, redueixen el nombre de vegades que cal comprovar cada restricció d'integritat.

Per a definir aquest ordre total entre regles actives, defineixen un graf anomenat *Triggering Hippergraph THG*. Els nodes es corresponen a restriccions i els arcs a regles actives. En aquest graf THG poden aparèixer cicles entre nodes. Les arestes que componen aquests cicles es corresponen a regles actives, l'execució de les quals pot no acabar. Per tant, per a assegurar la finalització de l'execució de les regles actives, cal assegurar que el THG sigui acíclic. Els autors proposen una heurística per a obtenir un subgraf del THG dirigit i acíclic que anomenen *Acyclic Directed Hippergraph ADHG*. Per a obtenir-lo, es proposa eliminar algunes de les arestes per tal de trencar els cicles existents. La selecció d'aquestes arestes es fa tenint en compte les actualitzacions que es realitzaran al executar-se la regla activa associada i un pes (o prioritat) assignat pel dissenyador a cada regla. El nou graf acíclic obtingut determina l'ordre d'execució de les regles actives assegurant la seva terminació.

De totes formes, cal dir que en el procés de conversió del THG al ADHG, no tenen perquè eliminar-se tots els cicles. Aquests cicles poden no ser eliminats per dues raons principalment: per assegurar que no quedi cap restricció sense una regla activa que la reperi, perquè a l'eliminar un cicle podem eliminar la única regla activa que repara aquesta restricció; o per voluntat expressa del dissenyador, el qual pot modificar a mà el graf obtingut. En aquests casos, en temps d'execució caldrà un procés de monitorització de l'execució de les regles actives que componen el cicle per tal d'evitar una execució infinita.

A [FP97] es descriu i formalitza amb més detall el mecanisme per a determinar l'ordre d'execució de les regles actives del DAHG. Es defineix una partició de les regles actives en diferents estrats o nivells: les regles anomenades *Upward Triggering rules (UT)* componen el nivell inferior de la partició i, la resta de regles actives anomenades *Downward Triggering rules (DT)* es reparteix en diferents nivells o estrats per sobre del nivell inferior. Les regles UT, al executar-se, només poden produir l'activació potencial de regles DT. Les regles DT poden activar regles actives de nivells o estrats inferiors (DT o UT). Així doncs, la no terminació de l'execució de les regles actives només pot ser causada per les regles UT, ja que són les úniques que poden introduir cicles en el DAHG.

Així doncs, si en temps d'execució no s'activa cap regla UT està assegurat que el procés de manteniment de restriccions d'integritat finalitza. A més a més, cap restricció d'integritat que ja ha estat reparada és violarà de nou per la reparació d'una altre restricció. En canvi, si s'executa alguna regla UT la no terminació no està assegurada.

El principal problema que té aquest mètode és que per tal d'assegurar la finalització del procés de manteniment de restriccions d'integritat pateixen una pèrdua de completesa en aquest procés. Al convertir el THG en un DAHG eliminant arestes del graf, deixen de considerar formes alternatives de reparar restriccions d'integritat. D'aquesta manera, aquest mètode pot no

generar totes les solucions existents pel manteniment de restriccions d'integritat com ja s'ha vist en el capítol 5. Per tant, les solucions obtingudes per aquest mètode depenen directament del conjunt final de regles actives considerat.

En el nostre Graf de Precedències, es consideren totes les formes de reparar cada restricció d'integritat. A més a més, els cicles que apareixen en el Graf de Precedències no sempre es corresponen a cicles reals en temps d'execució. Aleshores, no tenim que fer cap tipus de monitorització per tal d'assegurar la terminació del procés de manteniment de les restriccions d'integritat.

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

Aquest mètode [Ger94] està orientat al manteniment eficient de restriccions d'integritat. A diferència del mètode anterior [CFPT94], el propòsit d'aquest mètode és definir un ordre de tractament de les restriccions d'integritat per tal de reduir el nombre de vegades que una restricció s'ha de comprovar durant el procés de manteniment d'aquestes restriccions.

Per tal de definir aquest ordre entre restriccions, el dissenyador ha d'especificar per a cada restricció d'integritat el conjunt de regles actives ECA que la poden reparar. Un cop definides aquestes regles, el propi dissenyador els hi ha d'assignar un pes o prioritat. Aquest pes permet ordenar-les de forma que en violar-se una restricció, una única regla activa es escollida per a ser executada per a reparar-la. Aquesta serà la que tingui el pes més alt de totes les regles aplicables a la violació concreta.

Un cop especificades les regles actives associades a totes les restriccions d'integritat, es construeix el *Dependency Graph DG*. Els nodes d'aquest graf es corresponen a les restriccions d'integritat i les arestes a les regles actives que les reparen. Aquest graf és utilitzat per a definir automàticament, i en temps de compilació, un ordre total de tractament de les restriccions d'integritat. Aquest ordre es defineix assignant prioritats a les restriccions d'integritat.

L'assignació de la prioritat a cada restricció d'integritat es fa de la forma següent: es donen les prioritats més elevades a les restriccions associades als nodes que no tenen cap predecessor en el graf, ja que no poden ser violades per cap reparació d'altres restriccions. Dintre d'aquest grup, les restriccions que no poden ser reparades en cap cas tindran la màxima prioritat. La mínima prioritat es dona a les restriccions associades a nodes que no tenen successors en el graf, ja que la seva reparació no pot violar cap altra restricció. A la resta de restriccions, els hi assigna una prioritat en funció de les prioritats de les restriccions dels nodes predecessors. Per a definir aquestes prioritats, cal que el graf sigui acíclic. En cas de tenir algun cicle, caldrà desfer-lo escollint un dels nodes del cicle com a node inicial del cicle i eliminant totes les arestes que arriben a aquest node i que s'originen en un node del cicle. Un cop s'ha obtingut un graf acíclic, ja es poden assignar les prioritats a aquests nodes. El requeriment de tenir un graf acíclic és únicament un requisit per a fer l'assignació de prioritats als nodes.

En temps d'execució les restriccions d'integritat es comproven seguint l'ordre definit per les prioritats que tenen assignades. Cada una d'elles, en cas de ser violada, es repararà utilitzant la regla activa de més pes que pugui aplicar-se. Observi's que l'ordre entre restriccions i entre regles actives, no ha eliminat els cicles entre regles actives. Així que, en temps d'execució caldrà un procés de monitorització de l'execució de les regles que componen un cicle.

De la mateixa manera que el mètode anterior, el principal problema que té aquest mètode és que no considera totes les possibles formes de reparar una restricció d'integritat. Aquest fet és degut a què davant d'una violació, cal escollir una única regla activa (la de més pes) per a reparar la restricció, quan possiblement hi ha altres regles igualment aplicables. Així doncs les solucions obtingudes depenen directament dels pesos que el dissenyador ha assignat a cada una de les regles que reparen una restricció d'integritat.

7.2 Comparació amb el Mètode dels Esdeveniments

Una comparació a nivell d'eficiència entre el mètode proposat en aquesta tesi i el Mètode dels Esdeveniments és especialment difícil de fer. La raó d'aquesta dificultat rau en el fet que per a fer aquesta comparació, cal comparar la proposta d'implementació del nostre mètode respecte a la definició formal del Mètode dels Esdeveniments.

Així doncs, aquesta comparació s'ha d'entendre com una eina per a posar de manifest les millores en eficiència aconseguides pel nostre mètode respecte al nostre precursor.

Els indicadors utilitzats en aquesta comparació són els mateixos que s'han definit a les seccions 6.4.4 i 6.5.3. Respecte al procés de manteniment de la consistència compararem el nombre de condicions que s'han de comprovar, i respecte al procés d'actualització de vistes, comptabilitzarem els accessos que cal realitzar a la base de dades extensional i el nombre de regles d'esdeveniment que s'han de considerar per a obtenir totes les solucions a una petició d'actualització U .

Com ja s'ha comentat a la secció 5.4, el Mètode dels Esdeveniments està basat, al igual que el nostre, en una alternança entre les derivacions constructives i les derivacions de consistència. Donada una petició d'actualització U , en les derivacions constructives es va construir el conjunt de transacció T afegint-hi els esdeveniments bàsics necessaris per a satisfer la petició d'actualització U o per a reparar una violació d'alguna condició. En les derivacions de consistència es comproven que cada una de les condicions associades al conjunt C i a les restriccions d'integritat no es violin per els esdeveniments incorporats al conjunt T . Es pot veure per tant, que no hi ha clarament diferenciats un procés d'actualització de vistes d'un procés de manteniment de la consistència tal i com ocorria en la formalització del nostre mètode (capítol 4). En canvi, el l'arquitectura proposada al capítol 6 aquesta processos estan clarament diferenciats.

Per a poder diferenciar més clarament aquests dos processos i poder realitzar una comparació entre l'arquitectura del nostre mètode i la definició formal del Mètode dels Esdeveniments, cal fer els supòsits següents:

- Respecte el Mètode dels Esdeveniments, considerarem que les condicions generades durant un procés de traducció de vistes, s'acumularan al conjunt de condicions C , però aquestes no es comprovaran fins al final del procés de traducció, en un procés de manteniment de la consistència junt amb les condicions associades a les restriccions d'integritat.
- Respecte al nostre mètode, solament considerarem els esdeveniment d'inserció i d'esborrat. Per tant, els predicats de la base de dades considerats solament tindran arguments clau.

Actualització de Vistes

La traducció de les peticions d'actualització de vistes en actualitzacions de fets bàsics es realitza bàsicament durant les derivacions constructives. En aquest procés es consideren les regles d'esdeveniment de la $A(D)$ associades als esdeveniments derivats que apareixen a la petició U . Per a obtenir totes les traduccions existents a la petició U , cal tenir en compte totes i cada una d'aquestes regles. Així doncs, l'arbre de resolució associat a una derivació constructiva tindrà una branca per a cada una d'aquestes regles d'esdeveniment, és a dir, existirà una branca per a cada forma alternativa d'obtenir una traducció a la petició U . El problema és que no totes aquestes branques poden assolir l'èxit, i per tant, s'han explorat alternatives que són infructuoses.

Tenint en compte aquesta situació, podem definir un lema en el que es comptabilitzen el nombre de regles d'esdeveniment que s'han de considerar per a obtenir totes les solucions existents a una petició d'actualització d'un fet derivat. Tenint en compte que per un predicat derivat P a la base de dades augmentada $A(D)$ hi ha definides $2^n - 1$ regles d'esdeveniment d'inserció i n regles d'esdeveniment d'esborrat, podem definir el lema següent:

Lema 7.1: Sigui P un predicat derivat definit per una única regla $P(t_1, \dots, t_m) \leftarrow L_1 \wedge \dots \wedge L_n$, amb $n \geq 1$, on els literals L_1, \dots, L_n es corresponen a predicats bàsics i on tota variable que apareix als literals L_1, \dots, L_n també apareix als termes t_1, \dots, t_m . Aleshores, el nombre de regles d'esdeveniments que es consideren en el procés de traducció d'una actualització de P és el següent:

- Una inserció $\iota P(\underline{k}_1, \dots, \underline{k}_m)$: $2^n - 1$ regles d'esdeveniment d'inserció.
- Un esborrat $\delta P(\underline{k}_1, \dots, \underline{k}_m)$: n regles d'esdeveniment d'esborrat.

Per altra banda, en la definició de les regles d'esdeveniment d'un predicat derivat P hi apareixen literals que fan referència la base de dades. Aquests literals es corresponen als predicats Q_i que defineixen el predicat derivat P . Així doncs, a cada branca de l'arbre de

derivació, al considerar una regla d'esdeveniment diferent, es realitzen (o poden realitzar) tantes consultes com predicats defineixen el predicat P.

Lema 7.2: Sigui P un predicat derivat definit per una única regla $P(t_1, \dots, t_m) \leftarrow L_1 \wedge \dots \wedge L_n$, amb $n \geq 1$, on els literals L_1, \dots, L_n es corresponen a predicat bàsics i on tota variable que apareix als literals L_1, \dots, L_n també apareix als termes t_1, \dots, t_m . Aleshores, el nombre d'accessos lògics a la base de dades extensional necessaris per a traduir una petició d'actualització de P és el següent:

- Una inserció $\iota P(k_1, \dots, k_m)$: $(2^n - 1) * n$ accessos a la EDB
- Un esborrat $\delta P(k_1, \dots, k_m)$: n^2 accessos a la EDB.

Si es comparen els lemes anteriors (7.1 i 7.2) amb els lemes definits a la secció 6.5.3 (6.3 i 6.2) respectivament, es pot comprovar que la nostra proposta aporta una important millora en l'eficiència del procés de traducció d'una actualització d'una vista. A la taula 7.1 es mostren de forma resumida aquests lemes i es pot observar més clarament aquesta millora.

	Regles d'esdeveniment		Accessos EDB	
	ι	δ	ι	δ
May00	1	n	n	1
T095	$2^n - 1$	n	$(2^n - 1) * n$	n^2

Taula 7.1 Eficiència en el procés d'actualització de vistes

Concretament, la millora més rellevant fa referència a les peticions d'inserció de fets derivats. Tant en el nombre de regles d'esdeveniment com el nombre d'accessos a la base de dades, el Mètode dels Esdeveniments és exponencial respecte al nombre de literals que defineixen el predicat derivat, mentre que en el nostre cas, solament considerem una regla d'esdeveniment i solament requerim un accés per a cada predicat bàsic que defineix la vista a inserir. En el cas de l'esborrat, el nombre de regles d'esdeveniment considerades és el mateix, però en canvi el nombre d'accessos es redueix a un de sol.

Manteniment de la consistència

En aquest apartat considerarem la millora de l'eficiència mesurada en termes del nombre de condicions que cal comprovar per tal d'assegurar que una transacció T satisfà una petició inicial U i no viola cap restricció d'integritat.

Per tal d'assegurar que una transacció T satisfà aquests requeriments, en els dos mètodes tenim definits dos conjunts de condicions. Un primer conjunt de condicions associades a les restriccions d'integritat que ens assegurin que cap restricció d'integritat esdevindrà violada per l'aplicació d'aquesta transacció T. El segon conjunt és el conjunt C el qual conté les condicions que s'han obtingut durant un procés d'actualització de vistes, que ens assegurin que la transacció T satisfà realment la petició d'actualització U.

El Mètode dels Esdeveniment, a diferència del nostre, no defineix cap ordre específic per a la comprovació d'aquestes condicions. Així doncs, cada cop que es detecti que una condició és violada per T i calgui incorporar nous esdeveniments a T, caldrà comprovar de nou totes les condicions, ja que aquests nous esdeveniments podrien violar alguna de les condicions ja comprovades.

De totes formes considerarem que en mitjana al reparar una condició ja s'han comprovat la meitat de les condicions. Tenint en compte aquest fet podem definir el lema següent:

Lema 7.3 Sigui N el número total de condicions associades a les restriccions d'integritat i a al conjunt de condicions C. Sigui R el nombre total de reparacions que cal fer d'aquestes condicions. Llavors, el nombre de condicions que s'han de comprovar durant el procés de manteniment de la consistència per a obtenir una solució és de $N + R * (N/2)$.

Si es comparen el lema 7.3 amb el lema 6.1 definit a la secció 6.4.4, es pot comprovar que la nostra proposta aporta una important millora en l'eficiència del procés de manteniment de la consistència. A la taula 7.2 es mostren de forma resumida aquests lemes i es pot observar més clarament aquesta millora.

	Condicions a comprovar ²
May00	N
T095	$N + R * (N/2)$

Taula 7.2 Eficiència en el procés de manteniment de la consistència

Observi's que la millora en l'eficiència depèn del nombre de condicions que calgui reparar. En el cas de que cap condició esdevingui violada i reparada ($R=0$) els dos mètodes poden considerar-se igualment eficients tenint en compte únicament aquest indicador.

De totes formes i en qualsevol cas, cal comentar que en el nostre mètode les condicions obtingudes en un procés d'actualització de vistes han estat especialitzades respecte al contingut de la base de dades. Aquest fet fa que el nombre de condicions d'aquest tipus pugui ser menor respecte al Mètode dels Esdeveniments i que aquestes condicions no continguin referències a la base de dades. Així doncs, el procés de comprovació i reparació d'aquestes condicions és més eficient.

² El nombre de condicions N no inclou les condicions de comprovació del tipus ← Ev, amb Ev esdeveniment bàsic, ja que en els dos mètodes es tracten de forma idèntica i es simplifica l'expressió.