Uma Discussão sobre o Mapeamento Relacional Correto de Estruturas Complexas de Especialização em Esquemas Entidade-Relacionamento

Altigran Soares da Silva
DCC - UA
Caixa Postal 1245
69077-000 Manaus AM
alti@dcc.fua.br

Alberto H.F. Laender
DCC - UFMG
Caixa Postal 702
30161-970 Belo Horizonte MG
laender@dcc.ufmg.br

Marco A. Casanova
CCRio - IBM Brasil
Caixa Postal 4624
20071-001 Rio de Janeiro RJ
casanova@vnet.ibm.com

Abstract

The mapping of ER schemas containing complex specialization structures into the relational model requires the use of specific strategies to avoid inconsistent states in the final relational database. In this paper, we generalize a proposed strategy for mapping such structures and caracterize the class of ER schemas for which it generates relational schemas that corectly captures their semantics. We also show that this strategy may be adapted for generating optimized relational schemas for which the number of inclusion dependencies to be enforced is reduced.

1 Introdução

Um dos temas clássicos na área de banco de dados é a representação lógica de esquemas ER, esquemas conceituais construídos segundo o modelo entidade-relacionamento (modelo ER) [Chen76] ou suas extensões ([ElNa94, LaFe89, TeYF86, CTGP89], por exemplo), por meio de esquema relacionais [Codd70]. Isto se justifica pelas características próprias de cada modelo, uma vez que o modelo ER é mais adequado para a construção de abstrações a partir do mundo real, enquanto o modelo relacional dispõe de estruturas de representação que são implementadas em vários SGBDs disponíveis atualmente.

Diversos métodos têm sido propostos para a construção de esquemas relacionais a partir de esquemas ER, processo que chamamos de projeto lógico ou simplesmente **projeto** [ElNa94, Ceri83, TeYF86], sendo que alguns deles incorporam restrições de integridade ao esquema relacional resultante com o objetivo de garantir as propriedades semânticas do modelo ER [CaTL90, CaTL93, MaSh92]. Alguns desses métodos tiveram a sua correção demonstrada [MaSh92, Pach89] ou serviram de base para implementações de ferramentas de projeto [Ceri83, RiCC91, MaSh94].

Em particular, o método de projeto apresentado em [CaTL90, CaTL93] gera esquemas relacionais otimizados onde o número de dependências de inclusão entre as relações é reduzido.

Em [CCRL91, LCCR94] foram apresentados exemplos de esquemas ER que contêm estruturas de especialização complexas [TeYF86, TCGP89] cujo mapeamento exige procedimentos diferentes dos usados no mapeamento de estruturas de especialização envolvendo somente hierarquias.

Apresentamos neste trabalho uma generalização da estratégia usada nesses trabalhos para o mapeamento de estruturas complexas de especialização, argumentando a correção de tal estratégia com base nos resultados obtidos em [Silv95]. Caracterizamos também a classe de esquemas ER para os quais a referida estratégia produz esquemas relacionais que capturam corretamente a semântica das estruturas de especialização.

O trabalho a seguir está organizado da seguinte forma. A Seção 2 apresenta a notação e a terminologia que serão utilizadas ao longo do trabalho. A Seção 3 revisa a estratégia usual para mapeamento de estruturas hieráquicas de especialização. A Seção 4 apresenta a estratégia acima citada para mapeamento de estruturas de especialização complexas, generalizando-a e argumentando a sua correção. A Seção 5 caracteriza a classe de esquemas ER para a qual esta estratégia é válida. A Seção 6 apresenta uma estratégia de mapeamento equivalente à da Seção 4 no que diz respeito à correção do esquema relacional gerado, mas que, no entanto, gera esquemas relacionais otimizados [CaTL90, CaTL93]. Finalmente, a Seção 7 apresenta algumas conclusões e discute os resultados obtidos.

2 Notação e Terminologia

2.1 Esquemas Entidade-Relacionamento

Neste trabalho são utilizados os conceitos de esquema (*schema*) ER, esquemas (*schemes*) de entidade e de relacionamento, e estruturas de especialização. Também são mantidas restrições semânticas usualmente associadas aos relacionamentos e especializações [CTGP89, Silv95].

Um esquema de entidade possui um nome, um conjunto de atributos com seus respectivos domínios e, opcionalmente, uma chave primária. Uma chave é formada por um subconjunto do conjunto de atributos do esquema cujos valores identificam de forma única as instâncias do conjunto de entidades correspondente. Pelo menos um dos atributos de um esquema de entidade deve ser um *discriminador*, ou seja, deve sempre possuir valor diferente de nulo, o que serve para indicar a ocorrência de uma entidade (instância). Os atributos da chave primária são sempre discriminadores.

Um esquema de entidade pode ser *especializado* por um ou mais esquemas de entidade e também pode especializar um ou mais esquemas de entidade, sendo que se um esquema de entidade $\mathtt{E}$ especializa um esquema de entidade $\mathtt{F}$, então toda instância de $\mathtt{E}$ é também uma instância de $\mathtt{F}$. Dizemos que $\mathtt{F}$ é um genérico de $\mathtt{E}$.

A definição de uma chave primária para um esquema de entidade só é obrigatória se este esquema de entidade não especializa nenhum outro. Dizemos que um esquema de entidade *herda* os atributos e as chaves de seus genéricos e assim fazemos a distinção entre estes atributos e chaves, chamados *herdados*, e aqueles definidos no próprio esquema, ditos *nativos*. Em geral, podem ser definidas chaves alternativas para um esquema de entidade. No entanto, por simplicidade, elas não serão usadas neste trabalho, o que não afeta a generalidade dos resultados.

Definimos ainda as seguintes funções:

- Seja $\mathtt{E}$ um esquema de entidade. $\mathcal{I}(\mathtt{E})$ denota o conjunto de instâncias definido por $\mathtt{E}$.
- Seja $\mathtt{E}$ um esquema de entidade e $\mathtt{e} \in \mathcal{I}(\mathtt{E})$. $\mathcal{V}$ é uma *função de valoração*, tal que $\mathcal{V}(\mathtt{e}, \mathtt{A})$ está definida para um atributo $\mathtt{A}$ se, e somente se, $\mathtt{A}$ é um atributo nativo de $\mathtt{e}$ ou $\mathcal{V}(\mathtt{f}, \mathtt{A})$ está definida, sendo $\mathtt{F}$ um genérico de $\mathtt{E}$ e $e = f \in \mathcal{I}(\mathtt{F})$, neste caso $\mathcal{V}(\mathtt{e}, \mathtt{A}) = \mathcal{V}(\mathtt{f}, \mathtt{A})$.

Por questão de simplicidade, escreveremos $\mathcal{V}(\mathtt{e}, A)$ no lugar de $\mathcal{V}(\mathtt{e}, \mathtt{A}_1), \ldots, \mathcal{V}(\mathtt{e}, \mathtt{A}_n)$, onde $A = \{\mathtt{A}_1, \ldots, \mathtt{A}_n\}$ é um conjunto de atributos e $\mathtt{e}$ uma instância qualquer.

Neste trabalho, não nos preocuparemos com a definição e a semântica de relacionamentos, pois assumimos que elas são as usuais e não serão, na verdade, relevantes para a nossa discussão.

Definimos um*grafo de especialização* de um esquema ER como sendo um grafo dirigido $g = (V, A)$, tal que V é o conjunto de nomes de todos os esquemas de entidade do esquema ER e A é um conjunto de arcos tal que um arco $(\mathtt{E}, \mathtt{F})$ pertence a A sse $\mathtt{E}$ é uma especialização de $\mathtt{E}$ no esquema ER. Informalmente, chamaremos de *raiz* um vértice em V do qual não se originam arcos. Se existe em g um caminho de um vértice $\mathtt{V}$ até um vértice $\mathtt{U}$, dizemos que $\mathtt{V}$ é um *descendente* de $\mathtt{U}$ e que $\mathtt{U}$ é um *ascendente* de $\mathtt{V}$. Dizemos ainda que uma estrutura de especialização é *hierárquica* quando o grafo de especialização correspondente é uma árvore.

A definição de esquemas ER será feita usado a sintaxe definida em [Silv95] que é bastante semelhante àquela definida em [CTGP89]. Como os esquemas que serão aqui apresentados são bastante simples, omitiremos a descrição desta sintaxe.

2.2 Esquemas Relacionais

Um esquema relacional consiste de uma dupla $\langle \mathcal{T}, \mathcal{C} \rangle$, onde $\mathcal{T}$ é um conjunto de esquemas de relação e $\mathcal{C}$ é um conjunto de restrições de integridade.

Assim, um esquema de relação possui um nome, uma lista de atributos (cada um com seu respectivo domínio), uma chave primária e, opcionalmente, uma ou mais chaves alternativas. Pode-se especificar que um certo domínio não admite valores nulos. Uma chave primária é uma lista de atributos que não admitem valor nulo e uma chave alternativa é uma lista de atributos quaisquer. Usaremos a notação usual do modelo relacional (que pode ser encontrada, por exemplo, em [ElNa94]) ao longo do trabalho, sendo que para definir esquemas de relação usaremos o padrão SQL descrito em [ANSI90].

As restrições de integridade do conjunto $\mathcal{C}$ podem ser de dois tipos: dependências de inclusão e dependências de nulos.

Dependências de Inclusão. Seja $\mathcal{T}$ um conjunto de esquemas de relação. Uma *dependência de inclusão* [CTFP89] é uma expressão da forma

$$\mathsf{T}_1[X_1] \subseteq \mathsf{T}_2[X_2])$$

onde, para $j = 1, 2$, T_j é o nome de uma relação definida por um esquema de relação em $\mathcal{T}$ e X_j é uma seqüência de nomes de atributos distintos de T_j tal que X_1 e X_2 têm o mesmo número de atributos e o k-ésimo atributo de X_1 tem o mesmo tipo de domínio do k-ésimo atributo de X_2. Uma dependência de inclusão é satisfeita para os estados $t_1(\mathsf{T}_1)$ e $t_2(\mathsf{T}_2)$ se, somente se, para cada tupla t de $t_1(\mathsf{T}_1)$ existe uma tupla u de $t_2(\mathsf{T}_2)$ tal que $t[X_1] = u[X_2]$.

Dependências de Nulos. Os dois tipos de notação definidos abaixo serão usadas para expressar restrições de integridade que serão genericamente denominadas de *dependências de nulos* [SiLC94, Silv95].

Sejam $A = \{\mathsf{A}_1, \mathsf{A}_2, \ldots, \mathsf{A}_n\}$ e $B = \{\mathsf{B}_1, \mathsf{B}_2, \ldots, \mathsf{B}_m\}$ dois subconjuntos do conjunto de atributos de um esquema de relação R. A notação

$$\mathsf{R} : \mathsf{A}_1, \mathsf{A}_2, \ldots, \mathsf{A}_n \rightsquigarrow \mathsf{B}_1, \mathsf{B}_2, \ldots, \mathsf{B}_m$$

será usada para indicar que, se algum atributo A_i for nulo, então todos os B_j são nulos ($i = 1, \ldots, n, j = 1, \ldots, m$). Dizemos que existe uma *dependência de nulos* de B para A, ou seja, se os atributos de A forem nulos, os atributos de B serão necessariamente nulos.

A LDD de diversos SGBDs relacionais permitem especificar que um certo atributo jamais poderá assumir o valor nulo. No contexto de dependências de nulos isto pode ser expresso escrevendo $\mathsf{A}_k \rightsquigarrow \Box$ para um certo atributo A_k.

Seja $C = \{\mathsf{C}_1, \mathsf{C}_2, \ldots, \mathsf{C}_n\}$ um subconjunto de atributos de um esquema de relação R. A notação

$$\mathsf{R} : [\mathsf{C}_1, \mathsf{C}_2, \ldots, \mathsf{C}_n]$$

será usada para indicar que, se um dos atributos C_i for nulo, então todos são nulos. Dizemos que os atributos do conjunto C são *dependentes de nulos entre si (DNE)*, ou seja, que os atributos do conjunto C devem ter todos valor nulo ou todos valor não nulo.

O efeito das definições de dependências de inclusão e de dependências de nulos sobre as operações executadas em uma relação é definido especificando-se opções de propagação associadas a essas restrições. A utilização dessas opções está fora do escopo deste trabalho e não será aqui abordada. Uma discusão detalhada sobre este tópico pode ser encotrada em [Silv95].

3 Mapeamento de Estruturas de Especialização Simples

O Algoritmo 1 a seguir descreve de forma geral o procedimento necessário para mapear corretamente estruturas simples de especialização.

Algoritmo 1

Início

Seja *g um grafo de especilização;*

Para *cada vértice de g correspondente a um esquema de entidade* E *gerar um esquema de relação* T_E, *da seguinte forma:*

Mapear diretamente cada atributo de E *para um atributo de* T_E*;*

Para *cada arco* (E, F) *de g*

Seja T_F *o esquema de relação que mapeia o esquema de entidade* F*;*

Mapear cada atributo da chave primária de T_F *para um atributo de* T_E*;*

Criar uma chave alternativa em T_E *correspondente à chave primária de* T_F*;*

Seja *L o conjunto das chaves alternativas geradas;*

Se E *possuir chave primária K*

Então *mapear K para a chave primária de* T_E*;*

Senão *mapear para a chave primária de* T_E *alguma das chaves alternativas do conjuto L;*

Para *cada arco* (E, F)

Sejam T_E *e* T_F *os esquemas de relação que mapeiam respectivamente* E *e* F*;*

Seja *K a chave primária de* T_F*;*

Gerar uma dependência de inclusão $T_E[K] \subseteq T_F[K]$*;*

Fim.

Note que este algoritmo assume que um esquema de entidade que não possui chave primária definida deve obrigatoriamente especializar algum outro esquema de entidade. Neste caso, a chave primária da relação é escolhida entre as chaves primárias usadas nas relações correspondentes aos esquemas de entidade especializados. Note também que usamos dependências de inclusão ao invés de restrições de integridade referencial [ElNa94] por uma questão de conveniência e que, no âmbito da discussão deste trabalho, estes dois tipos de restrição de integridade são equivalentes. Estas duas observações também serão válidas ao longo deste texto de agora em diante.

Como exemplo de uma estrutura de especialização simples, considere o esquema ER SS cuja declaração é apresentada na Figura 1(a) e cujo grafo é mostrado na Figura 1(b). Seguindo o Algoritmo 1, gerariamos um esquema de relação para cada esquema de entidade, resultando o esquema relacional mostrado na Figura 2

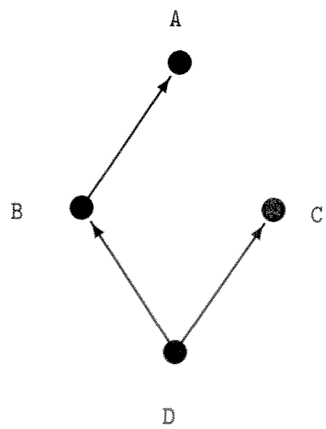
define entity A
atributes K_A char(4) not null,
N_A char(4) not null
key K_A

define entity B
atributes N_B char(4) not null
specialization of A

define entity C
atributes K_C char(4) not null,
N_C char(4) not null
key K_C

define entity D
atributes N_D char(4) not null
specialization of B,C

(a) Definição do Esquema



(b) Grafo

Figura 1: Esquema ER SS

Para esquemas ER como esse, o Algoritmo 1 gera esquemas relacionais cujas restrições de integridade são suficientes para garantir que as instâncias representadas por estados consistentes destes esquemas relacionais preservam as restrições semânticas das estruturas de especialização. Estratégias de mapeamento semelhantes a este algoritmo são descritas em vários trabalhos sobre

CREATE TABLE T_A
(K_A CHAR4) NOT NULL,
N_A CHAR(4) NOT NULL
PRIMARY KEY (K_A))

CREATE TABLE T_C
(K_C CHAR4) NOT NULL,
N_C CHAR(4) NOT NULL
PRIMARY KEY (K_C),

CREATE TABLE T_B
(K_A CHAR4) NOT NULL,
N_B CHAR(4) NOT NULL
PRIMARY KEY (K_A))

CREATE TABLE T_D
(K_C CHAR4) NOT NULL,
K_A CHAR4) NOT NULL,
N_D CHAR(4) NOT NULL
PRIMARY KEY (K_A),
UNIQUE (K_C))

$T_B[K_A] \subseteq T_A[K_A]$

$T_D[K_A] \subseteq T_B[K_A]$

$T_D[K_C] \subseteq T_C[K_C]$

Figura 2: Esquema Relacional SS

projeto lógico de bancos de dados, de modo que não nos deteremos na argumentação de sua correção.

Considere agora o esquema ER DS cuja definição é apresentada na Figura 3(a) e cujo grafo é mostrado na Figura 3(b). Para este esquema ER, seguindo o Algoritmo 1, gerariamos o esquema relacional apresentado na Figura 4.

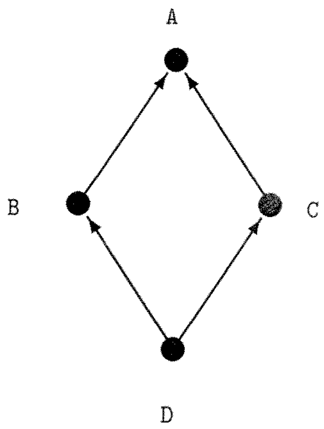
define entity **A**
atributes K_A char(4) not null,
N_A char(4) not null
key K_A

define entity **B**
atributes N_B char(4) not null
specialization of **A**

define entity **C**
atributes K_C char(4) not null,
N_C char(4) not null
key K_C
specialization of **A**

define entity **D**
atributes N_D char(4) not null
specialization of **B,C**

(a) Definição do Esquema



(b) Grafo

Figura 3: Esquema ER SD

CREATE TABLE T_A
(K_A CHAR4) NOT NULL,
A_dNA CHAR(4) NOT NULL
PRIMARY KEY (K_A))

CREATE TABLE T_C
(K_C CHAR4) NOT NULL,
K_A CHAR4) NOT NULL,
N_C CHAR(4) NOT NULL
PRIMARY KEY (K_C),
UNIQUE (K_A))

CREATE TABLE T_B
(K_A CHAR4) NOT NULL,
N_B CHAR(4) NOT NULL
PRIMARY KEY (K_A))

CREATE TABLE T_D
(K_C CHAR4) NOT NULL,
K_A CHAR4) NOT NULL,
N_D CHAR(4) NOT NULL
PRIMARY KEY (K_A),
UNIQUE (K_C))

$T_B[K_A] \subseteq T_A[K_A]$

$T_C[K_A] \subseteq T_A[K_A]$

$T_D[K_A] \subseteq T_B[K_A]$

$T_D[K_C] \subseteq T_C[K_C]$

Figura 4: Esquema Relacional SD1

A Figura 5 mostra um possível estado dessas relações o qual é consistente com as restrições de integridade definidas. No entanto, esta configuração não está de acordo com a semântica das estruturas de especialização.

T_A

K_A	N_A
a1	*x*
a2	*y*

T_B

K_A	N_B
a1	*u*
a2	*w*

T_C

K_A	K_C	N_C
a1	*c1*	*p*
a2	*c2*	*q*

T_D

K_A	K_C	N_D
a2	*c1*	*f*
a1	*c2*	*g*

Figura 5: Estado Inconsistente do Esquema Relacional SD1

Isso pode ser comprovado se notarmos que na relação T_A estão representadas duas entidades, digamos e_1 e e_2, identificadas respectivamente pelos valores $a1$ e $a2$ da chave primária K_A de A, sendo que na relação T_C as entidades e_1 e e_2 também estão representadas mas tendo associados respectivamente os valores $c1$ e $c2$ para a chave K_C de C. No entanto, a relação T_D associa os valores $a1$ e $c2$ a uma mesma instância, o que não pode ocorrer uma vez que estes valores identificam instâncias diferentes. A mesma situação ocorre com $a2$ e $c2$.

4 Mapeamento de Estruturas de Especialização Complexas

A correta representação relacional do esquema ER mostrado na Figura 3 foi apresentada em [CCRL91, LCCR94] e é mostrada na Figura 6. Neste esquema relacional, a chave primária K_A de **A** foi "propagada" ao longo da estrutura de especialização e sobre esta chave foram definidas dependências de inclusão, evitando que o problema encontrado no esquema relacional SD1 ocorra.

```
CREATE TABLE TA
( KA CHAR4) NOT NULL,
  NA CHAR(4) NOT NULL
PRIMARY KEY (KA))

CREATE TABLE TC
( KC CHAR4) NOT NULL,
  KA CHAR4) NOT NULL,
  NC CHAR(4) NOT NULL
PRIMARY KEY (KC),
UNIQUE (KA))

CREATE TABLE TB
( KA CHAR4) NOT NULL,
  NB CHAR(4) NOT NULL
PRIMARY KEY (KA))

CREATE TABLE TD
( KA CHAR4) NOT NULL,
  ND CHAR(4) NOT NULL
PRIMARY KEY (KA))
```

$T_B[K_A] \subseteq T_A[K_A]$

$T_C[K_A] \subseteq T_A[K_A]$

$T_D[K_A] \subseteq T_B[K_A]$

$T_D[K_A] \subseteq T_C[K_A]$

Figura 6: Esquema Relacional SD2

A Figura 7(a) apresenta um outro esquema ER cujo grafo é mostrado na Figura 7(b). Utilizando a mesma estratégia de "propagar" as chaves primárias das raízes do grafo, teríamos o esquema relacional apresentado na Figura 8.

Apresentaremos nesta seção uma argumentação de que a estratégia usada para mapear os esquemas ER das Figuras 3 e 7, originamente proposta em [CCRL91, LCCR94], pode ser generalizada para uma classe de esquemas ER que será aqui caracterizada.

Introduzimos inicialmente o conceito de chave dominante de um esquema de entidade.

Definição 1 *Um conjunto de atributos K é uma chave dominante de um esquema de entidade* E *quando:*

- E *não poussui genéricos e K é a sua chave primária ou*
- *K é chave dominante de algum genérico de* E

Usando a definição da semântica das estruturas de especialização [TCGP89], podemos demonstrar o seguinte Lema:

Lema 1 *Seja um esquema de entidade* E *e $K = \{K_0, \ldots, K_n\}$ $(n \geq 0)$ o conjunto de suas chaves (nativas e herdadas). Uma instância* e *pertence a $\mathcal{I}(\mathtt{E})$ sse $\mathcal{V}(\mathtt{e}, K_i)$ está definido e é igual a $\mathcal{V}(\mathtt{e}', K_i)$, onde $\mathtt{e}' \in \mathcal{I}(\mathtt{E})$ para todo $K_i \in K$*

define entity A
atributes K_A char(4) not null,
N_A char(4) not null
key K_A

define entity B
atributes K_B char(4) not null,
N_B char(4) not null
key K_B

define entity C
atributes N_C char(4) not null
specialization of A

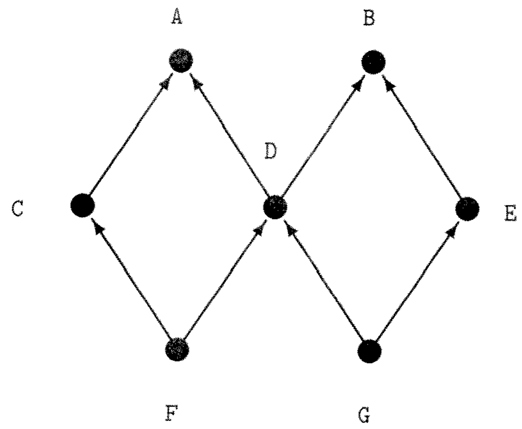
define entity D
atributes N_D char(4) not null
specialization of A,B

define entity E
atributes N_E char(4) not null
specialization of B

define entity F
atributes N_F char(4) not null
specialization of C,D

define entity G
atributes N_G char(4) not null
specialization of D,E

(a) Definição do Esquema



(b) Grafo

Figura 7: Esquema ER DD

O Lema 1 nos garante que uma instância e pertence ao conjunto de instâncias definido por um esquema de entidade E se os valores das chaves de E tomados em e ocorrem em alguma instância e′ de E. Uma consequência deste lema é que se garantirmos que todas as instâncias de um esquema de entidade E satisfazem esta propriedade com relação a um outro esquema F então o conjunto das instâncias de E é um subconjunto das de F, que é o que estabelece o Corolário 1.

Corolário 1 *Sejam* E, F *esquemas de entidade, tal que* $K = \{K_0, \ldots, K_n\}$ $(n \geq 0)$ *é o conjunto de chaves (nativas e herdadas) de* F. *Se para toda instância* $\mathtt{e} \in \mathcal{I}(\mathtt{E})$, $\mathcal{V}(\mathtt{e}, K_i)$ *está definido e existe uma instância* $\mathtt{f} \in \mathcal{I}(\mathtt{F})$ *tal que* $\mathcal{V}(\mathtt{f}, K_i) = \mathcal{V}(\mathtt{e}, K_i)$, *o que deve ocorrer para todo* $K_i \in K$, *então* $\mathcal{I}(\mathtt{E}) \subseteq \mathcal{I}(\mathtt{F})$.

Na realidade, se conseguirmos encontar um subconjunto L das chaves de F tal que todas as outras chaves podem ser determinadas por L então só precisamos garantir a situação do Lema 1 para as chaves deste subconjunto para concluirmos que $\mathcal{I}(\mathtt{E}) \subseteq \mathcal{I}(\mathtt{F})$.

Um subconjunto de K que satisfaz esta propriedade é o das chaves dominantes de F como mostra o Lema 2.

Lema 2 *Seja um esquema de entidade* E *e seja* $L = \{L_0, \ldots, L_n\}$ $(n \geq 0)$ *o conjunto das chaves dominantes de* E. L *determina todas as demais chaves de* E.

```
CREATE TABLE T_A
( K_A CHAR4) NOT NULL,
  N_A CHAR(4) NOT NULL
PRIMARY KEY (K_A))

CREATE TABLE T_C
( K_C CHAR4) NOT NULL,
  K_A CHAR4) NOT NULL,
  N_C CHAR(4) NOT NULL
PRIMARY KEY (K_C),
UNIQUE (K_A))

CREATE TABLE T_B
( K_A CHAR4) NOT NULL,
  N_B CHAR(4) NOT NULL
PRIMARY KEY (K_A))

CREATE TABLE T_D
( K_C CHAR4) NOT NULL,
  K_A CHAR4) NOT NULL,
  N_D CHAR(4) NOT NULL
PRIMARY KEY (K_A),
UNIQUE (K_C))
```

$\mathsf{T_B[K_A]} \subseteq \mathsf{T_A[K_A]}$

$\mathsf{T_C[K_A]} \subseteq \mathsf{T_A[K_A]}$

$\mathsf{T_D[K_A]} \subseteq \mathsf{T_B[K_A]}$

$\mathsf{T_D[K_C]} \subseteq \mathsf{T_C[K_C]}$

Figura 8: Esquema Relacional DD

Prova: *Em primeiro lugar, qualquer chave L_i em $L (0 \leq i \leq n)$ determina as demais, pois todas as instâncias onde este conjunto de chaves dominantes tem valor definido são também instâncias de* `E`*. Como todo L_i é chave em* `E`*, é possível determinar o valor dos demais para a mesma instância. É claro que se* `E` *é uma raiz, só existe um elemento em L que ainda assim é a chave primária de* `E`*.*

Para o restante das chaves, ou seja, as que não pertencem a L, seja uma chave $K_j \in K$, onde K é o conjunto de todas as chaves (nativas e herdadas) de `E`*. Temos então os seguintes casos:*

1. *K_j é uma chave nativa de* `E`*. Neste caso K' e as chaves em L são únicas para cada instância de* `E`*. Portanto, L determina qualquer chave nativa.*

2. *K_j é uma chave herdada por* `E`*. Neste caso, K_j é uma chave nativa de algum ascendente* `F` *de* `E` *que também é descendente de algum ascendente-raiz de* `E` *ou é, ele próprio, uma raiz. Assim, existe um subconjunto L' de L que determina K_j nas instâncias de* `F`*, como mostrado no item 1.*

Assim, chegamos ao Lema 3:

Lema 3 *Sejam dois esquemas de entidade* `E` *e* `F` *e seja $L = \{L_0, \ldots, L_n\}$ $(n \geq 0)$ o conjunto das chaves dominantes de* `F`*. Se para todas as instâncias $\mathtt{e} \in \mathcal{I}(\mathtt{E})$ existe uma instância $\mathtt{f} \in \mathcal{I}(\mathtt{F})$ tal que $\mathcal{V}(\mathtt{f}, L_i) = \mathcal{V}(\mathtt{e}, L_i)$ para todo $L_i \in L$, então $\mathcal{I}(\mathtt{E}) \subseteq \mathcal{I}(\mathtt{F})$.*

Prova: *Pelo Lema 2 sabemos que L determina todas as demais chaves de K, que é o conjunto de todas as chaves (nativas e herdadas) de* `F`*. Agora, pela nossa hipótese sobre o valor das chaves em L tomadas nas instâncias de* `E`*, vemos que se $\mathcal{V}(\mathtt{f}, L) = \mathcal{V}(\mathtt{e}, L)$ então $\mathcal{V}(\mathtt{f}, K) = \mathcal{V}(\mathtt{e}, K)$ e pelo Corolário 1 concluímos que $\mathcal{I}(\mathtt{E}) \subseteq \mathcal{I}(\mathtt{F})$.*

O resultado do Lema 3 é importante, pois estabelece um critério para determinar quando as instâncias de um esquema de entidade `E` formam um subconjunto do conjunto das instâncias de outro esquema de entidade `F`.

Com base neste resultado podemos generalizar a estratégia utilizada nos mapeamentos dos esquemas ER `SD` e `DD`, a qual pode ser descrita conforme o Algoritmo 2 nos mostra.

Algoritmo 2

Início

Seja *g um grafo de especilização;*

Para *cada vértice de g correspondente a um esquema de entidade* E, *gerar um esquema de relação* T_E *da seguinte forma:*

Mapear diretamente cada atributo de E *para um atributo de* T_E;

Seja L o conjunto de chaves dominantes de E

Mapear diretamente cada atributo das chaves em L para um atributo de T_E;

Se E *possuir uma chave primária K*

Então *mapear K para a chave primária de* T_E;

Senão *mapear para a chave primária de* T_E *alguma das chaves do conjuto L;*

Mapear todas as chaves em $L - \{K\}$ como chaves alternativas de T_E;

Para *cada arco* (E, F)

Sejam T_E *e* T_F *os esquemas de relação que mapeiam respectivamente* E *e* F;

Seja *K o conjunto de atributos que compõe as chaves dominantes de* T_F;

Gerar uma dependência de inclusão $\mathsf{T}_E[K] \subseteq \mathsf{T}_F[K]$;

Fim.

Note que pela Definição 1, qualquer chave dominante de F é também chave dominante de E.

Observando o esquema relacional SD1, que mapeia o esquema ER SD, notamos que foi gerada uma dependência de inclusão $\mathsf{T}_D[\mathsf{K}_A] \subseteq \mathsf{T}_C[\mathsf{K}_A]$. Isso garante, segundo o Lema 2, que o valor de K_C tomado em uma instância representada em T_D é necessariamente igual ao valor de K_C na mesma instância tomado na relação T_C, mesmo que este valor não esteja "fisicamente" representado em T_D, o que ocorre porque K_A determina K_C, uma vez que os dois atributos são chaves em T_C.

No primeiro mapeamento do esquema ER SD, o esquema relacional SD1, K_A determina K_C independentemente em T_A e em T_C, sendo este o principal problema deste mapeamento. Note que, para este exemplo, este problema poderia ser corrigido de forma simples se incluíssemos uma dependência de inclusão da forma $\mathsf{T}_D[\mathsf{K}_A, \mathsf{K}_C] \subseteq \mathsf{T}_C[\mathsf{K}_A, \mathsf{K}_C]$. O problema com esta solução é que seria necessário repetir a chave primária K_C de T_C em T_D o que, segundo os Lema 2, é redundante, além do que implicaria em incluir a chave primária de todo esquema de entidade especializado na generalização da solução.

Com relação ao esquema ER DD, podemos notar que os conjuntos de chaves dominantes dos esquemas de entidade são os seguintes:

- A : $\{\{\mathsf{K}_A\}\}$
- B : $\{\{\mathsf{K}_B\}\}$
- D : $\{\{\mathsf{K}_A\}, \{\mathsf{K}_B\}\}$
- C : $\{\{\mathsf{K}_A\}\}$
- E : $\{\{\mathsf{K}_B\}\}$
- F : $\{\{\mathsf{K}_A\}, \{\mathsf{K}_B\}\}$
- G : $\{\{\mathsf{K}_A\}, \{\mathsf{K}_B\}\}$

Assim, as dependências de inclusão geradas para o esquema relacional DD seguem a estratégia apresentada. Note também que mesmo que os esquemas de entidade C,D,E,F e G possuissem chaves primárias nativas, as dependências de inclusão geradas seriam as mesmas.

5 Restrição Estrutural

Caracterizaremos nesta seção a classe de esquemas ER à qual a estratégia de mapeamento apresentada na Seção 4 pode ser aplicada garantindo um mapeamento correto.

Considere o esquema ER ND apresentado na Figura 9. A Figura 10 apresenta um esquema relacional que é o mapeamento deste esquema ER realizado segundo a estratégia descrita na Seção 4. A Figura 11 mostra um estado inconsistente deste esquema relacional onde todas as dependências de inclusão são satisfeitas, mas que inconsistente com a semântica do modelo ER.

Neste exemplo, o estado de T_C mostra que existe uma instância identificada pelo valor $a1$ de K_A e $b1$ de K_B enquanto o estado de T_D mostra que a instância identificada pelo valor $a1$ de K_A está indentificada também pelo valor $b2$ de K_B, o que é inconsistente com a semântica do modelo ER. Isto ocorre porque, além do arco (D, A) do grafo da Figura 9, existem instâncias de D e A indiretamente associadas através da sequência de arcos (C, A), (C, B) e (D, B). Note que se existisse

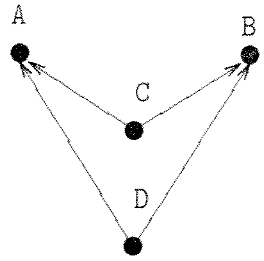
define entity A
atributes K_A char(4) not null,
N_A char(4) not null
key K_A

define entity C
atributes N_C char(4) not null
specialization of A,B

define entity B
atributes K_B char(4) not null,
N_B char(4) not null
key K_B

define entity D
atributes N_D char(4) not null
specialization of A,B

(a) Definição do Esquema



(b) Grafo

Figura 9: Esquema ER ND

CREATE TABLE T_A
(K_A CHAR4) NOT NULL,
N_A CHAR(4) NOT NULL
PRIMARY KEY (K_A))

CREATE TABLE T_C
(K_A CHAR4) NOT NULL,
K_B CHAR4) NOT NULL,
N_C CHAR(4) NOT NULL
PRIMARY KEY (K_A),
UNIQUE (K_B))

CREATE TABLE T_B
(K_B CHAR4) NOT NULL,
N_B CHAR(4) NOT NULL
PRIMARY KEY (K_B))

CREATE TABLE T_D
(K_A CHAR4) NOT NULL,
K_B CHAR4) NOT NULL,
N_D CHAR(4) NOT NULL
PRIMARY KEY (K_A),
UNIQUE (K_B))

$T_C[K_A] \subseteq T_A[K_A]$
$T_C[K_B] \subseteq T_B[K_B]$
$T_D[K_A] \subseteq T_A[K_A]$
$T_D[K_B] \subseteq T_B[K_B]$

Figura 10: Esquema Relacional ND

um arco (D,C) (ou (D,A)), teria sido gerada um dependência de inclusão $T_D[K_A, K_A] \subseteq T_C[K_A, K_B]$ (ou $T_C[K_A, K_B] \subseteq T_D[K_A, K_B]$) que garantiria a correção do mapeamento.

Na realidade, sabemos que pode existir algum vínculo entre as instâncias de C e D. No entanto, como não há um arco explicitando esse vínculo, não é gerada nenhuma restrição de integridade e, portanto, não podemos garantir as condições estabelecidadas pelo Lema 3 e assim garantir que a vinculação seja correta.

Em casos como este do esquema ER ND, não é possível garantir a correção do mapeamento gerado segundo a estratégia da Seção 4. Assim, imporemos uma *restrição estrutural* [Silv95] para os esquemas ER considerados válidos para que se possa gerar um mapemento correto segundo esta estratégia.

Seja um esquema ER cujo grafo de especialização é g. Dizemos que dois vértices A e B de g, correspondentes a esquemas de entidade, possuem um *vínculo* quando existe um conjunto de arcos entre A e B em g. Se A é um ascendente de B, estes dois vértices possuem um *vínculo de ascendência*, se A e B possuem um ascendente comum então existe um *vínculo de co-descendência* e se A e B possuem um descendente comum então existe um *vínculo de co-ascendência*. Vínculos de ascendência, co-descendência e co-ascendência são chamados *vínculos determinantes*. Qualquer outro tipo de vínculo é um *vínculo não determinante*.

A restrição estrutural RE imposta para um esquema ER pode ser assim enunciada [Silv95]:

RE. Em um grafo de especialização g de um esquema ER, se dois vértices A e B correspondentes

T_A

K_A	N_A
a1	*x*
a2	*y*

T_B

K_B	N_B
b1	*u*
b2	*w*

T_C

K_A	K_B	N_C
a1	*b1*	*p*
a2	*b2*	*q*

T_D

K_A	K_B	N_D
a1	*b2*	*f*
a2	*b1*	*g*

Figura 11: Estado Inconsistente do Esquema Relacional ND

a esquemas de entidade ligados por um arco não rotulado possuirem outros vínculos, pelo menos um deles deve ser determinante.

Neste trabalho, consideraremos válidos somente esquemas ER cujos grafos satisfazem à restrição estrutural. Note que, se em um esquema ER todos os esquemas de entidade possuem uma única chave dominante, a restrição estrutural sempre é satisfeita.

No grafo do esquema ER apresentado na Figura 9, desconsiderando o arco (D, A), o único vínculo existente entre os vértices D e A compreende os arcos (C, A), (C, B) e (D, B) e é um vínculo não determinante. Portanto, este esquema ER não é válido com respeito à restrição estrutural *RE*.

Note que o esquema ER correspondente ao grafo resultante da remoção do arco (D, A), por exemplo, é válido, bem como aquele correspondente ao grafo resultante da adição de um arco (D, C), pois, no primeiro, o arco (D, A) não existe e, no segundo, além deste arco existe o vínculo ascendente, e portanto determinante, formado pelos arcos (D, C) e (C, A)

Podemos agora enunciar o Teorema 1:

Teorema 1 *Seja g um grafo de especialização de um esquema ER* $\mathtt{S}_{\mathtt{E}}$. *Se g satisfaz à restrição estrutural RE, então o Algoritmo 2 mapeia corretamente* S *em um esquema relacional.*

6 Otimização do Mapeamento

Com o objetivo de reduzir o número de dependências de inclusão geradas no mapeamento de um esquema ER para um esquema relacional, uma técnica comumente usada é a de se representar em um mesmo esquema de relação vários esquemas de entidade e relacionamento de um esquema ER. Esta técnica é chamada de colapsamento de esquemas [CaTL90, CaTL93].

Seja $g = (V, A)$ um grafo de especialização de um esquema ER $\mathtt{S}_{\mathtt{E}}$. Um grafo de colapsamento para $\mathtt{S}_{\mathtt{E}}$ é um grafo $h = (V, A')$ onde $A' \subseteq A$ e onde para cada vértice V que não é uma raiz em g temos que todo caminho que inicia em V leva à apenas uma raiz. Informalmente, a idéia é gerar um subgrafo de g onde cada vértice que esteja "vinculado" a mais de uma raiz fique vinculado a apenas uma raiz.

Considerando o esquema ER SD, como o seu grafo de especialização possui apenas uma raiz, o seu grafo de colapsamento é o mesmo grafo representado na Figura 3(b), enquanto que para o esquema ER DD, dois possíveis grafos de colapsamento são mostrados na Figura 12.

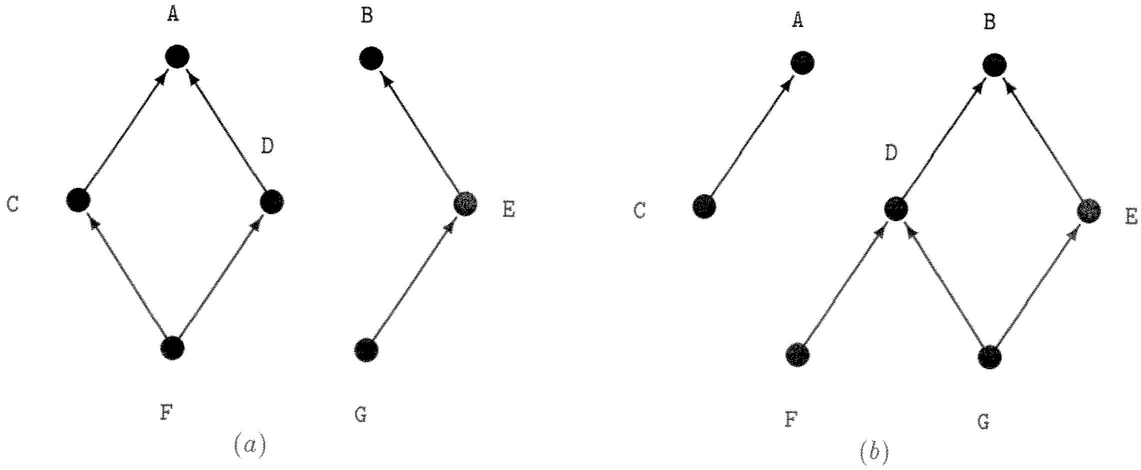


Figura 12: Grafos de Colapsamento para o esquema ER DD

Os grafos de colapsamento determinam quais são os esquema de entidade que serão representados por um mesmo esquema de relação. Por exemplo, para o esquema ER SD usaremos somente um esquema de relação para representar os esquema de entidade `A,B,C` e `D`. Quanto ao esquema ER `DD`, usaremos dois esquemas de relação, cada um correspondente a uma das raizes `A` e `B`. Se escolhermos o grafo de colapsamento da Figura 12(a), serão representados pelo esquema de relação $\mathsf{T}^*_{\mathsf{A}}$ os esquema de entidade `A,C,D` e `F` e pelo esquema de relação $\mathsf{T}^*_{\mathsf{B}}$ os esquemas de entidade `B,E` e `G`. Agora, se escolhermos o grafo de colapsamento da Figura 12(b), serão representados pelo esquema de relação $\mathsf{T}^*_{\mathsf{A}}$ os esquema de entidade `A` e `C` e pelo esquema de relação $\mathsf{T}^*_{\mathsf{B}}$ os esquemas de entidade `B,D,E,F` e `G`. Note que existem outros possíveis grafos de colapsamento.

Para o esquema ER `SD`, o mapeamento otimizado, ou seja, aquele feito segundo o grafo de colapsamento, geraria o esquema relacional SDO apresentado na Figura 13.

```
CREATE TABLE T*A
( KA CHAR4)  NOT NULL,
  NA CHAR(4) NOT NULL,
  NB CHAR4)  ,
  KC CHAR4)  ,
  NC CHAR(4) ,
  ND CHAR(4)
PRIMARY KEY (KA),
UNIQUE (KC)
```

(a) Esquema de Relação $\mathsf{T}^*_{\mathsf{A}}$

DN1. $\mathsf{T}^*_{\mathsf{A}} : [\mathsf{K}_{\mathsf{C}}, \mathsf{N}_{\mathsf{C}}]$
DN2. $\mathsf{T}^*_{\mathsf{A}} : \mathsf{N}_{\mathsf{B}} \rightsquigarrow \mathsf{N}_{\mathsf{D}}$
DN3. $\mathsf{T}^*_{\mathsf{A}} : \mathsf{K}_{\mathsf{C}}, \mathsf{N}_{\mathsf{C}} \rightsquigarrow \mathsf{N}_{\mathsf{D}}$

(b) Dependências de Nulos

Figura 13: Esquema Relacional SDO

A Figura 13(a) apresenta o esquema de relação $\mathsf{T}^*_{\mathsf{A}}$ através do qual todos os esquemas de entidade do esquema ER `SD` serão representados. Considere a Figura 14(a) que apresenta um possível estado da relação $\mathsf{T}^*_{\mathsf{A}}$, onde as tuplas estão numeradas apenas com o propósito de referenciá-las e onde λ indica valor nulo. A tupla 1 representa uma instância que ocorre em $\mathcal{I}(\mathtt{A})$, $\mathcal{I}(\mathtt{B})$, $\mathcal{I}(\mathtt{C})$ e $\mathcal{I}(\mathtt{D})$, enquanto as tuplas 3 e 4 representam instâncias que ocorrem em $\mathcal{I}(\mathtt{A})$ e $\mathcal{I}(\mathtt{B})$ e em $\mathcal{I}(\mathtt{A})$ e $\mathcal{I}(\mathtt{C})$, respectivamente.

Quando geramos esquemas relacionais usando grafos de colapsamento, algumas das dependências inclusão devem ser substituídas por dependências de nulos para garantir a correção do mapeamento. As dependências de nulos que devem ser usadas no esquema relacional SDO estão mostradas na Figura 13(b). Estas dependências de nulos garatem que em qualquer tupla de $\mathsf{T}^*_{\mathsf{A}}$ K_{C} e N_{C} são simultanemente nulos ou não nulos (*DN1*) e que N_{D} só tem valor não nulo se N_{B} tiver valor não nulo (*DN2*) e se K_{C} e N_{C} tiverem ambos valor não nulo (*DN3*).

Seja o estado da relação $\mathsf{T}^*_{\mathsf{A}}$ apresentado na Figura 14(b). A não existência de tais restrições poderia levar à ocorrência de tuplas como as mostradas nesta figura. Por exemplo, a tupla 1 mostra que existe uma instância de `C` para a qual o valor de $\mathtt{N}_{\mathsf{C}}$ é nulo e a tupla 2 representa uma instância que ocorre em $\mathcal{I}(\mathtt{D})$ mas não ocorre em $\mathcal{I}(\mathtt{B})$, o que é inconsistente com a semântica da especialização. A situação da tupla 1 não ocorre em um estado consistente de $\mathsf{T}^*_{\mathsf{A}}$ se a dependência de nulos *DN1* for garantida, da mesma forma que *DN2* e *DN3* evitam que situações como as das tupla 2 e 3, respectivamente, ocorram.

Note que não foi gerada uma dependência de nulos do tipo $\mathsf{T}^*_{\mathsf{A}} : \mathsf{K}_{\mathsf{A}}, \mathsf{N}_{\mathsf{A}} \rightsquigarrow \mathsf{K}_{\mathsf{C}}, \mathsf{N}_{\mathsf{C}}$, uma vez que, pelo esquema de relação $\mathsf{T}^*_{\mathsf{A}}$, K_{A} e N_{A} têm sempre valor não nulo, o que torna sempre válida esta dependência de nulos.

Se observarmos o esquema relacional SD2 (Figura 6), podemos constatar que as dependências de inclusão geradas para os arcos $(\mathtt{B},\mathtt{A})$ e $(\mathtt{C},\mathtt{A})$ deixaram de ser usadas no esquema SDO (estes dois arcos se dirigem para uma raiz) enquanto que as dependências de inclusão geradas para os arcos $(\mathtt{D},\mathtt{B})$ e $(\mathtt{D},\mathtt{C})$ foram substituídas pelas dependências de nulos *DN2* e *DN3* respectivamente.

Esta substituição deve ocorrer sempre que, para um determinado esquema ER, um arco existir tanto no grafo de especialização quanto no grafo de colapsamento utilizado no mapeamento, a menos que o vértice destino do arco seja uma raiz. Além disso, é preciso garantir que em um estado consistente da relação na qual está representada um certo esquema de entidade `E`, todos os atributos declarados como *not null* para `E` possuam todos valor não nulo (nas tuplas da relação

T^*_A	K_A	N_A	N_B	K_C	N_C	N_D
1	*a1*	x	u	*c1*	p	f
2	*a4*	y	λ	λ	λ	λ
3	*a2*	y	w	λ	λ	λ
4	*a3*	x	λ	*c2*	q	λ

(a) Estado Consistente de T^*_A

T^*_A	K_A	N_A	N_B	K_C	N_C	N_D
1	*a1*	x	u	*c1*	λ	f
2	*a2*	y	w	λ	λ	g
3	*a3*	y	λ	*c2*	q	f

(b) Estado Inconsistente de T^*_A

Figura 14: Estados do Esquema Relacional SDO

onde ocorre uma instância de E) ou todos valor nulo (nas tuplas da relação onde não ocorre uma instância de E). Se um arco (E, F) pertence ao grafo de especialização mas não pertence ao grafo de colapsamento, isso indica que E e F são esquemas de entidade que serão representados em esquemas de relação diferentes e, desta forma, deverá ser gerada uma dependência de inclusão adequada.

O procedimento para realizar o mapeamento otimizado de uma estrutura de especialização em um esquema ER pode ser descrito pelo Algoritmo 3.

Algoritmo 3

<u>Início</u>
  <u>Seja</u> *g um grafo de especilização;*
  <u>Seja</u> *f um grafo de colapsamento retirado de g;*
  <u>Para</u> *cada raiz de g correspondente a um esquema de entidade* E*, gerar um esquema de relação* T^*_E *da seguinte forma:*
    Mapear diretamente os atributos E *para* T^*_E*;*
    Mapear a chave primária de E *para a chave primária de* T^*_E*;*
    <u>Seja</u> X *o conjunto de vértices do subgrafo conexo maximal de f cuja raiz é* E*;*
    <u>Para</u> *cada elemento* X *de* X *exceto* E
      Mapear diretamente cada atributo de X *para um atributo de* T^*_E*, como um atributo que pode assumir valores nulos;*
      <u>Seja</u> N *o conjunto de atributos de* X *que não admitem valor nulo;*
      Gerar um dependência de nulos da forma $\mathsf{T}^*_E : [N]$*;*
      Mapear a chave primária de X *como chave alternativa de* T^*_E*;*
      <u>Seja</u> L *o conjunto de chaves dominantes de* E
      Mapear diretamente cada atributo das chaves em L *para um atributo de* T^*_E *somente se este atributo ainda não foi incluído para* T^*_E*;*
      Mapear toda chave $L_i \in L$ *como chave alternativa de* T^*_E*, a não ser que* L *já seja uma chave (primária ou alternativa) de* T^*_E*;*
  <u>Para</u> *cada arco* (E, F) *de g*
    <u>Sejam</u> N *e* M *respectivamente os conjuntos de atributos de* E *e* F *que não admitem valor nulo.*
    <u>Se</u> (E, F) *é um arco de f*
      <u>Então</u>
        Seja R *a raiz à qual* E *e* F *estão vinculados em f;*
        Gerar uma dependência de nulos da forma $\mathsf{T}^*_R : N \rightsquigarrow M$*;*
      <u>Senão</u>
        <u>Sejam</u> P *e* R *as raízes as quais* E *e* F *estão respectivamente vinculados em f;*
        <u>Seja</u> K *o conjunto de atributos que compõem as chaves dominantes de* T_F*;*
        Gerar uma dependência de inclusão da forma $\mathsf{T}^*_P[N \neq \lambda][K] \subseteq \mathsf{T}^*_R[M \neq \lambda][K]$*;*
<u>Fim.</u>

Um esquema relacional construído segundo o Algoritmo 3 deve manter as propriedades semânticas das estruturas de especialização da mesma forma que aqueles gerados segundo o Algoritmo 2. Para os arcos que não são incluídos no grafo de colapsamento, são geradas pelo Algoritmo 3 dependências de inclusão semelhantes a que seriam geradas pelo Algoritmo 2. Para os arcos que ocorrem somente no grafo de especialização, é possível mostrar que as dependências de nulos geradas pelo Algoritmo 3 garantem a condição exigida pelo Lema 3 da mesma forma que as dependências de inclusão que seriam geradas pelo Algoritmo 2 [Silv95].

7 Conclusões

Discutimos neste trabalho os aspectos relacionados à representação correta de estruturas complexas de especialização presentes em esquemas ER através de esquemas relacionais. Vimos que para estruturas de especialização não hierárquicas é necessário utilizar uma estratégia de mapeamento diferente das usualmente aplicadas em métodos de projeto lógico. Vimos também que esta estratégia pode ser adaptada para gerar esquemas relacionais otimizados onde o numéro de dependências de inclusão é reduzido pela eliminação de algumas destas depedências ou pela substituição por dependências de nulos, as quais, por serem restrições de integridade intra-relação, têm custo de manutenção reduzido.

A estratégia apresentada, que se baseia na estratégia delineada em [CCRL91, LCCR94], teve a sua correção argumentada em termos da semântica das estruturas de especialização.

Caracterizamos, ainda, a classe de esquemas ER para a qual pode-se afirmar que a estratégia apresentada gera esquemas relacionais corretos, ou seja, que garantem a representação correta das instâncias relativas ao esquema ER em questão. Esta classe de esquemas ER generaliza a das estruturas hiérarquicas de especialização e também a das estruturas de especialização que apresentam uma só raiz. A argumentação da correção da estratégia foi feita com base na formalização dos métodos de projeto e reprojeto apresentados em [Silv95], onde a correção da representação relacional de esquemas ER é mais profundamente discutida.

Como continuação deste trabalho, uma abordagem semelhante deve ser aplicada ao mapeamento de estruturas de generalização, sendo que discussões sobre esse tópico podem ser encontradas em [MaSh92] e em [LCCR94].

O assunto deste trabalho se insere no contexto do estudo iniciado em [CaTL90, CaTL93]. Como resultados, já foram obtidos o detalhamento, formalização e prova de correção do método de projeto e reprojeto de esquemas relacionais a partir de esquemas ER proposto nesses dois artigos [Silv95]. No momento, parte desse método está sendo implementada na forma de uma ferramenta de projeto/reprojeto, além de um estudo de caso que está sendo desenvolvido para a sua completa validação.

Referências

[ANSI90] ANSI X3H2-90-264 / ISO IEC JTC1 SC21 WG3, "(ISO working draft) Database Language SQL2", Melton, J. (ed.) (Jul. 1990).

[CTFP89] Casanova, M.A., Tucherman, L., Furtado, A.L. and Pacheco, A., "Optimization of relational schemas containing inclusion dependencies", *Proc. 15th Int. Conf. on Very Large Data Bases*, Amsterdam, Holland (Sept. 1989).

[CTGP89] Casanova, M.A., Tucherman, L., Gualandi, P.M., Pacheco, A. and Cavalcanti, M.R., A data definition language for an extended entity-relationship model, Technical Report CCR072, Rio Scientific Center, IBM Brazil (July 1989).

[CaTL90] Casanova, M.A., Tucherman, L. and Laender, A.H.F., "Algorithms for designing and maintaining optimized relational representations of entity-relationship schemas", *Proc. 9th Int. Conf. on Entity-Relationship Approach*, Lausanne, Switzerland (Oct. 1990).

[CaTL93] Casanova, M.A., Tucherman, L. and Laender, A.H.F., "On the design and maintenance of optimized relational representations of entity-relationship schemas" *Data and Knowledge Engineering 11*,1 (1993).

[CCRL91] Casanova, M.A., Carvalho, A.P., Ridolfi, L.F.G.G.M. and Laender, A.H.F. "An analysis of table constraints in SQL2 based on the entity-relationship model", Proc. 10th Int'l. Conf. on Entity-Relationship Approach, San Mateo, California (Oct. 1991).

[Chen76] Chen, P.P., "The entity-relationship model: toward a unified view of data", *ACM Transactions on Database Systems 1*, 1 (1976).

[Ceri83] Ceri, S. (ed.), *Methodology and Tools for Database Design*, North-Holland (1983).

[Codd70] Codd, E.F. "A relational model for large shared data banks", *Communications of the ACM 6*, 13 (1970).

[ElNa94] Elmasri, R. and Navathe, S., *Fundamentals of Database Systems, 2nd ed.*, Benjamin Cummings (1994).

[LCCR94] Laender, A.H.F., Casanova, M.A., Carvalho, A.P. and Ridolfi L.F.G.G.M., "An Analysis of SQL Integrity Constraints from an Entity-Relationship Model Perspective", *Information Systems 4*, 19 (1994).

[LaFe89] Laender, A.H.F and Fernandes, A.C.,"MER+ : Uma extensão do modelo de entidades e relacionamentos para projeto conceitual de bancos de dados", *Revista Brasileira de Computação 5*, 1 (1989).

[MaSh92] Markowitz, V.M. and Shoshani, A., "Representing extended entity-relationship structures in relational databases: a modular approach", *ACM Transactions on Database Systems 17*, 3 (Sept. 1992).

[MaSh94] Markowitz, V.M. and Shoshani, A., "An Overview of the Lawrence Berkeley Laboratory Extended Entity-Relationship Tools", Proc. 13th Int. Conf. on Entity-Relationship Approach, Manchester, UK (Dec. 1994).

[Pach89] Pacheco, A., "Tradução correta de esquemas entidade-relacionamento em esquemas relacionais", Anais do 9o Congresso da Sociedade Brasileira de Computação, Uberlândia, Brasil (Julho 1989).

[RiCC91] Ridolfi, L., Carvalho, A.P. and Casanova, M.A., Uma ferramenta para projeto conceitual de banco de dados baseada no modelo entidade relacionamento, Relatório Técnico CCR130, Centro Científico do Rio, IBM Brasil (Nov. 1991).

[SiLC94] Silva, A.S., Laender A.F.H., Casanova M.A., "Sobre a manutenção da correção semântica de representações relacionais de esquemas ER", Anais do 9o Simpósio Brasileiro de Banco de Dados, São Carlos, Brasil (Setembro 1994).

[Silv95] Silva, A.S., Uma Contribuição para o Problema de Manutenção de Representações Relacionais Otimizadas de Esquemas Entidade-Relacionamento, Dissertação de Mestrado, Departamento de Ciência da Computação, UFMG (1995).

[TeYF86] Teorey, T.J., Yang, D. and Fry, J.P., "A logical design methodology for relational databases using the extended entity-relationship model", *ACM Computing Survey 18*, 2 (June 1986).

[TCGP89] Tucherman, L., Casanova, M.A., Gualandi, P.M. and Pacheco, A., "Generalization and Subset Abstraction in the Entity-Relationship Model", Proc. 8th International Conf. on Entity-Relationship Approach, Toronto, Canada (Oct. 1989).