Universidade Federal de São Carlos
Centro de Ciências Exatas e de Tecnologia
Departamento de Computação
Programa de Pós-Graduação em Ciência da Computação

"O Uso da Teoria de Conjuntos Aproximados na Modelagem de Bases de Dados Relacionais e na Extração de Conhecimento"

João Marcos Vieira

São Carlos - SP
Maio/2005

Ficha catalográfica elaborada pelo DePT da Biblioteca Comunitária da UFSCar

V658ut

Vieira, João Marcos.
O uso da teoria de conjuntos aproximados na modelagem de bases de dados relacionais e na extração de conhecimento / João Marcos Vieira. -- São Carlos : UFSCar, 2005.
154 p.

Dissertação (Mestrado) -- Universidade Federal de São Carlos, 2005.

1. Inteligência artificial. 2. Base de dados relacional aproximada. 3. Conjuntos aproximados. 4. Sistema colaborativo aproximado-simbólico. I. Título.

CDD: 006.3 (20ª)

Aos meus pais
João Batista e Luzia

Às minhas irmãs
Monisa e Mahysa

Aos meus sobrinhos
Bruno, Rafaela e Beatriz

AGRADECIMENTOS

Deus por todas as bênçãos despejadas sobre mim.

Prof[a.] Maria do Carmo Nicoletti pela paciência, dedicação e incentivo durante a orientação deste trabalho de pesquisa.

Prof[a.] Marina T. Pires Vieira pela co-orientação e pelas valiosas contribuições para esta dissertação.

Prof[a.] Solange Rezende e Prof. Mauro Biajiz pelas valiosas contribuições em minha qualificação.

Meus pais pela educação, carinho, apoio moral e financeiro, me possibilitando chegar até aqui.

Meus familiares pelo carinho e apoio.

Minha adorada namorada Jennissy pela compreensão de minha ausência, carinho e apoio moral.

Meus companheiros de república por terem me acolhido e pelos momentos de lazer.

Meus amigos pela verdadeira amizade, apoio moral, horas de conversa e momentos de descontração e lazer.

Colegas de pós-graduação pelas horas de estudo.

Professores e funcionários do Programa de Pós-Graduação do Departamento de Computação da UFSCar.

Programa de Pós-Graduação do Departamento de Computação da UFSCar pela oportunidade.

Capes pelo apoio financeiro.

Eu pedi Força... e Deus me deu dificuldades para me fazer forte.
Eu pedi Sabedoria... e Deus me deu problemas para resolver.
Eu pedi Prosperidade... e Deus me deu cérebro e músculos para trabalhar.
Eu pedi Coragem... e Deus me deu perigo para superar.
Eu pedi Amor... e Deus me deu pessoas com problemas para ajudar.
Eu pedi Favores... e Deus me deu oportunidades.
Eu não recebi nada do que pedi...
Mas eu recebi tudo de que precisava.

Autor Desconhecido

RESUMO

Este trabalho de pesquisa apresenta e investiga dois modelos teóricos de modelagem de bases de dados que incorporam conceitos da Teoria de Conjuntos Aproximados a uma Base de Dados Relacional. O primeiro, o Modelo Relacional Aproximado, incorpora conceitos como a indiscernibilidade buscando dar mais flexibilidade e versatilidade às Bases de Dados Relacionais, tornando a maneira como os dados são tratados mais próxima da maneira como a mente humana os trata. O segundo, o Modelo Relacional Aproximado *Fuzzy*, estende o Modelo Relacional Aproximado agregando conceitos da Teoria de Conjuntos *Fuzzy*, visando representar as relações do modelo por meio de uma função de pertinência *fuzzy*. Isso permite quantificar a pertinência das tuplas às relações da base. Ambos os modelos são implementados tendo os pseudocódigos de seus operadores desenvolvidos e implementados.

Com base nestes modelos é proposto um sistema híbrido que utiliza os conceitos do Modelo Relacional Aproximado e Aproximado *Fuzzy* combinados a um método simbólico de aprendizado para viabilizar a extração de conhecimento certo e conhecimento com certo grau de incerteza, a partir de Bases de Dados Relacionais Aproximadas e Aproximadas *Fuzzy*.

ABSTRACT

This work investigates two relational database models that extend the standard relational database model. Both models extend the standard relational model by allowing ways to represent uncertainty. The rough relational database model borrows the basic concepts form the rough set theory and deals with uncertainty by approaching relations using their lower and upper approximations. The fuzzy rough relational database model generalizes the rough relational model by introducing a degree of membership associated to elements, in a rough relation. The operators that are an intrinsic part of each of the models are formally defined and their pseudocodes are presented and discussed in details. A prototype system ROUGH-ID3, which implements a hybrid knowledge extraction approach by integrating a set of rough database operators with the symbolic system ID3 is proposed.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

CAPÍTULO 1. INTRODUÇÃO

O armazenamento de dados em sistemas computacionais já se tornou uma prática comum e essencial nos dias de hoje, principalmente devido à queda no custo do armazenamento dos dados e à rápida automatização das empresas, que faz com que a quantidade de dados armazenados em bases de dados cresça a uma velocidade muito alta. Essas grandes quantidades de dados, armazenadas em bases de dados, data warehouses e outros tipos de repositórios de dados, de maneira centralizada ou distribuída, existem em muitos domínios, como: financeiro, médico, produção e manufatura, comercial, científico.

O grande volume de dados armazenados, principalmente em empresas e instituições acadêmicas e científicas, vem sendo muito valorizado e analisado, pois muitas informações realmente novas e interessantes estão "embutidas" nessas bases de dados, como: perfis de clientes no uso de cartão de crédito (que podem ser usados para combater fraudes), padrões de pacientes que desenvolveram doenças (que podem ser úteis na tentativa de propor diagnósticos e antecipar tratamentos), perfis de compra de clientes (para usar em futuras promoções). Conforme citado em [Lin e Cercone 1997], as bases de dados das grandes empresas contêm uma potencial mina de ouro de informações valiosas, porém, de acordo com Mitra, em [Mitra et al. 2002], estes dados raramente são obtidos de forma direta. Usualmente, estas informações não estão disponíveis devido à falta de ferramentas apropriadas para a sua extração; está além da capacidade do ser humano analisar tamanha quantidade de dados e extrair relações significativas entre eles.

A área de Mineração de Dados (*Data Mining*) surgiu no final da década de oitenta, e focaliza a extração de conhecimento a partir de grandes volumes de dados usando computador. Devido à sua natureza interdisciplinar, a pesquisa e desenvolvimento da área de Mineração de Dados têm estreitas relações com as contribuições oferecidas por diversas áreas como banco de dados, aprendizado de máquina, estatística, recuperação de informação, computação paralela e distribuída. Como apontado em [Zhou 2003], as áreas de banco de dados, com poderosas técnicas de gerenciamento de dados, aprendizado de máquina, com técnicas práticas de análise de dados e a estatística, com uma sólida fundamentação teórica, são as áreas de conhecimento e pesquisa que estão contribuindo mais efetivamente para o desenvolvimento e o estabelecimento da área de Mineração de Dados.

No artigo [Zhou 2003], o autor analisa as perspectivas destas três áreas, banco de dados, aprendizado de máquina e estatística, e enfatiza os diferentes aspectos de Mineração de Dados abordados por cada uma. De acordo com Zhou, a perspectiva de banco de dados enfatiza a eficiência, uma vez que focaliza o processo de descoberta como um todo, em um volume de dados imenso. A perspectiva de aprendizado de máquina focaliza a efetividade, dado que essa perspectiva é fortemente influenciada por heurísticas efetivas para a análise de dados. A perspectiva da estatística focaliza validade, dado que enfatiza o rigor matemático que subsidia os métodos da mineração.

Como não poderia deixar de ser, dadas as diferentes perspectivas com as quais a área de Mineração de Dados pode ser abordada, na literatura podem ser encontradas diversas caracterizações da área. No artigo [Zhou 2003], o autor evidencia a caracterização da área sob as perspectivas tratadas em três livros sobre Mineração de Dados avaliados por ele, sendo um de cada uma das três áreas. Sob a perspectiva da área de banco de dados, citada em [Han e Kamber 2001], a Mineração de Dados é "*o processo de descoberta de conhecimento interessante em grandes quantidades de dados armazenados em bases de dados, data warehouses ou outros repositórios de dados*"; sob a perspectiva da área de aprendizado de máquina, conforme apontada em [Witten e Frank 2000], é caracterizada como a "*extração de informação implícita, previamente desconhecida e potencialmente útil a partir de dados*"; e sob a perspectiva da área de estatística, conforme citado em [Hand et al. 2001], é "*a análise de conjuntos de dados supervisionados, normalmente em grandes quantidades, para encontrar relacionamentos inesperados e resumir os dados em novas formas que são compreensíveis e úteis para o proprietário dos dados*".

Na caracterização da área de Mineração de Dados é importante, também, discutir a caracterização do que na literatura é chamado de KDD (*Knowledge Discovery in Databases*). De acordo com Frawley (ver [Frawley et al. 1992]), KDD é a "*extração não trivial de informação previamente desconhecida, implícita e potencialmente útil, a partir de dados*". Na literatura existente atualmente as opiniões divergem a respeito dos termos Mineração de Dados e KDD. Existem autores que consideram os termos sinônimos [Mitchell 1999] [Wei 2003] enquanto outros consideram a Mineração de Dados apenas um dos passos do processo de KDD, embora seja o passo principal de todo o processo [Mitra et al. 2002] [Sarafis et al. 2002].

O foco central da pesquisa, seja ela chamada de KDD ou de Mineração de Dados, é o de como analisar dados e transformá-los em conhecimento, expresso em termos de formalismos de representação, como regras e relações entre dados. Existe conhecimento que pode ser extraído diretamente de dados sem o uso de qualquer técnica. Entretanto, existe também muito conhecimento que está de certa forma "embutido" na base de dados, na forma de relações existentes entre itens de dados que, para ser extraído, é necessário o desenvolvimento de técnicas especiais.

A Teoria dos Conjuntos Aproximados (TCA) e seus métodos têm sido usados em Inteligência Artificial (IA) e Ciência Cognitiva nas mais variadas áreas, com ênfase em aquisição e representação de conhecimento [Pawlak 1981] [Kohavi e Frasca 1994] [Komorowski et al. 2002], aprendizado de máquina [Grzymala-Busse et al. 1997] [Lingras 2001] [Nelson 2001] [Krishnaswamy et al. 2002] [Grzymala-Busse e Siddhaye 2004] [Grzymala-Busse 2003] [Grzymala-Busse 2004], sistemas de suporte à decisão e raciocínio indutivo [Pawlak 1995], modelagem de bases de dados [Hu et al. 2004] e KDD [Deogun et al. 1994] [Fernandez-Baizán et al. 1996] [Deogun et al. 1997] [Kusiak 2001].

O uso de técnicas como a TCA na modelagem de uma Base de Dados Relacional, como proposto em [Beauboeuf e Petry 1994] e estudado neste trabalho de pesquisa, possibilita que os dados de uma base de dados possam ser criados, manipulados e interpretados, de uma maneira mais próxima da percepção e do conhecimento humano por meio do uso do conceito de indiscernibilidade, que governa essa teoria. Dados diferentes são interpretados como equivalentes por serem indiscerníveis ou similares de acordo com certo critério. Um dos principais ganhos com a incorporação da TCA é o aumento na recuperação de informações das consultas realizadas, uma vez que é possível retornar, além dos dados que atendem exatamente o que foi solicitado pelo usuário, também aqueles dados que possivelmente atendem a sua real intenção. Essa combinação pode proporcionar perspectivas de abordagens mais flexíveis e versáteis do que aquelas proporcionadas pelos mecanismos inerentes da própria base de dados, pois se consegue uma recuperação aproximada de informações.

A Teoria de Conjuntos *Fuzzy* [Zadeh 1965], assim como a TCA, também tem sido muito utilizada em IA [Markowska-Kaczmar e Trelak 2003] [Jesus et al. 2004] [Alves et al. 2004]. Esse grande uso se deve, dentre outras razões, a sua grande capacidade de representar a incerteza e termos lingüísticos, considerados vagos, inclusive de maneira combinada com a TCA [Sarkar

2002] [Liu et al. 2004]. Tal combinação, incorporada a uma Base de Dados Relacional, proposta em [Beauboeuf et al. 1998] e estudada neste trabalho de pesquisa, possibilita saber o grau de confiança de certa informação retornada, com relação ao que foi solicitado em uma consulta, ou seja, quanto essa informação atende ao que foi solicitado.

Esta proposta de trabalho de pesquisa tem como objetivos principais:

1. Investigar o uso da TCA como formalismo para modelar Bases de Dados Relacionais, por meio do modelo proposto em [Beauboeuf e Petry 1994], focando o refinamento e a padronização do formalismo utilizado e o embasamento necessário para a implementação do referido modelo;
2. Com base na família de Operadores Relacionais Aproximados, propostos em [Beauboeuf e Petry 1994] e [Beauboeuf 2004] baseados nos Operadores Relacionais tradicionais, desenvolver seus pseudocódigos e implementações;
3. Investigar a generalização do Modelo Relacional Aproximado e seus operadores por meio da abordagem *fuzzy*, proposta em [Beauboeuf et al. 1998], também focando o refinamento e a padronização do formalismo utilizado e o embasamento necessário para a implementação do referido modelo;
4. Com base na família de Operadores Relacionais Aproximados *Fuzzy*, propostos em [Beauboeuf et al. 1998] e [Beauboeuf 2004] baseados nos Operadores Relacionais Aproximados, desenvolver seus pseudocódigos e implementações;
5. Propor uma abordagem híbrida de extração de conhecimento a partir de Bases de Dados Relacionais Aproximadas e Aproximadas *Fuzzy*, utilizando Operadores Relacionais Aproximados e Aproximados *Fuzzy* articulados a um método simbólico de aprendizado.

Com foco nos objetivos apontados, este trabalho de pesquisa está organizado da maneira como segue. O CAPÍTULO 2 apresenta uma introdução aos principais conceitos da TCA. O CAPÍTULO 3 apresenta o Modelo Relacional, revendo suas principais características, estrutura, organização e métodos de acesso à informação, enfatizando, principalmente, o formalismo da Álgebra Relacional que subsidia a abordagem dos operadores dos modelos apresentados nesta dissertação.

O CAPÍTULO 4 apresenta o Modelo Relacional Aproximado e seus principais conceitos. O CAPÍTULO 5 apresenta e discute cada um dos Operadores Relacionais Aproximados propondo seus pseudocódigos. O CAPÍTULO 6 apresenta o Modelo Relacional Aproximado *Fuzzy* e seus operadores, cujos conceitos derivam da generalização do Modelo Relacional Aproximado por meio de um tratamento *fuzzy*, e propõe os pseudocódigos de tais operadores.

O CAPÍTULO 7 descreve um sistema híbrido (Sistema ROUGH-ID3) composto por um sistema de consultas aproximadas e aproximadas *fuzzy* e um sistema simbólico de aprendizado, investigando sua colaboração para a extração de conhecimento a partir de Bases de Dados Relacionais Aproximadas e Aproximadas *Fuzzy*.

O CAPÍTULO 8 apresenta as principais conclusões do trabalho e algumas possíveis linhas de pesquisa para a sua continuação.

CAPÍTULO 2. TEORIA DOS CONJUNTOS APROXIMADOS – PRINCIPAIS CONCEITOS E RESULTADOS

A Teoria dos Conjuntos Aproximados (TCA) pode ser considerada uma extensão da Teoria Clássica de Conjuntos. Foi proposta por Pawlak em [Pawlak 1982] como um novo formalismo matemático para tratar incerteza e imprecisão, tendo evoluído a partir de estudos sobre Sistemas de Informação, descritos em [Pawlak 1981]. Existem vários artigos que apresentam e discutem as idéias básicas e o formalismo utilizado na TCA; ver, por exemplo, [Pawlak 1984a], [Pawlak 1985a], [Nicoletti e Uchôa 1997a], [Nicoletti e Uchôa 1998], [Nicoletti e Uchôa 2002], [Pawlak 1991] e [Uchôa 1998].

O conceito de conjuntos aproximados tem sido freqüentemente comparado com o de conjuntos *fuzzy*, da Teoria de Conjuntos *Fuzzy* (TCF) proposta por Zadeh em [Zadeh 1965]; algumas vezes ambos chegam a ser abordados como modelos competitivos [Pawlak 1991] para a representação de conhecimento impreciso. Tal comparação não tem fundamento já que a indiscernibilidade, tratada pela TCA, e a incerteza, tratada pela Teoria dos Conjuntos *Fuzzy* (TCF), são abordagens diferentes para o conhecimento impreciso e ambas podem ser usadas de forma complementar.

Uma das principais vantagens da TCA é poder representar as similaridades conceituais entre os dados de um determinado sistema, agrupando valores que são conceitualmente similares ou equivalentes. Valores que pertencem a um mesmo grupo são considerados indiscerníveis e, assim, o sistema que implementa a TCA pode levar em consideração o significado por trás dos dados e a relação que existe entre eles, e não tratar os seus valores somente de maneira isolada.

O principal objetivo deste capítulo é apresentar e discutir os conceitos e resultados fundamentais da TCA, fornecendo os subsídios conceituais e formais necessários ao uso dessa teoria na abordagem e extensão do Modelo de Base de Dados Relacional. Com esse intuito, também, são abordados alguns conceitos da Teoria de Conjuntos *Fuzzy* (TCF) que podem ser incorporados à TCA para promover uma caracterização mais refinada do Modelo Relacional Aproximado, tratado no CAPÍTULO 4.

2.1 Conceitos Básicos da TCA

Tanto a TCA quanto as Bases de Dados Relacionais são fundamentadas no conceito de relação em um conjunto. Com o objetivo de padronizar a notação e estabelecer o desenvolvimento formal das abordagens tratadas neste trabalho, o ANEXO A define os conceitos matemáticos básicos, e estabelece as notações utilizadas que vão permitir um tratamento rigoroso do formalismo.

A TCA é baseada em relações de equivalência (ver Definição A.5, do ANEXO A). Geralmente a relação de equivalência é estabelecida com base nos valores de atributos que descrevem elementos em um domínio (universo). A relação vai induzir uma partição do universo em classes de equivalência (ver Definição A.7). Cada classe vai conter aqueles objetos que são indiscerníveis entre si, ou seja, aqueles para os quais os valores dos atributos que definem a relação são os mesmos.

Definição 2.1: Um *espaço aproximado* é um par ordenado A = (U, R), onde:

- U é um conjunto finito e não vazio de objetos, denominado *universo*.
- R é uma relação de equivalência em U, denominada *relação de indiscernibilidade.*

  Objetos que pertencem a uma mesma classe de equivalência de R são indiscerníveis em A. Se xRy então x e y são indiscerníveis em A.

Definição 2.2: Seja A = (U, R) um espaço aproximado. As classes de equivalência induzidas por R em U são chamadas de *conjuntos elementares* do espaço aproximado A. Objetos pertencentes a uma mesma classe de equivalência ou conjunto elementar de R são ditos *indiscerníveis* em A.

A partição de U por R, notada por U/R, pode ser vista como o conjunto $\tilde{R} = U/R = \{E_1,...,E_n\}$, onde cada E_i, $1 \leq i \leq n$, é um conjunto elementar de A. Assim, o espaço aproximado A = (U, R) também pode ser notado por $A = (U, \tilde{R})$.

Outra notação que pode ser usada para conjuntos elementares é a convencional de classes de equivalência: dado um conjunto elementar E que possui o elemento x, então E pode ser notado como $[x]_R = \{y \in U \mid xRy\}$. O conjunto vazio $\varnothing$ é assumido como um conjunto elementar para todo espaço aproximado A.

Definição 2.3: Seja A = (U, R) um espaço aproximado. Um *conjunto definível* em A é qualquer união finita de seus conjuntos elementares.

Exemplo 2.1: Seja U = $\{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$ e U/R = $\{\{x_1, x_2\}, \{x_3, x_4, x_5\}, \{x_6\}, \{x_7, x_8\}\}$. A = (U, R) é um espaço aproximado definido pela relação de equivalência R sobre U e está representado na Figura 2.1.

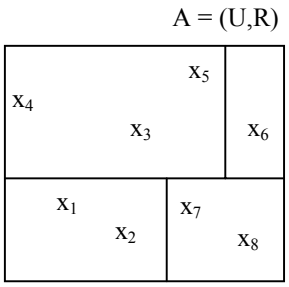


Figura 2.1: O espaço aproximado U/R.

Neste espaço aproximado são conjuntos elementares os conjuntos: $E_1 = \{x_1, x_2\}$, $E_2 = \{x_3, x_4, x_5\}$, $E_3 = \{x_6\}$, $E_4 = \{x_7, x_8\}$ e $E_5 = \varnothing$. São conjuntos definíveis, por exemplo, os conjuntos

$$X_1 = \{x_1, x_2\}, X_2 = \{x_3, x_4, x_5\}, X_3 = \{x_6\}, X_4 = \{x_7, x_8\}, X_5 = \varnothing,$$
$$X_6 = \{x_1, x_2, x_3, x_4, x_5\}, X_7 = \{x_1, x_2, x_6\}, X_8 = \{x_1, x_2, x_7, x_8\}, X_9 = \{x_3, x_4, x_5, x_6\},$$
$$X_{10} = \{x_3, x_4, x_5, x_7, x_8\}, X_{11} = \{x_6, x_7, x_8\} \text{ e } X_{12} = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\} = U$$

Note que X_1, X_2, X_3 e X_4 são conjuntos elementares, X_5 é conjunto elementar por definição e os demais são uniões de conjuntos elementares. O próprio conjunto universo, reescrito como X_{12}, é um conjunto definível, dado que é a união de todos os conjuntos elementares.

A Figura 2.2 mostra os conjuntos definíveis X_1, X_6 e X_9 e o conjunto não-definível $X_{13} = \{x_1, x_2, x_3, x_7\}$ do espaço aproximado A = (U/R).

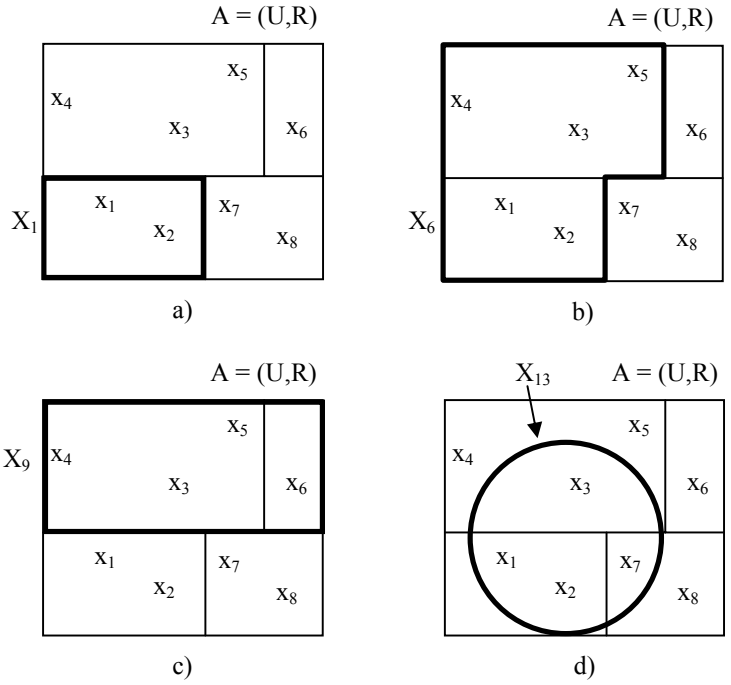


Figura 2.2: a) o conjunto definível X_1; b) o conjunto definível X_6; c) o conjunto definível X_9; d) o conjunto não-definível X_{13}.

2.2 Aproximações de um Conjunto

Considere o conjunto $X_8 = \{x_1, x_2, x_7, x_8\}$ do Exemplo 2.1. Tendo em conta o espaço aproximado do exemplo, esse conjunto pode ser representado, exatamente, como a união dos dois conjuntos elementares $E_1 = \{x_1, x_2\}$ e $E_4 = \{x_7, x_8\}$. Dizer isso é dizer que o espaço aproximado, da maneira como está definido, tem informações suficientes para representar X_8 em termos de suas informações básicas i.e., seus conjuntos elementares. Por outro lado, se o conjunto considerado for o conjunto X_{13}, por exemplo, isso não acontece. Não é possível expressar o conjunto X_{13}, exatamente, em termos dos conjuntos elementares do espaço aproximado em questão. A representação do conjunto X_{13} em termos das informações disponibilizadas pelo espaço aproximado (i.e., seus conjuntos elementares) pode, quanto muito, ser uma representação aproximada. A representação aproximada leva em consideração duas aproximações, a inferior e a superior, formalmente definidas na Definição 2.4.

Definição 2.4: Seja A = (U, R) um espaço aproximado e $X \subseteq U$ um subconjunto arbitrário de objetos de U. A formalização da representação do conjunto X em termos da informação

disponível, isto é, dos conjuntos elementares em A é feita por meio de sua aproximação inferior e superior:

- *aproximação inferior de X em A*: união dos conjuntos elementares de A que estão totalmente contidos em X:

$$A_{A\text{-inf}}(X) = \bigcup_{\substack{E_i \subseteq X \\ E_i \in U/R, 1 \le i \le n}} E_i = \{ x \mid [x]_R \subseteq X \}$$

  ou seja, é o maior conjunto definível em A inteiramente contido em X. A aproximação inferior do conjunto X pode ser considerada como o conjunto dos elementos do conjunto universo que, com certeza, pertencem a X.

- *aproximação superior de X em A*: união dos conjuntos elementares de A que possuem intersecção não-vazia com X:

$$A_{A\text{-sup}}(X) = \bigcup_{\substack{E_i \cap X \neq \varnothing \\ E \in U/R, 1 \le i \le n}} E_i = \{ x \mid [x]_R \cap X \neq \varnothing \}$$

  ou seja, é o menor conjunto definível em A que contém X. A aproximação superior do conjunto X pode ser considerada como o conjunto dos elementos do conjunto universo que, possivelmente, pertencem a X.

Exemplo 2.2: Seja U o conjunto universo e R uma relação de equivalência em U, definindo o espaço aproximado A = (U, R). Seja $X \subseteq U$ um conjunto qualquer definido nesse espaço, como mostra a Figura 2.3. A Figura 2.4 e a Figura 2.5 mostram, respectivamente, a aproximação inferior e superior de X no espaço aproximado considerado.

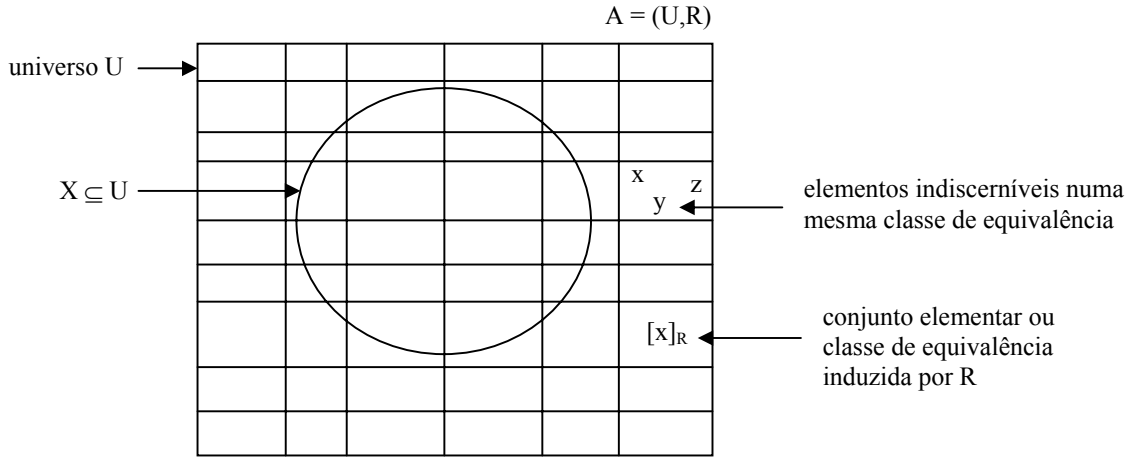


Figura 2.3: Espaço aproximado A = (U, R) e X ⊆ U.

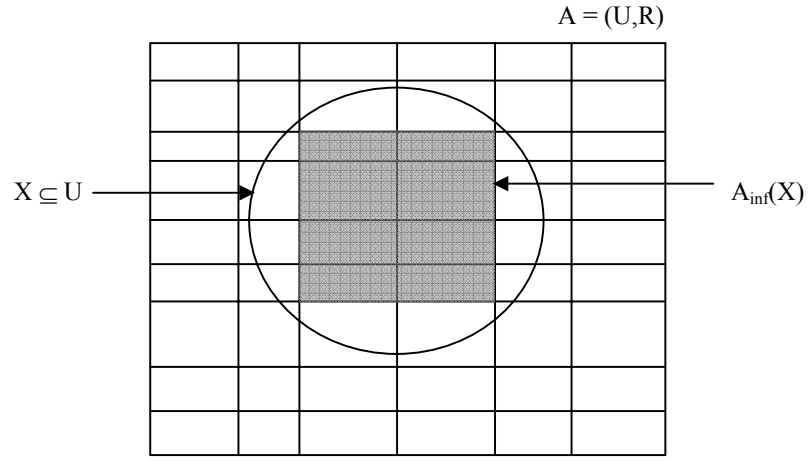


Figura 2.4: Aproximação inferior de X ⊆ U em A = (U, R).

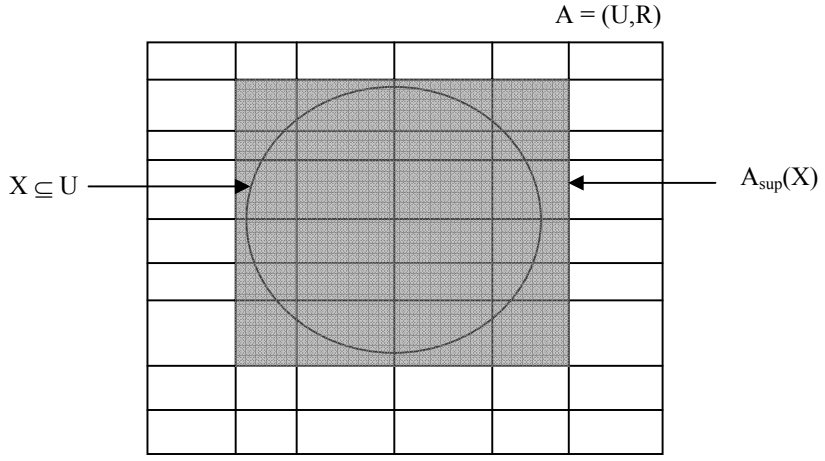


Figura 2.5: Aproximação superior de X ⊆ U em A = (U, R).

Os conceitos de aproximação inferior e superior de um conjunto $X \subseteq U$ em um espaço aproximado $A = (U, R)$ subsidiam a proposta de três conceitos derivados, definidos formalmente a seguir.

Definição 2.5: Seja $A = (U, R)$ um espaço aproximado e $X \subseteq U$. Chama-se

- *região positiva de X em A*, a região formada pelos elementos que podem ser certamente classificados como pertencentes a X, em R, sendo equivalente à aproximação inferior de X em A:

$$\text{pos}_A(X) = A_{A\text{-inf}}(X)$$

- *região negativa de X em A*, a região formada pelos elementos que podem ser certamente classificados como não pertencentes a X, em R, ou ainda, formada pelos conjuntos elementares que não pertencem à aproximação superior de X em A:

$$\text{neg}_A(X) = U - A_{A\text{-sup}}(X)$$

- *região duvidosa de X em A*, a região ou fronteira de X, formada pelos elementos que não se pode ter certeza da pertinência a X, em R.

$$\text{duv}_A(X) = A_{A\text{-sup}}(X) - A_{A\text{-inf}}(X)$$

Quando o espaço aproximado é conhecido e não há risco de confusão, escreve-se $A_{inf}(X)$, $A_{sup}(X)$, pos(X), neg(X) e duv(X), em substituição a $A_{A\text{-inf}}(X)$, $A_{A\text{-sup}}(X)$, $\text{pos}_A(X)$, $\text{neg}_A(X)$ e $\text{duv}_A(X)$, respectivamente, para simplificar a notação e será seguida no presente trabalho.

Exemplo 2.3: Considere o mesmo espaço aproximado $A = (U, R)$ mostrado na Figura 2.3. A região positiva, região negativa e a região duvidosa do conjunto $X \subseteq U$ são mostradas na Figura 2.6, na Figura 2.7 e na Figura 2.8, respectivamente.

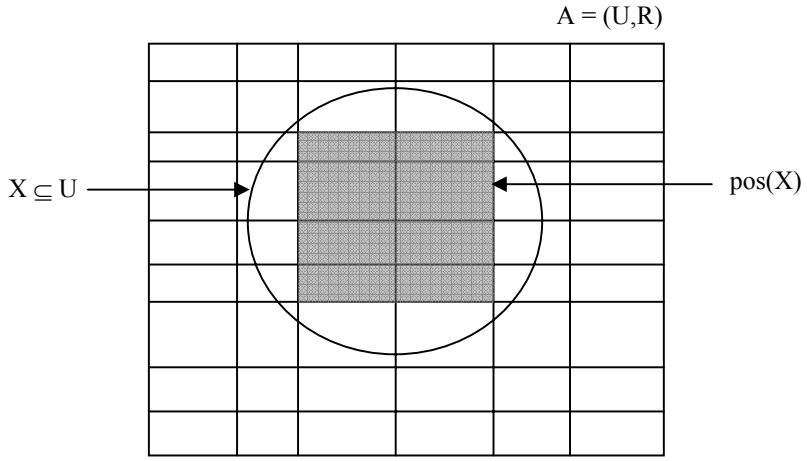


Figura 2.6: Região positiva de $X \subseteq U$.

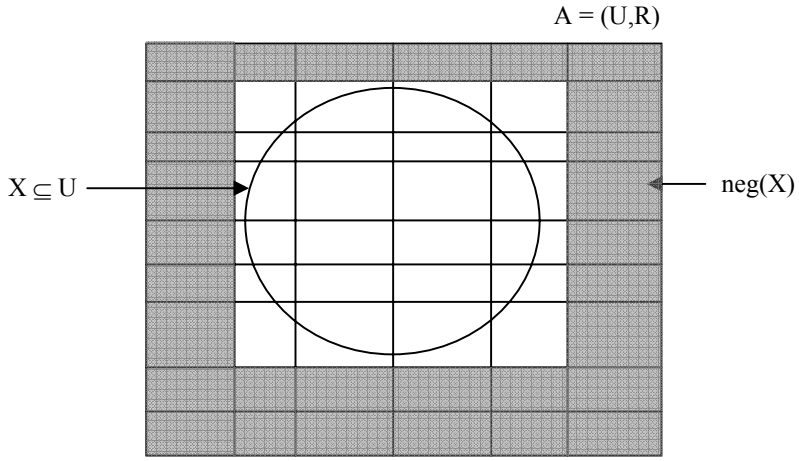


Figura 2.7: Região negativa de $X \subseteq U$.

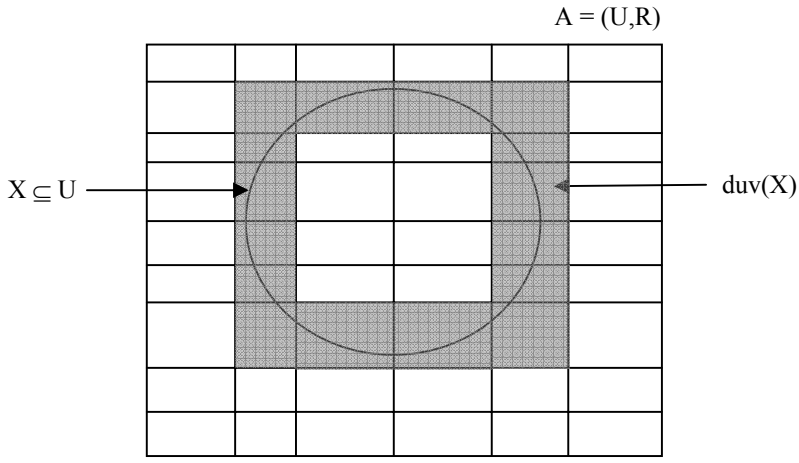


Figura 2.8: Região Duvidosa de $X \subseteq U$.

Exemplo 2.4: Considere o espaço aproximado A = (U, R), onde U = $\{x_1, x_2, x_3,..., x_{11}, x_{12}\}$ e U/R = $\{\{x_1, x_2, x_9\}, \{x_3, x_7\}, \{x_4, x_5\}, \{x_6, x_{11}\}, \{x_8, x_{10}, x_{12}\}\}$. Seja também o conjunto X = $\{x_1, x_3, x_7, x_8, x_{10}, x_{11}, x_{12}\}$, conforme apresentado na Figura 2.9. Então:

$$A_{inf}(X) = pos(X) = \{x_3, x_7\} \cup \{x_8, x_{10}, x_{12}\} = \{x_3, x_7, x_8, x_{10}, x_{12}\}$$

$$A_{sup}(X) = \{x_1, x_2, x_9\} \cup \{x_3, x_7\} \cup \{x_8, x_{10}, x_{12}\} \cup \{x_6, x_{11}\} = \{x_1, x_2, x_3, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}\}$$

$$neg(X) = U - A_{sup}(X) = \{x_4, x_5\}$$

$$duv(X) = A_{sup}(X) - A_{inf}(X) = \{x_1, x_2, x_6, x_9, x_{11}\}$$

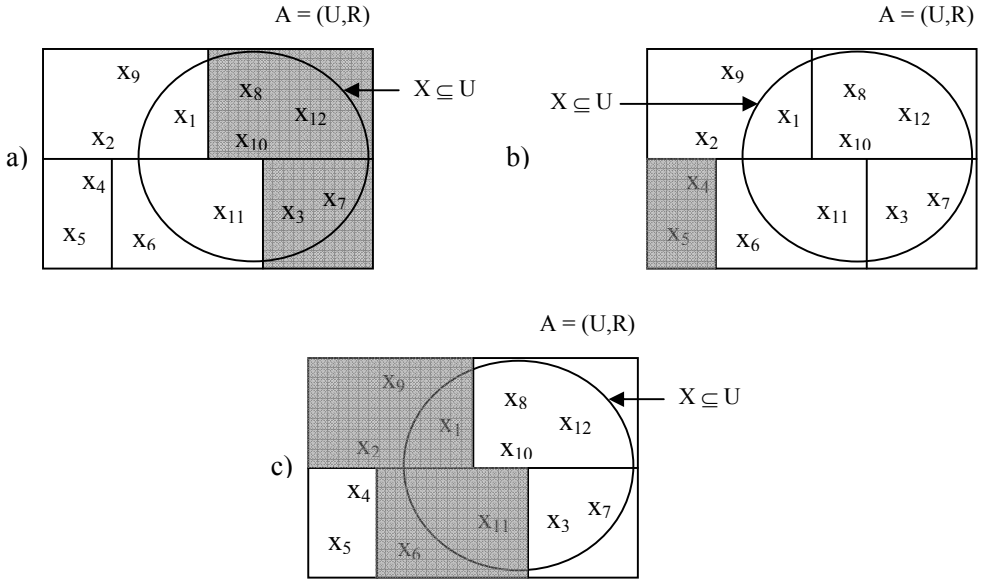


Figura 2.9: A região cinza representa, em a) pos(X); b) neg(X); c) duv(X).

2.3 Extensão dos Conceitos de Igualdade e Inclusão da Teoria Clássica de Conjuntos para a Teoria de Conjuntos Aproximados

Considere um espaço aproximado A = (U, R) e o conjunto X $\subseteq$ U. Nesse espaço, o conjunto X pode ser aproximado por meio de suas duas aproximações, a inferior e a superior. Muitos outros conjuntos, no entanto, podem também ser aproximados com exatamente as mesmas aproximações inferior e superior de X. Esse fato subsidia a idéia de conjuntos aproximados, como formalmente definido na Definição 2.6.

Definição 2.6: Seja A = (U, R) um espaço aproximado e seja $X \subseteq U$. O *conjunto aproximado* X' do conjunto X é a família de todos os subconjuntos de U que possuem a mesma aproximação inferior e a mesma aproximação superior com relação a X em A.

Exemplo 2.5: Seja A = (U, R) o espaço aproximado mostrado na Figura 2.10 e considere o conjunto $X \subseteq U$. Na família de subconjuntos de U, $F = \{X_1, X_2, X_3, ..., X_n\}$, todos os conjuntos têm a mesma aproximação inferior e superior que X e, conseqüentemente, definem um conjunto aproximado X' de X.

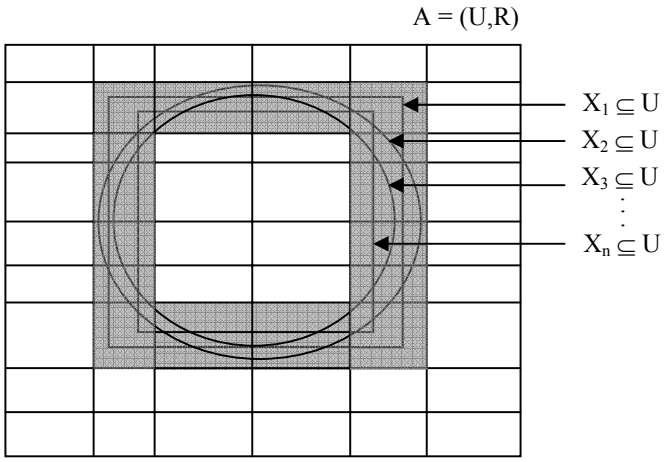


Figura 2.10: Na família de conjuntos $F = \{X_1, X_2, X_3, ..., X_n\}$ todos os conjuntos têm a mesma A_{inf} e A_{sup} e definem um conjunto aproximado de X no espaço aproximado A = (U,R).

Definição 2.7: Dois conjuntos são ditos *aproximadamente iguais* se e somente se possuem as mesmas regiões: positiva, negativa e duvidosa (ver Definição 2.5), ou seja, definem o mesmo espaço aproximado.

- *X é aproximadamente inf-igual a Y*, $X \underset{\sim A}{=} Y$, se e somente se $A_{inf}(X) = A_{inf}(Y)$
- *X é aproximadamente sup-igual a Y*, $X \tilde{=}_A Y$, se e somente se $A_{sup}(X) = A_{sup}(Y)$
- *X é aproximadamente igual a Y*, $X \approx_A Y$, se e somente se $A_{inf}(X) = A_{inf}(Y)$ e $A_{sup}(X) = A_{sup}(Y)$

Definição 2.8: Assim como o conceito de igualdade, a inclusão também pode ser estendida para a TCA.

- *X é aproximadamente inf-incluído a Y*, $X \underset{\sim A}{\subset} Y$, se e somente se $A_{inf}(X) \subseteq A_{inf}(Y)$
- *X é aproximadamente sup-incluído a Y*, $X \tilde{\subset}_A Y$, se e somente se $A_{sup}(X) \subseteq A_{sup}(Y)$
- *X é aproximadamente incluído a Y*, $X \underset{\sim A}{\tilde{\subset}} Y$, se e somente se $A_{inf}(X) \subseteq A_{inf}(Y)$ e $A_{sup}(X) \subseteq A_{sup}(Y)$

2.4 Medidas de uma Aproximação

Seja A = (U, R) um espaço aproximado e considere o conjunto $X \subseteq U$. Para se medir quão bem X pode ser representado em A são definidas as seguintes medidas:

Definição 2.9: A *medida interna de X em A* indica a cardinalidade da aproximação inferior de X em A e é calculada por meio da fórmula:

$$\omega_{A\text{-inf}}(X) = |A_{A\text{-inf}}(X)|$$

Definição 2.10: A *medida externa de X em A* indica a cardinalidade da aproximação superior de X em A e é calculada por meio da fórmula:

$$\omega_{A\text{-sup}}(X) = |A_{A\text{-sup}}(X)|$$

Definição 2.11: A *qualidade da aproximação inferior de X em A* indica a porcentagem de elementos que, com certeza, pertencem a X e é calculada por meio da fórmula:

$$\gamma_{A\text{-inf}}(X) = \frac{\omega_{A\text{-inf}}(X)}{|U|} = \frac{|A_{A\text{-inf}}(X)|}{|U|}$$

Definição 2.12: A *qualidade da aproximação superior de X em A* indica a porcentagem de elementos que, possivelmente, pertencem a X e é calculada por meio da fórmula:

$$\gamma_{A\text{-sup}}(X) = \frac{\omega_{A\text{-sup}}(X)}{|U|} = \frac{|A_{A\text{-sup}}(X)|}{|U|}$$

Definição 2.13: A *acuracidade de X em A* indica a porcentagem de uma decisão ser correta na classificação de um elemento de U com relação à pertinência a X e é calculada por meio da fórmula:

$$\omega_A(X) = \frac{\gamma_{A-inf}(X)}{\gamma_{A-sup}(X)} = \frac{\omega_{A-inf}(X)}{\omega_{A-sup}(X)} = \frac{|A_{A-inf}(X)|}{|A_{A-sup}(X)|}$$

Definição 2.14: O *índice discriminante de X em A* indica a porcentagem de elementos que podem certamente ser classificados como pertencentes ou não a X e é calculado por meio da fórmula:

$$\alpha_A(X) = \frac{|U - duv_A(X)|}{|U|} = \frac{|U - (A_{A-sup}(X) - A_{A-inf}(X))|}{|U|} = \frac{|U| - |A_{A-sup}(X) - A_{A-inf}(X)|}{|U|}$$

Exemplo 2.6: Seja A = (U, R), onde U = $\{x_1, x_2, x_3, ..., x_{11}, x_{12}\}$ e U/R = $\{\{x_1, x_2, x_9\}, \{x_3, x_7\}, \{x_4\}, \{x_5, x_6, x_{11}\}, \{x_8, x_{10}, x_{12}\}\}$ e também o conjunto X = $\{x_1, x_3, x_7, x_8, x_{10}, x_{12}\}$.

- A medida interna de X em A é: $\omega_{A-inf} = |A_{A-inf}(X)| = 5$
- A medida externa de X em A é: $\omega_{A-sup} = |A_{A-sup}(X)| = 8$
- A qualidade da aproximação inferior de X em A é: $\gamma_{A-inf}(X) = \frac{\omega_{A-inf}(X)}{|U|} = \frac{5}{12} = 0.42$
- A qualidade da aproximação superior de X em A é: $\gamma_{A-sup}(X) = \frac{\omega_{A-sup}(X)}{|U|} = \frac{8}{12} = 0.67$
- A acuracidade de X em A é: $\omega_A(X) = \frac{\gamma_{A-inf}(X)}{\gamma_{A-sup}(X)} = \frac{0.42}{0.67} = 0.63$
- O índice discriminante de X em A é: $\alpha_A(X) = \frac{|U| - |A_{A-sup}(X) - A_{A-inf}(X)|}{|U|} = \frac{12-3}{12} = 0.75$

2.5 Tipos de Conjuntos Aproximados

Dependendo do índice discriminante (ver Definição 2.14) (e indiretamente dos conceitos de aproximação inferior e superior (ver Definição 2.4)) conjuntos aproximados podem ser caracterizados como pertencentes a quatro diferentes tipos, definidos a seguir.

Definição 2.15: Seja o espaço aproximado A = (U, R) e X ⊆ U.

- Se $\alpha_A(X) = 1$ ou $A_{inf}(X) = A_{sup}(X)$, diz-se que X é *totalmente definível*.

Neste caso a pertinência dos objetos de X pode ser especificada pelas descrições dos conjuntos elementares induzidos por R, como mostra a Figura 2.11.

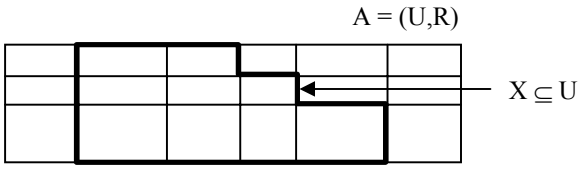


Figura 2.11: X ⊆ U é totalmente definível.

- Se $0 < \alpha_A(X) < 1$ ou $A_{inf}(X) \neq \varnothing$ e $A_{sup}(X) \neq U$, diz-se que X é *parcialmente definível*.

Neste caso existem objetos de U que não podem ser certamente classificados como pertencentes ou não a X. Duas subcategorias podem ser definidas:

- Se $0 < \alpha_A(X) < 1$ e $A_{inf}(X) \neq \varnothing$ e $A_{sup}(X) = U$, diz-se que X é *parcialmente definível* e *externamente indefinível.*

  Neste caso é impossível excluir qualquer elemento de U de ser também elemento de X, como mostra a Figura 2.12.

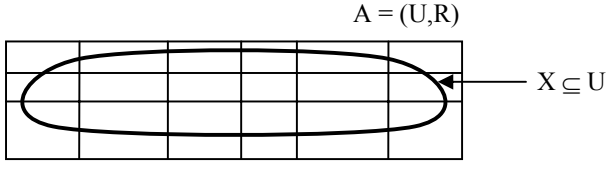


Figura 2.12: X ⊆ U é parcialmente definível e externamente indefinível.

- Se $0 < \alpha_A(X) < 1$ e $A_{inf}(X) = \varnothing$ e $A_{sup}(X) \neq U$, diz-se que X é *parcialmente definível e internamente indefinível.*

  Neste caso é impossível garantir a pertinência de qualquer elemento de U a X, como mostra a Figura 2.13.

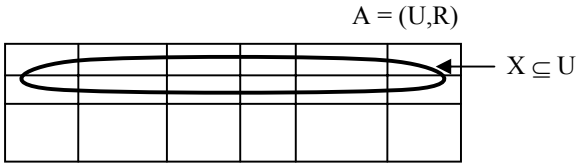


Figura 2.13: $X \subseteq U$ é parcialmente definível e internamente indefinível.

- Se $\alpha_A(X) = 0$ ou $A_{inf}(X) = \varnothing$ e $A_{sup}(X) = U$, diz-se que X é *totalmente indefinível.*

Neste caso é totalmente impossível a especificação da pertinência de objetos ao conjunto X, como mostra a Figura 2.14.

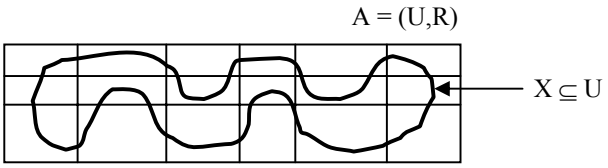


Figura 2.14: $X \subseteq U$ é totalmente indefinível.

2.6 Conjuntos Aproximados e Conjuntos *Fuzzy*

2.6.1 Teoria de Conjuntos *Fuzzy*

Na Teoria Clássica de Conjuntos, dado um conjunto universo U e um conjunto A, onde $A \subseteq U$, pode-se dizer certamente e sem ambigüidade que um determinado elemento pertence ou não pertence ao conjunto *crisp*[1] A. Um conjunto *crisp* A qualquer, definido em um conjunto universo U, pode ser representado por meio de sua função característica, notada por $\mu_A(x):U \rightarrow \{0,1\}$, onde:

$$\mu_A(x) = \begin{cases} 1 & \text{se } x \in A \\ 0 & \text{se } x \notin A \end{cases}$$

A Teoria de Conjuntos *Fuzzy* (TCF) [Zadeh 1965] é uma generalização da Teoria Clássica de Conjuntos, na qual os valores atribuídos pela função característica aos elementos de um conjunto pertencem a um intervalo específico e indicam o grau de pertinência desses elementos com

[1] A palavra *crisp* é usada para referenciar os conjuntos da Teoria Clássica de Conjuntos.

relação ao conjunto em questão. Essa função é chamada de *função de pertinência fuzzy* e o conjunto definido por ela é chamado *conjunto fuzzy* ou *nebuloso*. O intervalo de valores, no qual a função de pertinência *fuzzy* assume valores, usado na literatura é o [0,1]. Cada função de pertinência associa elementos de um conjunto universo, o qual é sempre *crisp*, a valores no intervalo [0,1]. O grau de pertinência 0 indica total exclusão e 1 indica total pertinência.

Definição 2.16: Dado um conjunto universo U, a função de pertinência que caracteriza um conjunto *fuzzy* A é notada por $\mu_A(x):U \rightarrow [0,1]$.

Devido à possibilidade dos elementos de um conjunto *fuzzy* pertencerem a ele com diferentes graus de certeza, termos lingüísticos (quente, frio, pequeno, grande, alto, baixo), considerados vagos na teoria clássica, são muito bem representados através de conjuntos *fuzzy*, conforme mostrado na Figura 2.15.

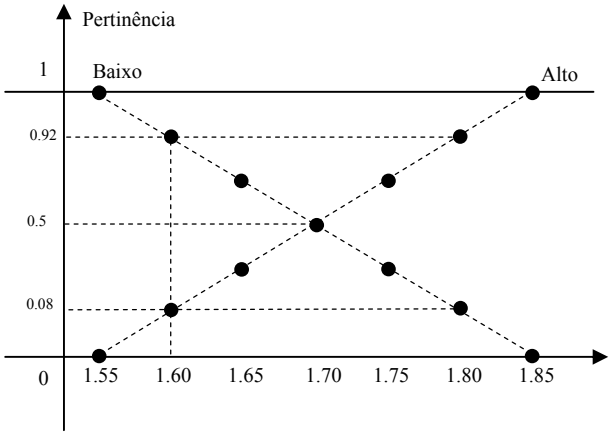


Figura 2.15: Exemplo de conjuntos *fuzzy*.

Dado o conjunto universo $U = \{x_1, x_2, ..., x_n\}$, o conjunto *fuzzy* $A \subseteq U$ pode ser representado por $A = \{\mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + ... + \mu_A(x_n)/x_n\}$, onde o símbolo "+" significa união dos elementos e não soma algébrica, conforme mostrado no Exemplo 2.7.

Exemplo 2.7: Seja o conjunto universo $U = \{1.55, 1.60, 1.65, 1.70, 1.75, 1.80, 1.85\}$[2], representando altura de indivíduos de um determinado grupo. Os conjuntos *fuzzy* Baixo e Alto,

[2] Este trabalho adota o ponto (".") como separador decimal para evitar confusão na notação de conjuntos.

apresentados na Figura 2.15, são representados como segue: Baixo = {1/1.55 + 0.92/1.60 + 0.68/1.65 + 0.5/1.70 + 0.18/1.75 + 0.02/1.80 + 0/1.85} e Alto {0/1.55 + 0.08/1.60 + 0.32/1.65 + 0.5/1.70 + 0.82/1.75 + 0.98/1.80 + 1/1.85}.

Os conceitos de igualdade e continência entre conjuntos *fuzzy* geralmente são abordados de acordo com as seguintes definições [Zadeh 1965]:

Definição 2.17: Dois conjuntos *fuzzy* A e B são *iguais* se e somente se $\mu_A(x) = \mu_B(x)$ para todo $x \in U$.

Definição 2.18: Dados A e B subconjuntos *fuzzy* de U, diz-se que A está *contido* em B, ou A é *subconjunto* de B, $A \subseteq B$, se e somente se $\mu_A(x) \leq \mu_B(x)$ para todo $x \in U$.

Diferentemente das definições anteriores, que são utilizadas em praticamente toda a literatura sobre o assunto, as definições das operações de união, intersecção e complemento possuem diversos operadores definindo-as. Porém, dentro dessa grande variedade, existem alguns, propostos por [Zadeh 1965], que são mais utilizados. Estes são definidos a seguir e seus exemplos utilizam o conjunto universo U e os conjuntos *fuzzy* Baixo e Alto apresentados no Exemplo 2.7.

Definição 2.19: Dado um conjunto *fuzzy* A de U, o *complemento* de A, notado por ¬A, é o conjunto –A tal que $\mu_{-A}(x) = 1 - \mu_A(x)$, $\forall x \in U$.

Exemplo 2.8: A operação *complemento* de Alto está representada na Figura 2.16.

Alto	
1.55	0.00
1.60	0.08
1.65	0.32
1.70	0.50
1.75	0.82
1.80	0.98
1.85	1.00

Complemento

Não-Alto (Baixo)	
1.55	1.00
1.60	0.92
1.65	0.68
1.70	0.50
1.75	0.18
1.80	0.02
1.85	1.00

Figura 2.16: Resultado da operação ¬Alto.

Definição 2.20: Dados A e B subconjuntos *fuzzy* de U, a *união* de A e B, notada por $A \cup B$, é o conjunto C, tal que $\mu_C(x) = \max[\mu_A(x), \mu_B(x)]$. A operação max[a, b], também notada por $a \vee b$, é o maior elemento do conjunto {a, b}.

Exemplo 2.9: A operação *união* de Alto e Baixo está representada na Figura 2.17.

Alto	
1.55	0.00
1.60	0.08
1.65	0.32
1.70	0.50
1.75	0.82
1.80	0.98
1.85	1.00

Baixo	
1.55	1.00
1.60	0.92
1.65	0.68
1.70	0.50
1.75	0.18
1.80	0.02
1.85	0.00

OR

Alto OR Baixo	
1.55	1.00
1.60	0.92
1.65	0.68
1.70	0.50
1.75	0.82
1.80	0.98
1.85	1.00

Figura 2.17: Resultado da operação Alto $\cup$ Baixo.

Definição 2.21: Dados A e B subconjuntos *fuzzy* de U, a intersecção de A e B, notada por $A \cap B$, é o conjunto C tal que $\mu_C(x) = \min[\mu_A(x), \mu_B(x)]$. A operação min[a, b], também notada por $a \wedge b$, é o menor elemento do conjunto {a, b}.

Exemplo 2.10: A operação *intersecção* de Alto e Baixo está representada na Figura 2.18.

Alto	
1.55	0.00
1.60	0.08
1.65	0.32
1.70	0.50
1.75	0.82
1.80	0.98
1.85	1.00

Baixo	
1.55	1.00
1.60	0.92
1.65	0.68
1.70	0.50
1.75	0.18
1.80	0.02
1.85	0.00

AND

Alto AND Baixo	
1.55	0.00
1.60	0.08
1.65	0.32
1.70	0.50
1.75	0.18
1.80	0.02
1.85	0.00

Figura 2.18: Resultado da operação Alto $\cap$ Baixo.

Da maneira como foram definidas acima, as operações complemento, união e intersecção são consideradas um padrão entre conjuntos *fuzzy* e, os operadores max e min são os mais utilizados em Sistemas Baseados em Conhecimento e sistemas de controle. Uma vantagem de se utilizar

esses operadores na definição das operações é a de torná-las as únicas, juntamente com o complemento padrão, que satisfazem as propriedades algébricas de conjuntos, exceto a Lei da Contradição e a Lei do Meio Excluído, descritas abaixo.

- Lei da Contradição: $A \cap (-A) = \varnothing$
- Lei do Meio Excluído: $A \cup (-A) = U$

2.6.2 O uso de TCF na caracterização das regiões da TCA

Como visto anteriormente (ver Definição 2.5), as regiões positiva, duvidosa e negativa de um conjunto X definido em um espaço aproximado, identificam aqueles elementos que com certeza pertencem a X, podem (ou não) pertencer a X e definitivamente não pertencem a X, respectivamente.

Utilizando os conceitos da TCF é possível quantificar a pertinência dos elementos às regiões associadas ao conjunto X, por meio de uma função de pertinência, nos moldes da função de pertinência *fuzzy*, chamada *função de pertinência aproximada*. Além das duas propostas utilizadas neste trabalho para essa função, que são definidas a seguir, também pode ser encontrado mais sobre esse tema em [Nicoletti e Uchôa 1997b] e [Nicoletti e Uchôa 1997a].

I. Primeira Proposta

A primeira função de pertinência aproximada foi proposta em [Pawlak 1985b] e consiste de:

Definição 2.22: Dados um espaço aproximado A = (U, R) e um conjunto aproximado em A, $X \subseteq U$, X pode ser representado por meio de uma função de pertinência definida em U, com valores em {0, 0.5, 1}, como:

$$\mu_X(x) = \begin{cases} 1 & \text{se e somente se } x \in pos(X) \\ 0.5 & \text{se e somente se } x \in duv(X) \\ 0 & \text{se e somente se } x \in neg(X) \end{cases} \quad (2.1)$$

ou equivalentemente,

$$\mu_X(x) = \begin{cases} 1 & \text{se e somente se } x \in A_{inf}(X) \\ 0.5 & \text{se e somente se } x \in A_{sup}(X) - A_{inf}(X) \\ 0 & \text{se e somente se } x \in (U - A_{sup}(X)) \end{cases} \quad (2.2)$$

Como pode ser facilmente visto na Figura 2.19, os possíveis valores para a função de pertinência aproximada, nesta proposta, não caracterizam devidamente os elementos pertencentes à região duvidosa. Isto se deve ao fato de que, dentre estes elementos, existem aqueles que estão mais próximos da fronteira do que outros, dependendo da relação de indiscernibilidade que particiona o universo, das informações que descrevem cada elemento e do próprio conjunto X. Particularmente, note na Figura 2.19 as duas classes de equivalência hachuradas, ambas pertencentes à região duvidosa de X. Todos os elementos de U, dessas duas classes, têm grau de pertinência 0.5 ao conjunto X.

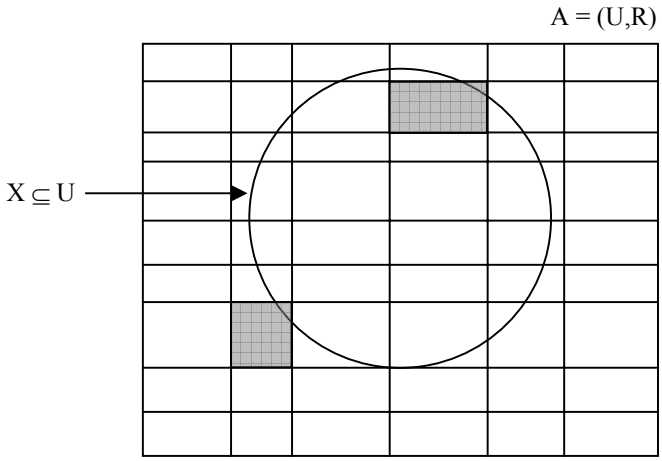


Figura 2.19: Elementos da área hachurada têm grau de pertinência 0.5 ao conjunto X.

Observa-se que esta proposta não contempla as operações de união e intersecção padrão para as funções de pertinência *fuzzy*, conforme provado em [Pawlak 1984b] e [Pawlak 1985b]. Este fato pode ser observado na Figura 2.20 onde, para qualquer x dentro da área hachurada, $\mu_{X\cup Y}(x) \neq \max[\mu_X(x), \mu_Y(x)]$, pois $\mu_{X\cup Y}(x) = 1$ e $\max[\mu_X(x), \mu_Y(x)] = 0.5$, e na Figura 2.21 onde, para qualquer x dentro das áreas hachuradas, $\mu_{X\cap Y}(x) \neq \min[\mu_X(x), \mu_Y(x)]$, pois $\mu_{X\cap Y}(x) = 0$ e $\min[\mu_X(x), \mu_Y(x)] = 0.5$.

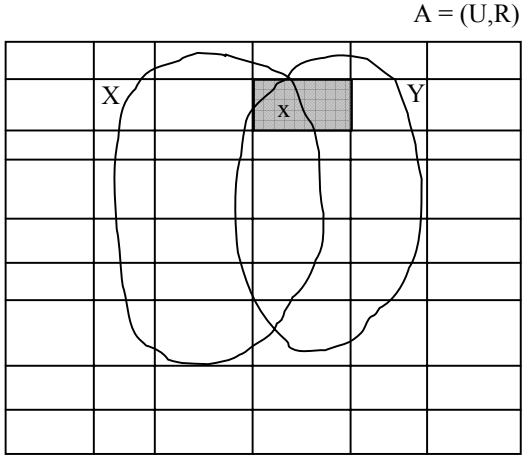


Figura 2.20: Para qualquer ponto x da área hachurada, $\mu_{X\cup Y}(x) \neq \max[\mu_X(x), \mu_Y(x)]$, pois $\mu_{X\cup Y}(x) = 1$ e $\max[\mu_X(x), \mu_Y(x)] = 0.5$.

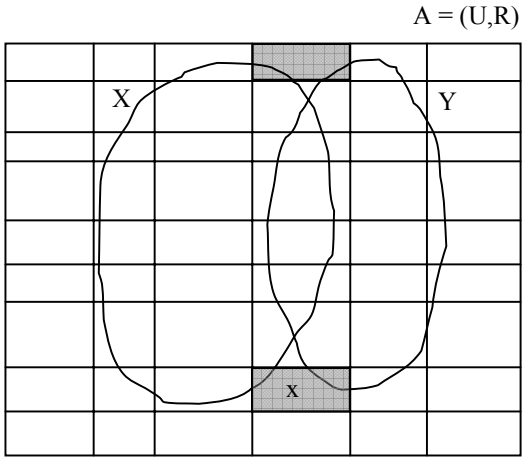


Figura 2.21: Para qualquer ponto x da área hachurada, $\mu_{X\cap Y}(x) \neq \min[\mu_X(x), \mu_Y(x)]$, pois $\mu_{X\cap Y}(x) = 0$ e $\min[\mu_X(x), \mu_Y(x)] = 0.5$.

Com o objetivo de contornar os problemas mostrados na Figura 2.20 e Figura 2.21, em [Wygralak 1989] foi feita uma redefinição das operações de união e intersecção, como:

$$\mu_{X\cup Y}(x) = \begin{cases} \min[1, \mu_X(x) + \mu_Y(x)] & \text{se } \mu_X(x) = \mu_Y(x) = 0.5 \text{ e } [x]_R \subseteq X \cup Y \\ \max[\mu_X(x), \mu_Y(x)] & \text{caso contrário} \end{cases}$$

$$\mu_{X\cap Y}(x) = \begin{cases} \max[0, \mu_X(x) + \mu_Y(x) - 1] & \text{se } \mu_X(x) = \mu_Y(x) = 0.5 \text{ e } [x]_R \cap (X \cap Y) = \varnothing \\ \min[\mu_X(x), \mu_Y(x)] & \text{caso contrário} \end{cases} \quad (2.3)$$

Note, entretanto, que embora os problemas tenham sido solucionados, a proposta continua não representando com precisão a pertinência dos elementos da região duvidosa.

II. Segunda Proposta

A segunda função de pertinência aproximada foi apresentada em [Pawlak 1994a]. Esta proposta consegue representar mais fielmente a pertinência dos elementos de um universo a um conjunto X e consiste de:

Definição 2.23: Dado o espaço aproximado A = (U, R) e o conjunto aproximado em A, $X \subseteq U$, a pertinência de um elemento x, do universo U, a X, pode ser expressa por meio de uma função de pertinência em U, calculada pela fórmula:

$$\mu_X(x) = \frac{|[x]_R \cap X|}{|[x]_R|} \tag{2.4}$$

onde $[x]_R$ denota a classe de equivalência de x, de acordo com a relação de indiscernibilidade R. Note que $[x]_R$ deve possuir pelo menos um elemento, qualquer $x \in U$.

Essa função de pertinência aproximada é calculada com base no conhecimento a respeito dos elementos do universo, ou seja, da partição U/R. Para a medida da pertinência de um elemento $x \in U$ ao conjunto X, usa-se a cardinalidade do conjunto resultante da intersecção de X com a classe de equivalência à qual x pertence, normalizada pelo número de elementos da classe.

O Exemplo 2.11 descreve uma situação de cálculo do valor de pertinência aproximada usando a equação (2.4).

Exemplo 2.11: Seja o espaço aproximado A = (U, R), o conjunto aproximado $X \subseteq U$, e os elementos x e y pertencentes à região duvidosa de X, mostrados na Figura 2.22. Nesta, para facilitar uma perfeita compreensão da equação (2.4), as classes de equivalência que fazem parte da região duvidosa de X têm 2 valores: o número de elementos que a classe compartilha com X e o número de elementos que a classe não compartilha com X. Note que a classe de equivalência à qual x pertence tem 7 elementos, 5 deles pertencentes a X. Já a classe de equivalência à qual y pertence tem 9 elementos, 2 deles também pertencentes a X. A classe de equivalência à qual x pertence compartilha mais elementos com X do que a classe à qual y pertence. A função de

pertinência aproximada (2.4) reflete isso, pois o grau de pertinência aproximado de x a X é maior que o grau de pertinência aproximado de y a X, como demonstra o cálculo abaixo

$$\mu_X(x) = \frac{5}{7} = 0.714 \text{ e } \mu_X(y) = \frac{2}{9} = 0.286$$

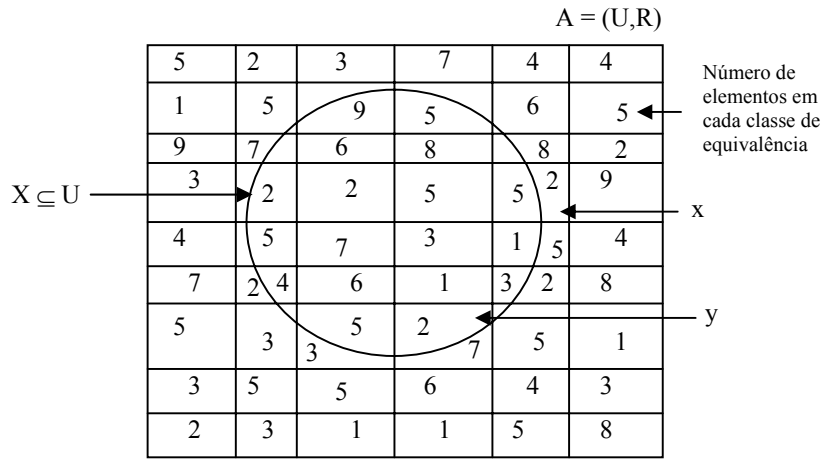


Figura 2.22: Espaço aproximado A = (U, R).

As propriedades que seguem são válidas para as duas propostas de função de pertinência aproximada:

1. $\mu_X(x) = 0 \Leftrightarrow x \in neg(x)$
2. $\mu_X(x) = 1 \Leftrightarrow x \in pos(x)$
3. $0 < \mu_X(x) < 1 \Leftrightarrow x \in duv(x)$
4. $0 \leq \mu_X(x) \leq 1$
5. $\mu_{-X}(x) = 1 - \mu_X(x)$
6. se xRy, então $\mu_X(x) = \mu_X(y)$
7. se $R = \{(x, x) \mid x \in U\}$ então $\mu_X(x)$ é a função característica (função de pertinência *crisp*)
8. $\mu_{X \cup Y}(x) \geq \max[\mu_X(x), \mu_y(x)]$
9. $\mu_{X \cap Y}(x) \leq \min[\mu_X(x), \mu_y(x)]$

A próxima propriedade é válida apenas para a segunda proposta de função de pertinência aproximada:

10. se Y é uma família de subconjuntos de U disjuntos entre si e $Z = \bigcup_{X \in Y} X$, então $\mu_Z(x) = \sum_{X \in Y} \mu_X(x)$, para qualquer $x \in U$

Observa-se que o problema de não contemplar as operações de união e intersecção, mostrado na Figura 2.20 e na Figura 2.21 para a primeira proposta, continua existindo na segunda proposta.

Conforme citado em [Pawlak 1994b], por meio da função de pertinência aproximada, e devido às propriedades da mesma, pode-se reescrever os conceitos de aproximação inferior e superior bem como de região positiva, duvidosa e negativa, como mostrado a seguir:

$$A_{inf}(X) = \{x \in U \mid \mu_X(x) = 1\}$$
$$A_{sup}(X) = \{x \in U \mid \mu_X(x) > 0\}$$
$$pos(X) = \{x \in U \mid \mu_X(x) = 1\}$$
$$duv(X) = \{x \in U \mid 0 < \mu_X(x) < 1\}$$
$$neg(X) = \{x \in U \mid \mu_X(x) = 0\}$$

2.7 Considerações Finais

Este capítulo apresentou os principais conceitos da TCA e alguns conceitos da TCF visando fornecer o formalismo necessário para o uso dessas teorias, principalmente da TCA, na extensão do Modelo Relacional. Visando também fornecer subsídios conceituais e formais necessários para se discutir esta extensão, no próximo capítulo são apresentados e discutidos os conceitos fundamentais do Modelo Relacional e da Álgebra Relacional.

CAPÍTULO 3. O MODELO RELACIONAL E A ÁLGEBRA RELACIONAL

O uso generalizado das bases de dados fez do armazenamento e manutenção de informação uma das aplicações computacionais mais importantes e mais usadas. Tal volume de informação exige sistemas com grandes capacidades de armazenamento e gerenciamento e que usem um modelo de bases de dados adequado pois, conforme citado em [Aho e Ullman 1992], a facilidade de acesso e manutenção da base de dados são profundamente afetadas pela maneira como a informação é organizada. O Modelo Relacional foi introduzido por Codd em [Codd 1970] e pode ser considerado um dos modelos mais simples de base de dados e o que tem estruturas mais uniformes, sendo atualmente um dos modelos mais usados.

O principal objetivo deste capítulo é apresentar os conceitos do Modelo Relacional e da Álgebra Relacional com o intuito de fornecer os conceitos fundamentais e introduzir o formalismo de Bases de Dados Relacionais necessários para abordar a extensão do Modelo Relacional com a TCA, que é um dos principais objetivos deste trabalho.

3.1 Conceitos do Modelo Relacional

Em uma Base de Dados Relacional os dados são representados por meio de uma coleção de relações, que normalmente são representadas em forma de tabelas de dados. Em tais tabelas, cada linha representa uma coleção de valores de dados relacionados que podem ser interpretados como um fato descrevendo uma única entidade ou um relacionamento. A tabela e suas colunas possuem nomes, que ajudam na interpretação do significado dos valores.

De uma maneira informal, na terminologia de Bases de Dados Relacionais, a tabela de dados é chamada de relação, as linhas são chamadas de tuplas e as colunas são chamadas de atributos, que assumem valores de determinados domínios.

Estes e outros termos utilizados em Bases de Dados Relacionais são formalmente definidos na seqüência, conforme apresentado em [Elmasri e Navathe 2003].

Definição 3.1: Um *domínio* D é um conjunto de valores atômicos.

Um método comum de especificar um domínio é o de especificar o tipo do dado que caracteriza os valores de dados do domínio, como pode ser visto no Exemplo 3.1, o qual é bem detalhado pois serve de base para outros exemplos deste capítulo. É útil também especificar o nome do domínio, para ajudar na interpretação de seus valores.

Exemplo 3.1:

Nome do Domínio	Tipo de Dado	Informação Adicional para Interpretação dos Valores de Domínio
Telefone	Cadeia alfanumérica de comprimento 15	Conjunto de números de telefones válidos no formato (dd) dddd-dddd.
Notas	Cadeia numérica de inteiros de comprimento até 2	Conjunto de números inteiros entre 0 e 10.
Conceitos	Caractere alfabético	Conjunto das letras A, B, C, D e E.
Titulos	Cadeia alfanumérica de comprimento até 60	Conjunto de títulos válidos de filmes.
Anos	Cadeia numérica de inteiros de comprimento 4	Conjunto de possíveis anos.
Diretores	Cadeia alfanumérica de comprimento até 60	Conjunto de possíveis nomes de diretores de filmes.
Atores	Cadeia alfanumérica de comprimento até 60	Conjuntos de possíveis nomes de atores de filmes.
Cores	Cadeia alfanumérica de comprimento até 5	Conjunto dos sistemas de cores que um filme pode ser gravado: B & W (Branco e Preto) e Color (Colorido).
Duracao	Cadeia numérica de inteiros de comprimento até 3	Conjunto de números que representam a duração de um filme
Generos	Cadeia alfanumérica de comprimento até 30	Conjunto dos possíveis gêneros de um filme.
Criticas	Cadeia de caracteres do tipo '*' de comprimento até 5	Conjunto dos valores *, **, ***, **** e ***** que representam a avaliação dos críticos sobre um filme.

Definição 3.2: O *esquema de uma relação* R, notado por $R(A_1, A_2, ..., A_n)$ é formado pelo nome da relação, R, e uma lista de atributos $A_1, A_2, ..., A_n$.

- Cada atributo A_i é nome do papel desempenhado por algum domínio D no esquema da relação R. D é chamado de domínio de A_i e denotado por $dom(A_i)$;
- Um esquema de relação é usado para descrever a relação;

- O grau da relação é o número de atributos de seu esquema de relação.

Na Base de Dados Relacional podem ser especificadas restrições com relação aos dados, visando manter a integridade dos mesmos dentro da base. Dentre as restrições que fazem parte de uma Base de Dados Relacional está a Restrição de Domínio que será definida a seguir, sendo as demais definidas no decorrer do capítulo.

Definição 3.3: A *Restrição de Domínio* especifica que o valor de cada atributo A_i deve ser um valor atômico do domínio que o atributo A_i representa.

Exemplo 3.2: O esquema de relação FILME(TITULO, ANO, DIRETOR, ATOR_PRINC_1, ATOR_PRINC_2, COR, DURACAO, GENERO, CRITICA), de grau 9, descreve os filmes de uma locadora e tem os seguintes domínios, definidos no Exemplo 3.1: dom(TITULO) = Titulos, dom(ANO) = Anos, dom(DIRETOR) = Diretores, dom(ATOR_PRINC_1) = Atores, dom(ATOR_PRINC_2) = Atores, dom(COR) = Cores, dom(DURACAO) = Duracao, dom(GENERO) = Generos e dom(CRITICA) = Criticas.

Definição 3.4: Uma *relação* (ou *instância de uma relação*) r do esquema da relação $R(A_1, A_2, ..., A_n)$ é um conjunto de tuplas (linhas da tabela de dados) e é denotada por $r = \{t_1, t_2, ..., t_m\}$. Cada tupla t é uma lista ordenada de n valores, denotada por $t = <v_1, v_2, ..., v_n>$, onde cada valor v_i, $1 \leq i \leq n$, é um elemento de $dom(A_i)$ ou o valor especial *null*.

Observações:

- O valor *null* é usado sempre que um atributo, para uma tupla em particular, tiver seu valor desconhecido ou não possuir valor;
- A notação $t[A_i]$ é usada para representar o valor v_i do atributo A_i na tupla t;
- A notação $t[A_u, A_w, ..., A_z]$, onde $A_u, A_w, ..., A_z$ é uma lista de atributos de R, representa a sub-tupla de valores $<v_u, v_w, ..., v_z>$ de t, correspondentes aos atributos especificados na lista.

A Definição 3.5 é uma reescrita da Definição 3.4, na qual a lista ordenada de atributos é estabelecida pelo produto cartesiano, como segue.

Definição 3.5: Uma *relação* r(R) é um subconjunto do produto cartesiano dos domínios que definem R.

$$r(R) \subseteq dom(A_1) \times dom(A_2) \times \cdots \times dom(A_n)$$

O produto cartesiano especifica todas as possíveis combinações de valores dos domínios. Assumindo que todos os domínios são finitos, o número total de tuplas no produto cartesiano é

$$| dom(A_1) | * | dom(A_2) | * \cdots * | dom(A_n) |$$

Considere um esquema de relação $R(A_1, A_2, ..., A_n)$. Dentre todas as possíveis combinações disponibilizadas pelo produto cartesiano na Definição 3.5, uma instância da relação, num dado tempo – *estado corrente da relação* – reflete apenas as tuplas válidas que representam um determinado estado do mundo real. Em geral, como o estado do mundo real muda, a relação também muda, sendo transformada em um outro estado da relação. O esquema da relação R, entretanto, é relativamente estático e não muda (exceto em raras situações – como aquela, por exemplo, em que um atributo deve ser adicionado para representar uma nova informação que não estava originalmente armazenada na relação).

É possível que vários atributos tenham o mesmo domínio; quando isso acontece, eles indicam diferentes papéis, ou interpretações do domínio.

Exemplo 3.3: Seja o esquema da relação FILME(TITULO, ANO, DIRETOR, ATOR_PRINC_1, ATOR_PRINC_2, COR, DURACAO, GENERO, CRITICA) apresentado no Exemplo 3.2 e a instância de relação $r(FILME) = \{t_1, t_2, t_3, t_4\}$. Onde

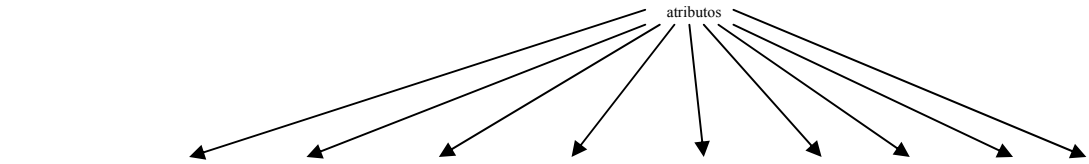
t_1 = <‘The Day World Ended’, ‘1956’, ‘Roger Corman’, ‘Richard Denning’, *Null*, ‘B&W’, ‘82’, ‘Ficção’, ‘**’>,

t_2 = <‘The Boat (Das Boat)’, ‘1981’, ‘Wolfang Petersen’, ‘Jurgen Prochbow’, ‘Herbert Gronemeyer’, ‘Color’, *Null*, ‘Guerra’, ‘*****’>,

t_3 = <‘David and Bathsheba’, ‘1951’, ‘Henry King’, ‘Gregory Peck’, ‘Susan Hayward’, ‘Color’, ‘116’, ‘Épico’, ‘**’>, e

t_4 = <‘Dave’, ‘1993’, ‘Ivan Reitman’, ‘Kevin Kline’, *Null*, ‘Color’, ‘100’, ‘Comédia’, ‘****’>.

A relação FILME está representada na Figura 3.1.



FILME	TITULO	ANO	DIRETOR	ATOR_PRINC_1	ATOR_PRINC_2	COR	DURACAO	GENERO	CRITICA
tuplas	The Day World Ended	1956	Roger Corman	Richard Denning	Null	B&W	82	Ficção	**
	The Boat (Das Boat)	1981	Wolfang Petersen	Jurgen Prochbow	Herbert Gronemeyer	Color	Null	Guerra	*****
	David and Bathsheba	1951	Henry King	Gregory Peck	Susan Hayward	Color	116	Épico	**
	Dave	1993	Ivan Reitman	Kevin Kline	Null	Color	100	Comédia	****

Figura 3.1: Atributos e tuplas da relação FILME.

Exemplo 3.4: Seja a tupla t_2 = <'The Boat (Das Boat)', '1981', 'Wolfang Petersen', 'Jurgen Prochbow', 'Herbert Gronemeyer', 'Color', *Null*, 'Guerra', '*****'>, extraída do Exemplo 3.3. Então: t_2[GENERO] = <'Guerra'>, t_2[TITULO] = <'The Boat (Das Boat)'> e t_2[TITULO, DURACAO, ATOR_PRINC_1] = <'The Boat (Das Boat)', *Null*, 'Jurgen Prochbow'>.

O Modelo Relacional não pressupõe uma ordem na especificação das tuplas que fazem parte da relação. É possível, entretanto, que uma ordem seja especificada, usando como referência os valores de um ou mais atributos. Seguindo essa idéia pode-se dizer que a Figura 3.2 mostra uma relação que é considerada idêntica à relação mostrada na Figura 3.1.

Já os valores que definem uma tupla são ordenados de acordo com o esquema da relação, seguindo a ordem dos atributos. Apesar disso, esta ordem não tem muita importância, desde que seja mantida a relação dos valores com os atributos.

FILME	TITULO	ANO	DIRETOR	ATOR_PRINC_1	ATOR_PRINC_2	COR	DURACAO	GENERO	CRITICA
	Dave	1993	Ivan Reitman	Kevin Kline	Null	Color	100	Comédia	****
	The Boat (Das Boat)	1981	Wolfang Petersen	Jurgen Prochbow	Herbert Gronemeyer	Color	Null	Guerra	*****
	The Day World Ended	1956	Roger Corman	Richard Denning	Null	B&W	82	Ficção	**
	David and Bathsheba	1951	Henry King	Gregory Peck	Susan Hayward	Color	116	Épico	**

Figura 3.2: Relação FILME com ordenação diferente das tuplas.

Observação: Cada valor em uma tupla é um valor atômico, ou seja, não é divisível. No Modelo Relacional tanto a composição de atributos quanto atributos multivalorados não são permitidos.

3.2 Atributos Chave de uma Relação

Por definição, uma relação (ver Definição 3.4) é um conjunto de tuplas e, também por definição, todos os elementos de um conjunto são distintos. Então, todas as tuplas de uma relação devem ser distintas, ou seja, não podem existir tuplas com o mesmo conjunto de valores para todos os atributos. Um conjunto ou subconjunto de atributos do esquema da relação que mantém esta propriedade das tuplas serem distintas em qualquer instância de relação r de R é chamado de superchave e é definido a seguir.

Definição 3.6: É chamado de *superchave* qualquer subconjunto de atributos do esquema da relação R que mantém a propriedade das tuplas de qualquer instância de relação r de R serem distintas. Considere o subconjunto SK do conjunto de atributos que define o esquema de relação R. SK é uma superchave se para quaisquer duas tuplas distintas t_1 e t_2 em qualquer instância de relação r de R, a seguinte restrição acontece:

$$t_1[SK] \neq t_2[SK]$$

Toda relação tem pelo menos uma superchave que é o conjunto de todos os atributos do esquema da relação.

Exemplo 3.5: Seja o esquema FILME apresentado no Exemplo 3.2. Considerando, hipoteticamente, que nunca poderiam existir dois ou mais filmes com o mesmo título, podemos dizer que, além do conjunto de todos os atributos do esquema FILME, outras superchaves do esquema da relação FILME seriam: {TITULO}, {TITULO, DURACAO} e {TITULO, DURACAO, GENERO}.

Definição 3.7: Uma *chave* do esquema da relação R é uma superchave mínima, ou seja, se retirarmos qualquer atributo da superchave, esta deixa de ser uma superchave. Um atributo chave é usado para identificar unicamente cada tupla da relação e isso deve ser mantido em cada instância da relação, mesmo com a inserção de novas tuplas na relação. Um esquema de relação pode ter mais de uma chave e neste caso cada uma das chaves é chamada de chave candidata. Normalmente escolhe-se uma das chaves candidatas para ser aquela cujos valores serão usados para identificar as tuplas na relação. A escolhida recebe o nome de *chave primária da relação*. É

convencionado sublinhar a chave primária da relação, como é mostrado no Exemplo 3.6 e na Figura 3.3.

Com a definição de chave da relação é possível definir formalmente as restrições de chave e de integridade de entidade, também pertencentes à Base de Dados Relacional.

Definição 3.8: A *restrição de chave* especifica que todas as chaves candidatas de cada relação devem ter valores únicos para cada tupla em qualquer instância do esquema desta relação.

Definição 3.9: A *restrição de integridade de entidade* especifica que a chave primária da relação não pode ter valor nulo, garantindo assim a unicidade das tuplas.

Veja por exemplo, o atributo RG do esquema da relação CLIENTE, apresentado no Exemplo 3.6. Como não é possível existir dois números de carteira de identidade iguais para pessoas diferentes, isso garante que cada tupla será única em relação a esse atributo. O mesmo vale para o atributo CPF, do mesmo esquema da relação CLIENTE, que é uma chave candidata. Considerando ainda o atributo RG, se o valor deste campo pudesse ser nulo e existissem dois ou mais clientes com este campo nulo não seria mais possível identificar unicamente estas tuplas com relação a este atributo.

Exemplo 3.6: Seja o esquema CLIENTE(NOME, <u>RG</u>, CPF, ENDERECO). Existem duas chaves candidatas para essa relação: RG e CPF, mas a chave candidata RG foi escolhida para ser a chave primária da relação CLIENTE.

A escolha do atributo chave deve levar em consideração o significado dos atributos participantes do esquema da relação. Não se devem usar atributos do tipo NOME, como na relação CLIENTE apresentada no Exemplo 3.6, pois não se pode garantir que nunca um nome irá se repetir.

O mesmo pode acontecer com a relação FILME. Pensando em uma base de dados de filmes do mundo real ao invés de uma hipotética, como vinha sendo usado até agora, não se pode garantir que o título de um filme seja único, portanto deve-se escolher uma outra chave para esta relação. Dentre os atributos existentes na relação, nenhum pode ser caracterizado como chave da

relação, portanto é adicionado um novo atributo, chamado CODIGO, que contém um valor único para cada tupla pertencente à relação, que caracteriza univocamente cada tupla da relação FILME. O domínio do atributo CODIGO é dom(CODIGO) = Codigos, no qual Codigos é o conjunto de todos os valores inteiros de 6 dígitos. O novo esquema da relação FILME, com o acréscimo do atributo e também chave primária CODIGO, é: FILME(CODIGO, TITULO, ANO, DIRETOR, ATOR_PRINC_1, ATOR_PRINC_2, COR, DURACAO, GENERO, CRITICA). A relação FILME, com o novo atributo, está representada na Figura 3.3.

FILME	CODIGO	TITULO	ANO	DIRETOR	ATOR_PRINC_1	ATOR_PRINC_2	COR	DURACAO	GENERO	CRITICA
	2	The Day World Ended	1956	Roger Corman	Richard Denning	Null	B&W	82	Ficção	**
	3	The Boat (Das Boat)	1981	Wolfang Petersen	Jurgen Prochbow	Herbert Gronemeyer	Color	Null	Guerra	*****
	4	David and Bathsheba	1951	Henry King	Gregory Peck	Susan Hayward	Color	116	Épico	**
	5	Dave	1993	Ivan Reitman	Kevin Kline	Null	Color	100	Comédia	****

Figura 3.3: Relação FILME com o atributo e chave primária CODIGO.

3.3 Esquema de Base de Dados Relacional e Restrições de Integridade

O que foi apresentado até agora focalizava uma única relação; uma base de dados relacional real, entretanto, possui, normalmente, muitas relações e cada uma destas possui muitas tuplas que se relacionam entre si de várias maneiras.

Definição 3.10: Um *esquema de base de dados relacional* S é um conjunto de esquemas de relações $S = \{R_1, R_2, ..., R_m\}$ e um conjunto de restrições de integridade IC.

Definição 3.11: Seja o esquema de base de dados relacional S e o conjunto de restrições de integridade IC. Uma *instância de base de dados relacional* DB de S é um conjunto de instâncias de relações $DB = \{r_1, r_2, ..., r_m\}$ tal que cada r_i é uma instância do esquema de relação R_i $(1 \leq i \leq n)$ que satisfaz as restrições de integridade especificadas em IC. Uma instância de uma base de dados representa o estado da base de dados em um dado momento.

Por meio da análise do esquema da base de dados LOCADORA, representada na Figura 3.4, pode-se observar que alguns atributos podem representar o mesmo conceito do mundo real, como o atributo RG e o atributo RG_CLIENTE dos esquemas das relações CLIENTE e

DEPENDENTE, respectivamente. Ambos se referem ao número da carteira de identidade de um determinado cliente. Eles também poderiam ter o mesmo nome, uma vez que definem relações diferentes. Outra observação é a de que também se pode ter dois atributos com o mesmo nome, desde que estejam em relações diferentes, sem que estes representem o mesmo conceito do mundo real. Por exemplo, considere o atributo CODIGO, presente no esquema da relação FILME e no esquema da relação LOCACAO. No primeiro esquema o atributo representa o valor que torna único cada filme enquanto que no segundo caso, representa o valor que torna única cada locação de filme efetuada.

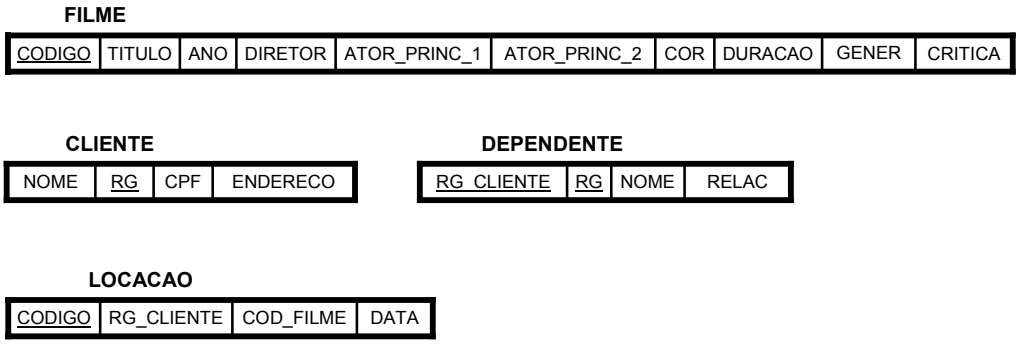


Figura 3.4: O esquema da base de dados relacional LOCADORA.

O relacionamento dos dados entre relações pode ser melhor entendido quando visualizado por meio de dados. Isto é mostrado na Figura 3.5, onde está representada uma instância da base de dados LOCADORA.

FILME

CODIGO	TITULO	ANO	DIRETOR	ATOR_PRINC_1	ATOR_PRINC_2	COR	DURACAO	GENERO	CRITICA
2	The Day World Ended	1956	Roger Corman	Richard Denning	Null	B&W	82	Ficção	**
3	The Boat (Das Boat)	1981	Wolfang Petersen	Jurgen Prochbow	Herbert Gronemeyer	Color	Null	Guerra	*****
4	David and Bathsheba	1951	Henry King	Gregory Peck	Susan Hayward	Color	116	Épico	**
5	Dave	1993	Ivan Reitman	Kevin Kline	Null	Color	100	Comédia	****

CLIENTE

NOME	RG	CPF	ENDERECO
José da Silva	111111111	222222222	Rua Barão do Rio Branco, 12
Maria Cristina de Abreu	333333333	444444444	Av. São Carlos, 34
João Carlos Rodrigues	555555555	666666666	Av. 9 de Julho, 56
Cláudia Martins	777777777	888888888	Rua Aprígio de Araújo, 78
Rosana Moreira	999999999	909090909	Rua Santa Úrsula, 90
Luzia Santana	222222222	111111111	Rua Florência de Abreu, 21
Carlos Alberto Silveira	444444444	333333333	Rua Antônio Pedroso, 43
Arlindo Miranda	666666666	555555555	Rua Alfredo Scaranello, 65
Daniel Toniello	888888888	777777777	Rua Sebastião Sampaio, 87
Maria Cláudia Miranda	909090909	999999999	Rua Alfredo Scaranello, 9

LOCACAO

CODIGO	RG_CLIENTE	COD_FILME	DATA
3	222222222	4	10/10/2003
4	444444444	3	09/11/2003
5	888888888	2	09/11/2003
6	222222222	2	12/11/2003
7	666666666	5	12/11/2003
8	111111111	4	12/11/2003
9	999999999	4	13/11/2003
10	333333333	5	15/11/2003

DEPENDENTE

RG_CLIENTE	RG	NOME	RELAC
555555555	515151515	José Rodrigues	Filho
555555555	151515151	Sheila Rodrigues	Esposa
222222222	212121212	Rafaela Santana	Neta
777777777	717171717	Célia Martins	Filha

Figura 3.5: Uma instância da base de dados relacional LOCADORA.

Definição 3.12: Um conjunto de atributos FK de um esquema de relação R_1 é uma chave estrangeira de R_1 se satisfaz as duas regras:

1. Os atributos em FK têm o mesmo domínio que os atributos da chave primária PK de um outro esquema de relação R_2. Os atributos FK são referências ou referem-se à relação R_2.
2. O valor de FK em uma tupla t_1 de R_1 ocorre como valor de PK em alguma tupla t_2 de R_2 ou é nulo. No primeiro caso tem-se que $t_1[FK] = t_2[PK]$ e diz-se que t_1 referencia ou refere-se à t_2.

Exemplo 3.7: Considere o esquema da base de dados LOCADORA apresentada na Figura 3.4. O atributo RG_CLIENTE do esquema da relação DEPENDENTE é chave estrangeira da relação e faz referência à chave primária RG do esquema da relação CLIENTE. Outros exemplos de chave

estrangeira presentes no esquema da base de dados LOCADORA são os atributos RG_CLIENTE e COD_FILME do esquema de relação LOCACAO que referenciam, respectivamente, a chave primária RG do esquema de relação CLIENTE e a chave primária CODIGO do esquema de relação FILME. Estas referências das chaves estrangeiras para com as chaves primárias podem ser visualizadas por meio da instância da base de dados LOCADORA, apresentada na Figura 3.5.

Com a definição do conceito de chave estrangeira é possível apresentar o conceito de restrição de integridade referencial que, diferentemente das outras restrições que fazem referência a uma única relação, é especificada entre duas relações e é definida a seguir.

Definição 3.13: A *restrição de integridade referencial* determina que uma tupla que possuir valor em sua chave estrangeira deve referenciar obrigatoriamente uma outra tupla existente. Isto serve para manter a consistência entre as tuplas, evitando que uma tupla referencie uma outra inexistente.

3.4 Operações de Atualização em Relações

Existem três operadores básicos que atualizam as relações: *inserção(INSERT)*, *eliminação(DELETE)* e *alteração(MODIFY)*. Quando uma relação sofre alguma alteração, por meio dos operadores de atualização, é necessário verificar se as restrições de chave e integridade não foram violadas. A seguir são definidos os três operadores de atualização de uma relação juntamente com as respectivas verificações das restrições necessárias para cada operador.

Definição 3.14: O operador *INSERT*[3] é utilizado para adicionar novas tuplas a uma relação. Para este operador é necessário verificar as restrições de domínio e de integridade. No caso de uma violação a inserção pode ser rejeitada ou ser exigida que a causa da violação seja corrigida antes da inserção ser efetivada.

[3] Os nomes de operadores são mantidos em inglês para facilitar referência aos operadores originais.

Exemplo 3.8: Seja a instância da base de dados LOCADORA apresentada na Figura 3.5 e as seguintes operações:

- Inserir < 'João Marcos Vieira', '262626262', '686868686', 'Rua Antônio Pedroso, 268' > na relação CLIENTE.

  - Neste caso a adição da nova tupla não viola nenhuma das restrições.

- Inserir < '5', '222222222', '3', '14/11/2003' > na relação LOCACAO.

  - Neste caso a adição da nova tupla viola a restrição de integridade de chave, pois já existe na relação LOCACAO uma tupla com o mesmo valor para a chave primária CODIGO. A solução seria rejeitar a inserção ou corrigir o valor do atributo CODIGO da nova tupla.

- Inserir < 'João Marcos Vieira', *Null*, '686868686', 'Rua Antônio Pedroso, 268' > na relação CLIENTE.

  - Neste caso a adição da nova tupla viola a restrição de integridade de entidade, pois o valor da chave primária RG da relação CLIENTE é nulo. A solução seria rejeitar a inserção ou colocar um valor para a chave primária RG.

- Inserir < '686868686', '222222222', 'João Marcos Vieira', 'Filho' > na relação DEPENDENTE.

  - Neste caso a adição da nova tupla viola a restrição de integridade referencial, pois não existe na relação CLIENTE uma tupla com o valor indicado para a chave estrangeira RG_CLIENTE da relação DEPENDENTE. A solução seria rejeitar a inserção ou inserir, antes de completar a primeira inserção, uma nova tupla na relação CLIENTE que contenha o valor indicado.

Definição 3.15: O operador *DELETE* é utilizado para remover tuplas de uma relação. Para este operador a única restrição de integridade a ser verificada é a de integridade referencial, pois é a única que pode ser violada por meio da remoção de tuplas de uma relação. No caso de ocorrer violação, três soluções podem ser utilizadas: rejeitar a remoção; disparar uma remoção em cascata, ou seja, remover também todas as tuplas que fazem referência à tupla a ser removida, o que nem sempre é possível ou indicado; modificar o valor da referência para outro valor válido ou para o valor nulo, quando possível. Quando a chave estrangeira da tupla a ser modificada, por conta da remoção de outra tupla, fizer parte da chave primária, não será possível colocar o valor nulo já que isso violaria a restrição de integridade de entidade.

Exemplo 3.9: Seja a instância da base de dados LOCADORA apresentada na Figura 3.5 e as seguintes operações:

- Eliminar da relação CLIENTE a tupla com RG = '555555555' e CPF = '666666666'.
  - Neste caso a remoção da tupla não viola a restrição de integridade referencial.
- Eliminar da relação CLIENTE a tupla com RG = '222222222'.
  - Neste caso a remoção da tupla viola a restrição de integridade referencial, pois existem tuplas nas relações DEPENDENTE e LOCACAO referenciando valor da chave primária RG da tupla a ser removida. Qualquer uma das três soluções seria possível, desde que não violem nenhuma restrição de integridade e respeitem o real significado dos dados para que estes não se tornem inconsistentes. Não seria aconselhável, por exemplo, alterar o valor do atributo RG_CLIENTE para outro valor válido, pois as locações feitas por esse cliente seriam transferidas para outro cliente.

Definição 3.16: O operador *MODIFY* é utilizado para modificar valores de atributos em uma tupla (ou tuplas) de alguma instância de relação R. Para este operador também é necessário fazer verificações, assim como nos outros operadores. Quando os atributos a serem alterados não são chaves primárias ou chaves estrangeiras, normalmente não há problema, bastando verificar se não houve violação da restrição de domínio. Quando o atributo a ser alterado é uma chave estrangeira é necessário verificar a restrição de integridade referencial. Já quando o atributo a ser alterado é uma chave primária, é como se a tupla fosse removida e depois inserida novamente com um novo

valor para a chave. Então, tanto as verificações feitas para os respectivos operadores, *DELETE* e *INSERT*, quanto as soluções para um caso de violação são as mesmas para este operador.

Exemplo 3.10: Seja a instância da base de dados LOCADORA apresentada na Figura 3.5 e as seguintes operações:

- Modificar o ANO da relação FILME na tupla com CODIGO = '2' para '1965'.
  - Neste caso a modificação da tupla não viola nenhuma das restrições de integridade.
- Modificar o RG_CLIENTE da relação LOCACAO na tupla com CODIGO = '4' para '434343434'.
  - Neste caso a modificação da tupla viola a restrição de integridade referencial, pois não existe na relação CLIENTE uma tupla com o valor indicado para atributo RG_CLIENTE.
- Modificar o RG da relação CLIENTE na tupla com RG = '222222222' para '333333333'.
  - Neste caso a modificação da tupla viola a restrição de integridade chave, pois já existe na relação CLIENTE uma tupla com o valor indicado para atributo RG.

3.5 A Álgebra Relacional

A Álgebra Relacional é uma coleção de operações que manipulam as relações de uma base de dados. Tais operações podem ser usadas para selecionar tuplas de uma única relação ou combinar tuplas relacionadas de várias relações diferentes. O resultado de qualquer operação da Álgebra Relacional sempre é uma nova relação, que por sua vez também pode ser manipulada por tais operações.

Na seqüência, todas as operações da Álgebra Relacional são definidas e, para melhor entendimento, são também exemplificadas, sempre utilizando a instância da base de dados LOCADORA, representada pela Figura 3.5, para formular os exemplos.

Definição 3.17: O operador *seleção(SELECT)*, denotado pelo símbolo σ, seleciona um subconjunto de tuplas de uma relação que satisfazem uma determinada condição de seleção. A sintaxe da operação é:

$$\sigma_{<\text{condição}>}(<\text{relação}>),$$

onde <condição> é a condição da seleção e <relação> é o nome da relação. A condição da seleção é uma expressão booleana, formada com os atributos da relação especificada, na forma de cláusulas do tipo:

<atributo> <operador> <atributo> ou <atributo> <operador> <constante>,

onde <atributo> é o nome do atributo a ser comparado, <operador> é o operador de comparação, normalmente um dos operadores {=, <, ≤, >, ≥, ≠} e <constante> é um valor constante.
Uma expressão de condição pode ter várias cláusulas conectadas pelos operadores booleanos AND, OR ou NOT. A ordem em que as cláusulas são informadas não importa.

O operador *SELECT* é unário, comutativo e permite o encadeamento de operações de seleção em uma única operação usando o conectivo AND. O resultado das operações de seleção é sempre uma relação de mesmo grau e com os mesmos atributos da relação especificada.

O operador *SELECT* é comutativo ou seja,

$$\sigma_{<cond1>}(\sigma_{<cond2>}(R)) = \sigma_{<cond2>}(\sigma_{<cond1>}(R))$$

e, portanto, a seqüência de operadores *SELECT* pode ser aplicada em qualquer ordem. Além disso, uma aplicação em seqüência de operadores *SELECT* pode ser "traduzida" como uma única aplicação do operador *SELECT* onde a condição é expressa como uma conjunção ou seja:

$$\sigma_{<cond1>}(\sigma_{<cond2>}(...(\sigma_{<condn>}(R))...)) = \sigma_{<cond1> \text{ AND } <cond2> \text{ AND } \cdots \text{ AND } <condn>}(R)$$

Exemplo 3.11: Seja a instância da base de dados LOCADORA representada na Figura 3.5 e as operações de seleção a seguir:

- $\sigma_{ANO > 1970 \text{ AND } COR = \text{'Color'}}(FILME)$

  - A operação seleciona todas as tuplas da relação FILME em que o valor do atributo ANO é maior que 1970 e o valor do atributo COR igual a 'Color' (ver Figura 3.6(a)).

- $\sigma_{(ANO > 1970 \text{ AND } COR = \text{'Color'}) \text{ OR } (GENERO = \text{'Ficção'})}(FILME)$

  - A operação seleciona tanto as tuplas da relação FILME em que o valor do atributo ANO é maior que 1970 e o valor do atributo COR igual a 'Color' quanto aquelas tuplas em que o valor do atributo GENERO é igual a 'Ficção' (ver Figura 3.6(b)).

- $\sigma_{DATA \geq \text{'09/11/2003'} \text{ AND } DATA \leq \text{'12/11/2003'}}(LOCACAO)$

- A operação seleciona todas as tuplas da relação LOCACAO em que o valor do atributo DATA é maior ou igual a '09/11/2003' e menor ou igual a '12/11/2003' (ver Figura 3.6(c)).

a)

CODIGO	TITULO	ANO	DIRETOR	ATOR_PRINC_1	ATOR_PRINC_2	COR	DURACAO	GENERO	CRITICA
3	The Boat (Das Boat)	1981	Wolfang Petersen	Jurgen Prochbow	Herbert Gronemeyer	Color	Null	Guerra	*****
5	Dave	1993	Ivan Reitman	Kevin Kline	Null	Color	100	Comédia	****

b)

CODIGO	TITULO	ANO	DIRETOR	ATOR_PRINC_1	ATOR_PRINC_2	COR	DURACAO	GENERO	CRITICA
2	The Day World Ended	1956	Roger Corman	Richard Denning	Null	B&W	82	Ficção	**
3	The Boat (Das Boat)	1981	Wolfang Petersen	Jurgen Prochbow	Herbert Gronemeyer	Color	Null	Guerra	*****
5	Dave	1993	Ivan Reitman	Kevin Kline	Null	Color	100	Comédia	****

c)

CODIGO	RG_CLIENTE	COD_FILME	DATA
4	444444444	3	09/11/2003
5	888888888	2	09/11/2003
6	222222222	2	12/11/2003
7	666666666	5	12/11/2003
8	111111111	4	12/11/2003

Figura 3.6: O resultado das operações do Exemplo 3.11:
a) $\sigma_{ANO > 1970 \text{ AND } COR = \text{'Color'}}(FILME)$;
b) $\sigma_{(ANO > 1970 \text{ AND } COR = \text{'Color'}) \text{ OR } (GENERO = \text{'Ficção'})}(FILME)$;
c) $\sigma_{DATA \geq \text{'09/11/2003'} \text{ AND } DATA \leq \text{'12/11/2003'}}(LOCACAO)$.

Definição 3.18: O operador *projeção(PROJECT)*, denotado pelo símbolo π, retorna as tuplas de uma relação projetadas sobre um subconjunto dos atributos da relação origem, eliminando os demais. Caso ocorra duplicidade de tuplas, devido à projeção, estas são eliminadas. A sintaxe da operação é:

$$\pi_{<atributos>}(<relação>)$$

onde <atributos> é a lista de atributos sobre os quais a relação origem será projetada e <relação> é o nome da relação origem. Ao contrário da operação de seleção, a operação de projeção não é comutativa; o resultado desta operação, entretanto é também uma relação, contendo tuplas distintas usando apenas os atributos selecionados.

Exemplo 3.12: Considere a base de dados LOCADORA e as operações de projeção a seguir:

- $\pi_{TITULO, DURACAO, GENERO, CRITICA}(FILME)$

- A operação projeta a relação FILME sobre os atributos TITULO, DURACAO, GENERO e CRITICA (ver Figura 3.7(a)).

- $\pi_{NOME, ENDERECO}(CLIENTE)$

  - A operação projeta a relação CLIENTE sobre os atributos NOME e ENDERECO (ver Figura 3.7(b)).

a)

TITULO	DURACAO	GENERO	CRITICA
The Day World Ended	82	Ficção	**
The Boat (Das Boat)	Null	Guerra	*****
David and Bathsheba	116	Épico	**
Dave	100	Comédia	****

b)

NOME	ENDERECO
José da Silva	Rua Barão do Rio Branco, 12
Maria Cristina de Abreu	Av. São Carlos, 34
João Carlos Rodrigues	Av. 9 de Julho, 56
Cláudia Martins	Rua Aprígio de Araújo, 78
Rosana Moreira	Rua Santa Úrsula, 90
Luzia Santana	Rua Florência de Abreu, 21
Carlos Alberto Silveira	Rua Antônio Pedroso, 43
Arlindo Miranda	Rua Alfredo Scaranello, 65
Daniel Toniello	Rua Sebastião Sampaio, 87
Maria Cláudia Miranda	Rua Alfredo Scaranello, 9

Figura 3.7: O resultado das operações do Exemplo 3.12:
a) $\pi_{TITULO, DURACAO, GENERO, CRITICA}(FILME)$;
b) $\pi_{NOME, ENDERECO}(CLIENTE)$.

É possível realizar várias operações da Álgebra Relacional em seqüência e isso é feito de duas maneiras: aninhando todas as operações em uma única expressão algébrica relacional ou executando as operações uma por vez, criando resultados intermediários. Neste último caso as relações que mantém os resultados intermediários devem ter nome o que não é necessário nas operações vistas nos exemplos anteriores ou nas operações aninhadas.

Exemplo 3.13: Considere a base de dados LOCADORA e a operação $\pi_{RG\_CLIENTE, COD\_FILME}(\sigma_{DATA \geq '09/11/2003' \text{ AND } DATA \leq '12/11/2003'}(LOCACAO))$. Esta operação, primeiramente, seleciona todas as tuplas da relação LOCACAO nas quais o valor do atributo DATA é maior ou igual a '09/11/2003' e menor ou igual a '12/11/2003'. Depois, com a relação resultado desta operação, obtém outra relação que é o resultado da projeção sobre os atributos RG_CLIENTE e COD_FILME. O resultado final da operação está representado na Figura 3.8(a).

Exemplo 3.14: Considere a base de dados LOCADORA e a seqüência de operações:

$$\text{TEMP\_LOC} \leftarrow \sigma_{\text{DATA} \geq \text{'09/11/2003'} \text{ AND DATA} \leq \text{'12/11/2003'}}(\text{LOCACAO})$$

$$\text{RESULT} \leftarrow \pi_{\text{RG\_CLIENTE, COD\_FILME}}(\text{TEMP\_LOC})$$

A primeira operação seleciona todas as tuplas da relação LOCACAO nas quais o valor do atributo DATA é maior ou igual a '09/11/2003' e menor ou igual a '12/11/2003' resultando na relação intermediária TEMP_LOC. A segunda operação projeta a relação intermediária sobre os atributos RG_CLIENTE e COD_FILME. O resultado final da operação está representado na Figura 3.8(b).

Quando se usa a técnica de estabelecer uma seqüência de operações com relações intermediárias é possível renomear os atributos da relação resultado. Isto pode ser muito útil em operações complexas nas quais se deseja alterar o nome de algum atributo para facilitar o entendimento do seu significado na relação.

Exemplo 3.15: Considere a base de dados LOCADORA e a seqüência de operações do Exemplo 3.14 com uma modificação para renomear o atributo COD_FILME:

$$\text{TEMP\_LOC} \leftarrow \sigma_{\text{DATA} \geq \text{'09/11/2003'} \text{ AND DATA} \leq \text{'12/11/2003'}}(\text{LOCACAO})$$

$$\text{RESULT(RG\_CLIENTE, CODIGO\_FILME)} \leftarrow \pi_{\text{RG\_CLIENTE, COD\_FILME}}(\text{TEMP\_LOC})$$

A operação é exatamente a mesma realizada no Exemplo 3.14 com a diferença de renomeação do atributo COD_FILME para CODIGO_FILME no resultado final da operação, que está representado na Figura 3.8(c).

a)

RG_CLIENTE	COD_FILME
444444444	3
888888888	2
222222222	2
666666666	5
111111111	4

b) TEMP_LOC

CODIGO	RG_CLIENTE	COD_FILME	DATA
3	222222222	4	10/10/2003
4	444444444	3	09/11/2003
5	888888888	2	09/11/2003
6	222222222	2	12/11/2003
7	666666666	5	12/11/2003
8	111111111	4	12/11/2003
9	999999999	4	13/11/2003
10	333333333	5	15/11/2003

RESULT

RG_CLIENTE	COD_FILME
444444444	3
888888888	2
222222222	2
666666666	5
111111111	4

c) RESULT

RG_CLIENTE	CODIGO_FILME
444444444	3
888888888	2
222222222	2
666666666	5
111111111	4

Figura 3.8: O resultado das operações de Exemplo 3.13, Exemplo 3.14 e Exemplo 3.15, respectivamente:
a) $\pi_{RG\_CLIENTE,\ COD\_FILME}(\sigma_{DATA \geq \text{'09/11/2003'} \text{ AND } DATA \leq \text{'12/11/2003'}}(LOCACAO))$;
b) a mesma operação de a), mas utilizando relações intermediárias;
c) idem b), mas renomeando o atributo COD_FILME para CODIGO_FILME.

As operações que envolvem conjuntos da matemática clássica também são aplicáveis a relações de uma Base de Dados Relacional, pois uma relação (ver Definição 3.4) é um conjunto de tuplas. Antes da definição de tais operadores, entretanto, é introduzido o conceito de compatibilidade na união, importante para a definição destes operadores.

Definição 3.19: Sejam as relações $R(A_1, A_2, ..., A_n)$ e $S(B_1, B_2, ..., B_n)$. R e S são ditas *união compatíveis* se elas têm o mesmo grau e $dom(A_i) = dom(B_i)$ para $1 \leq i \leq n$. Ou seja, têm a mesma quantidade de atributos e cada par de atributos correspondentes tem o mesmo domínio.

Definição 3.20: Sejam as relações R e S. O resultado da operação $R \cup S$, chamada *União(UNION)* e denotada por $\cup$, é uma relação contendo todas as tuplas de R e S, sendo que todas as duplicidades são eliminadas.

Definição 3.21: Sejam as relações R e S. O resultado da operação R $\cap$ S, chamada *Intersecção(INTERSECTION)* e denotada por $\cap$, é uma relação contendo todas as tuplas que pertencem a R e que também pertencem a S.

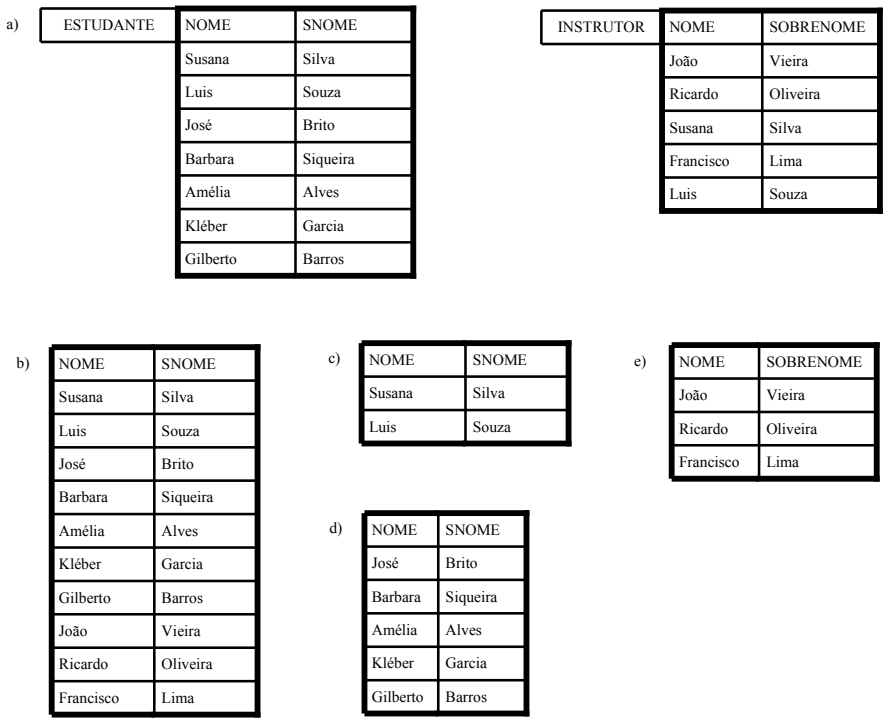
Definição 3.22: Sejam as relações R e S. O resultado da operação R – S, chamada *Diferença(DIFFERENCE)* e denotada por –, é uma relação contendo todas as tuplas que pertencem a R, mas que não pertencem a S.

É importante observar que as operações *UNION* e *INTERSECTION* são comutativas, o que não acontece com a operação *DIFFERENCE*, porém as três operações requerem que as relações envolvidas na operação sejam união compatíveis (ver Definição 3.19).

Exemplo 3.16: Sejam os esquemas das relações ESTUDANTE(NOME, SNOME) e INSTRUTOR(NOME, SOBRENOME), as quais são união compatíveis. Sejam também as instruções:

- ESTUDANTE $\cup$ INSTRUTOR
- ESTUDANTE $\cap$ INSTRUTOR
- ESTUDANTE – INSTRUTOR
- INSTRUTOR – ESTUDANTE

As relações e o resultado dessas operações estão representados na Figura 3.9.



a)

ESTUDANTE	NOME	SNOME
	Susana	Silva
	Luis	Souza
	José	Brito
	Barbara	Siqueira
	Amélia	Alves
	Kléber	Garcia
	Gilberto	Barros

INSTRUTOR	NOME	SOBRENOME
	João	Vieira
	Ricardo	Oliveira
	Susana	Silva
	Francisco	Lima
	Luis	Souza

b)

NOME	SNOME
Susana	Silva
Luis	Souza
José	Brito
Barbara	Siqueira
Amélia	Alves
Kléber	Garcia
Gilberto	Barros
João	Vieira
Ricardo	Oliveira
Francisco	Lima

c)

NOME	SNOME
Susana	Silva
Luis	Souza

d)

NOME	SNOME
José	Brito
Barbara	Siqueira
Amélia	Alves
Kléber	Garcia
Gilberto	Barros

e)

NOME	SOBRENOME
João	Vieira
Ricardo	Oliveira
Francisco	Lima

Figura 3.9: Representação do Exemplo 3.16: a) relações união compatíveis; b) ESTUDANTE ∪ INSTRUTOR; c) ESTUDANTE ∩ INSTRUTOR; d) ESTUDANTE – INSTRUTOR; e) INSTRUTOR – ESTUDANTE.

Definição 3.23: A operação *Junção(JOIN)*, denotada por ⋈, é usada para relacionar tuplas de duas relações, que satisfazem à condição de junção, em uma única tupla, processando o relacionamento entre as relações. A sintaxe da operação é:

$$<\text{relação1}> \bowtie_{<\text{condição}>} <\text{relação2}>,$$

onde <condição> é a condição da junção e <relação1> e <relação2> são os nomes das relações.

A condição da junção é uma expressão formada por cláusulas do tipo: <atributo_r1> <operador> <atributo_r2>, onde <atributo_r1> é o nome do atributo de <relação1> a ser comparado, <operador> é o operador de comparação, normalmente um dos operadores {=, <, ≤, >, ≥, ≠} e <atributo_r2> é o nome do atributo de <relação2> a ser comparado. Os atributos <atributo_r1> e <atributo_r2>, chamados de atributos da junção, devem ter o mesmo domínio. Uma condição de junção pode ter várias cláusulas; quando esse for o caso, elas devem ser conectadas pelo operador booleano AND. A operação *JOIN* na sua forma geral é chamada de THETA JOIN e quando as comparações são todas do tipo igualdade a operação é chamada de EQUI JOIN. Considerando

R1(A_1, A_2, ..., A_n) e R2(B_1, B_2, ..., B_m) os esquemas das relações <relação1> e <relação2> respectivamente, o resultado da operação *JOIN* será a relação Q com n + m atributos Q(A_1, A_2, ..., A_n, B_1, B_2, ..., B_m) contendo uma tupla para cada combinação de tuplas, uma da <relação1> e uma da <relação2>, que satisfaçam à condição da junção.

Na operação EQUI JOIN os atributos comparados têm valores idênticos e por isso a relação resultado acaba tendo atributos duplicados, requerendo assim uma operação de projeção para eliminá-los. Por essa razão foi criada uma outra operação, chamada de *Junção Natural (NATURAL JOIN)* definida a seguir.

Definição 3.24: A operação *NATURAL JOIN*, denotada por $*$, é usada como a operação *JOIN*, mas automaticamente elimina os atributos duplicados mantendo apenas o atributo da junção da primeira relação. Como nesta operação as comparações são sempre de igualdade, o operador pode ser omitido colocando-se apenas a lista dos atributos da junção. A sintaxe então seria: <relação1> $*_{(<lista1>),(<lista2>)}$ <relação2>, onde <relação1> e <relação2> são os nomes das relações e <lista1> e <lista2> são as listas de atributos da junção das relações <relação1> e <relação2>, respectivamente. Os atributos da junção, assim como na operação *JOIN*, devem ter o mesmo domínio.

Exemplo 3.17: Sejam as relações FILME, CLIENTE e LOCACAO representadas na Figura 3.5 e suponha que se deseja retornar o nome dos clientes e os nomes e data de locação dos filmes locados por cada cliente. Esse resultado é obtido por meio da seqüência de operações a seguir:

- TEMP_FILME(CODIGOF, TITULO) $\leftarrow \pi_{CODIGO,\ TITULO}$(FILME)

  - Projeta na relação intermediária TEMP_FILME as tuplas da relação FILME sobre os atributos CODIGO e TITULO, renomeando o atributo CODIGO para CODIGOF, a fim de evitar confusão com outro campo chamado CODIGO de outra relação após a junção (ver Figura 3.10(a)).

- TEMP_R1 $\leftarrow$ LOCACAO $\bowtie_{COD\_FILME\ =\ CODIGOF}$ TEMP_FILME

  - Realiza a operação *JOIN* ou a *EQUI JOIN*, relacionando as tuplas da relação LOCACAO com as tuplas da relação intermediária TEMP_FILME por meio dos atributos

COD_FILME e CODIGOF. O resultado é a relação intermediária TEMP_R1 (ver Figura 3.10(b)).

- R1(RG_CLIENTE, COD_FILME, TITULO, DATA_LOC) $\leftarrow \pi_{\text{RG\_CLIENTE, COD\_FILME, TITULO, DATA}}$ (TEMP_R1)

  - Projeta na relação intermediária R1 as tuplas da relação intermediária TEMP_R1 sobre os atributos RG_CLIENTE, COD_FILME, TITULO e DATA, renomeando o atributo DATA para DATA_LOC, a fim de deixar claro que o campo se referencia à data da locação de um filme, ou seja, ajudar na interpretação do significado do campo na relação (ver Figura 3.10(c)).

- TEMP_CLI $\leftarrow \pi_{\text{NOME, RG}}$(CLIENTE)

  - Projeta na relação intermediária TEMP_CLI as tuplas da relação CLIENTE sobre os atributos NOME e RG (ver Figura 3.10(d)).

- R $\leftarrow$ TEMP_CLI $*_{\text{(RG),(RG\_CLIENTE)}}$ R1

  - Realiza a operação *NATURAL JOIN* relacionando as tuplas da relação intermediária TEMP_CLI com as tuplas da relação intermediária R1 por meio dos atributos RG e RG_CLIENTE onde o atributo RG_CLIENTE é eliminado. O resultado é a relação R (ver Figura 3.10(e)).

a) TEMP_FILME

CODIGOF	TITULO
2	The Day World Ended
3	The Boat (Das Boat)
4	David and Bathsheba
5	Dave

b) TEMP_R1

CODIGO	RG_CLIENTE	COD_FILME	DATA	CODIGOF	TITULO
3	222222222	4	10/10/2003	4	David and Bathsheba
4	444444444	3	09/11/2003	3	The Boat (Das Boat)
5	888888888	2	09/11/2003	2	The Day World Ended
6	222222222	2	12/11/2003	2	The Day World Ended
7	666666666	5	12/11/2003	5	Dave
8	111111111	4	12/11/2003	4	David and Bathsheba
9	999999999	4	13/11/2003	4	David and Bathsheba
10	333333333	5	15/11/2003	5	Dave

c) R1

RG_CLIENTE	COD_FILME	TITULO	DATA_LOC
222222222	4	David and Bathsheba	10/10/2003
444444444	3	The Boat (Das Boat)	09/11/2003
888888888	2	The Day World Ended	09/11/2003
222222222	2	The Day World Ended	12/11/2003
666666666	5	Dave	12/11/2003
111111111	4	David and Bathsheba	12/11/2003
999999999	4	David and Bathsheba	13/11/2003
333333333	5	Dave	15/11/2003

d) TEMP_CLI

NOME	RG
José da Silva	111111111
Maria Cristina de Abreu	333333333
João Carlos Rodrigues	555555555
Cláudia Martins	777777777
Rosana Moreira	999999999
Luzia Santana	222222222
Carlos Alberto Silveira	444444444
Arlindo Miranda	666666666
Daniel Toniello	888888888
Maria Cláudia Miranda	909090909

e) R

NOME	RG	COD_FILME	TITULO	DATA_LOC
José da Silva	111111111	4	David and Bathsheba	12/11/2003
Maria Cristina de Abreu	333333333	5	Dave	15/11/2003
Rosana Moreira	999999999	4	David and Bathsheba	13/11/2003
Luzia Santana	222222222	4	David and Bathsheba	10/10/2003
Luzia Santana	222222222	2	The Day World Ended	12/11/2003
Carlos Alberto Silveira	444444444	3	The Boat (Das Boat)	09/11/2003
Arlindo Miranda	666666666	5	Dave	12/11/2003
Daniel Toniello	888888888	2	The Day World Ended	09/11/2003

Figura 3.10: Resultado das operações feitas no Exemplo 3.17:
a) TEMP_FILME(CODIGOF, TITULO) $\leftarrow \pi_{\text{CODIGO, TITULO}}$(FILME);
b) TEMP_R1 $\leftarrow$ LOCACAO $\bowtie_{\text{COD\_FILME = CODIGOF}}$ TEMP_FILME;
c) R1(RG_CLIENTE, COD_FILME, TITULO, DATA_LOC) $\leftarrow \pi_{\text{RG\_CLIENTE, COD\_FILME, TITULO, DATA}}$ (TEMP_R1);
d) TEMP_CLI $\leftarrow \pi_{\text{NOME, RG}}$(CLIENTE);
e) R $\leftarrow$ TEMP_CLI $*_{\text{(RG),(RG\_CLIENTE)}}$ R1.

3.6 Considerações Finais

Este capítulo focalizou a apresentação do Modelo Relacional, mostrando seus conceitos e definições formais, e da Álgebra Relacional, definindo e exemplificando seus principais

operadores. O próximo capítulo apresenta e discute o Modelo Relacional Aproximado e seus conceitos, tendo como base as abordagens teóricas vistas nos capítulos anteriores e visando refinar e padronizar a notação utilizada pelos autores do modelo.

CAPÍTULO 4. MODELO RELACIONAL APROXIMADO

Atualmente, em termos mundiais, o volume de dados armazenado é gigantesco e continua crescendo rapidamente. Segundo pesquisa publicada em [Fortes 2003], somente nas empresas, esse número vem crescendo 34% ao ano no Brasil e 37% no mundo. Infelizmente, devido à incapacidade do ser humano de interpretar tamanha quantidade de dados, muita informação e conhecimento, possivelmente úteis, podem estar sendo desperdiçados, ficando ocultos dentro das bases de dados espalhadas pelo mundo. Em conseqüência disso, a necessidade de se desenvolver novas ferramentas e técnicas de extração de conhecimento a partir de dados armazenados, também vem crescendo e se mostrando cada vez mais indispensável.

O objetivo deste capítulo é apresentar uma extensão do Modelo Relacional e da Álgebra Relacional (ver CAPÍTULO 3) utilizando os conceitos da TCA (ver CAPÍTULO 2) para posteriormente, ainda neste trabalho de pesquisa, combiná-lo a um método simbólico, investigando a contribuição dessa combinação para o desenvolvimento de novas ferramentas de extração de conhecimento.

4.1 Considerações Sobre o Modelo Relacional Aproximado

No mundo real não faltam situações em que a incerteza esteja presente, seja ela causada por uma informação incompleta, mal transmitida ou sujeita a mais de uma interpretação. Se essa informação precisa ser armazenada numa base de dados, podem surgir problemas em sua futura utilização como, por exemplo, no caso de se aplicar uma ferramenta de extração de conhecimento sobre essas informações incertas. Os resultados obtidos podem ser distorcidos, comprometendo o conhecimento extraído e os padrões encontrados que serão, possivelmente, falsos ou incertos.

Devido a esses problemas existe uma grande necessidade de que sejam desenvolvidas ferramentas capazes de interpretar e extrair informações com diferentes graus de granularidade. A TCA, conforme apresentado no CAPÍTULO 2, possui a habilidade de tratar a indiscernibilidade e, por meio da incorporação dessas suas características no Modelo Relacional, foi desenvolvido um novo modelo, apresentado a seguir, que pode viabilizar o desenvolvimento de tais ferramentas.

O Modelo Relacional Aproximado, proposto por Beauboeuf e Petry [Beauboeuf e Petry 1994] [Beauboeuf 2004] é uma extensão do Modelo Relacional que incorpora os conceitos básicos da TCA, tipicamente a relação de indiscernibilidade entre elementos e as aproximações inferior e superior, que são baseadas nessa relação (ver Definição 2.4).

Numa aplicação que implementa o Modelo Relacional Aproximado uma consulta retorna uma relação aproximada baseada na indiscernibilidade de valores de atributo. Uma relação aproximada é composta por dois conjuntos de tuplas, a saber: a aproximação inferior e a aproximação superior. Os elementos da aproximação inferior são respostas que, *com certeza*, pertencem à relação. Os elementos da aproximação superior são respostas que, *possivelmente*, pertencem à relação.

O principal objetivo do Modelo Relacional Aproximado é dar maior poder de recuperação de informações ao Modelo Relacional. Isso traz, como conseqüência, uma maior flexibilidade à linguagem SQL, dado que o mecanismo de consulta usa classes de equivalência ao invés da igualdade de valores, nas recuperações de informação.

Numa aplicação que implementa o Modelo Relacional o retorno é composto por aqueles elementos que certamente atendem ao que foi solicitado pelo consultor, enquanto que numa aplicação do Modelo Relacional Aproximado são também adicionados elementos que possivelmente atendem ao que foi solicitado pelo consultor.

4.2 Conceitos do Modelo Relacional Aproximado

Alguns conceitos do Modelo Relacional, aplicados ao Modelo Relacional Aproximado tiveram seus nomes alterados por fazerem parte do novo modelo e, quando nada em contrário for mencionado, terão o mesmo significado do modelo original.

Ambos os modelos representam os dados como uma coleção de relações (tabelas) contendo tuplas. Essas relações são conjuntos. As tuplas de uma relação são seus elementos e, como elementos de um conjunto, não comparecem duplicados e tampouco são ordenados.

Uma tupla t_i de uma Base de Dados Relacional Aproximada tem a forma $<d_{i1}, d_{i2}, ..., d_{in}>$, onde d_{ij} é um valor de domínio de um determinado domínio $dom(A_j)$ e A_j $(1 \leq j \leq n)$ é um atributo que nomeia um papel desempenhado pelo domínio D_j. Se, por um lado, em uma Base de Dados Relacional, $d_{ij} \in dom(A_j)$, em uma Base de Dados Relacional Aproximada, $d_{ij} \subseteq dom(A_j)$.

Ou seja, uma tupla, no Modelo Relacional Aproximado, pode ter, como componentes, conjuntos de valores de um domínio. Seja $P(dom(A_j))$ o conjunto potência[4] de $dom(A_j) - \emptyset$.

Definição 4.1: Uma *relação aproximada* R é um subconjunto de $P(dom(A_1)) \times P(dom(A_2)) \times \cdots \times P(dom(A_n))$.

Definição 4.2: Uma tupla $t_i = <d_{i1}, d_{i2}, ..., d_{in}>$ é chamada de *tupla arbitrária* se $t_i \in P(dom(A_1)) \times P(dom(A_2)) \times \cdots \times P(dom(A_n))$ e, portanto, $d_{ij} \subseteq dom(A_j)$, j = 1, ..., n.

Definição 4.3: Uma tupla $t_i = <d_{i1}, d_{i2}, ..., d_{in}>$ é chamada de *tupla aproximada* se $t_i \in R$ e, portanto, $\in P(dom(A_1)) \times P(dom(A_2)) \times \cdots \times P(dom(A_n))$. Cada $d_{ij} \subseteq dom(A_j)$, j = 1, ..., n.

Exemplo 4.1: Considere um domínio passível de ser descrito por três atributos A_1, A_2 e A_3, cujos possíveis valores são, respectivamente,

$$A_1 = \{a, b\},\ A_2 = \{c, d\} \text{ e } A_3 = \{e, f\}$$

Então

$P(\{a, b\}) = \{\{a\}, \{b\}, \{a, b\}\}$

$P(\{c, d\}) = \{\{c\}, \{d\}, \{c, d\}\}$

$P(\{e, f\}) = \{\{e\}, \{f\}, \{e, f\}\}$

Tem-se pois que

$P(\{a, b\}) \times P(\{c, d\}) \times P(\{e, f\}) =$

$= \{\{a\}, \{b\}, \{a, b\}\} \times \{\{c\}, \{d\}, \{c, d\}\} \times \{\{e\}, \{f\}, \{e, f\}\} =$

$= \{<\{a\}, \{c\}, \{e\}>, <\{a\}, \{c\}, \{f\}>, <\{a\}, \{c\}, \{e, f\}>,$

$<\{a\}, \{d\}, \{e\}>, <\{a\}, \{d\}, \{f\}>, <\{a\}, \{d\}, \{e, f\}>,$

$<\{a\}, \{c, d\}, \{e\}>, <\{a\}, \{c, d\}, \{f\}>, <\{a\}, \{c, d\}, \{e, f\}>,$

$<\{b\}, \{c\}, \{e\}>, <\{b\}, \{c\}, \{f\}>, <\{b\}, \{c\}, \{e, f\}>,$

[4] Conjunto de todos os possíveis subconjuntos.

<{b}, {d}, {e}>, <{b}, {d}, {f}>, <{b}, {d}, {e, f}>,
<{b}, {c, d}, {e}>, <{b}, {c, d}, {f}>, <{b}, {c, d}, {e, f}>,
<{a, b}, {c}, {e}>, <{a, b}, {c}, {f}>, <{a, b}, {c}, {e, f}>,
<{a, b}, {d}, {e}>, <{a, b}, {d}, {f}>, <{a, b}, {d}, {e, f}>,
<{a, b}, {c, d}, {e}>, <{a, b}, {c, d}, {f}>, <{a, b}, {c, d}, {e, f}>}.

Uma possível relação aproximada, pois, é:

{<{a}, {d}, {f}>, <{a, b}, {c}, {f}>, <{a, b}, {c, d}, {e}>, <{a, b}, {c, d}, {e, f}>}

Note que uma relação pertencente a uma Base de Dados Relacional padrão (i.e. não aproximada) pode ser vista como um caso particular da Base de Dados Relacional Aproximada, como mostra a relação:

{<{a}, {c}, {e}>, <{a}, {c}, {f}>, <{b}, {d}, {e}>, <{b}, {d}, {f}>}

Enquanto no Modelo Relacional as tuplas são definidas por valores atômicos de atributos, no Modelo Relacional Aproximado elas podem ser definidas, também, por conjuntos de tais valores[5].

Em uma Base de Dados Relacional, uma vez que os valores dos atributos são atômicos, existe uma única interpretação para cada tupla t_i – a própria tupla. Numa Base de Dados Relacional Aproximada esse nem sempre é o caso.

Seja [a] a notação usada para representar a classe de equivalência do elemento a. Se a é um conjunto de valores, então a classe de equivalência [a] é formada pela união das classes de equivalência dos elementos de a. Se $a = \{a_1, a_2, ..., a_n\}$, então $[a] = [a_1] \cup [a_2] \cup \cdots \cup [a_n]$.

Definição 4.4: Uma *interpretação* $\alpha = <a_1, a_2, ..., a_n>$ de uma tupla aproximada $t_i = <d_{i1}, d_{i2}, ..., d_{in}>$ é qualquer atribuição de valor tal que $a_j \in d_{ij}$, $j = 1, ..., n$.

[5] Para simplificar a notação as chaves em conjuntos unitários são omitidas.

O espaço das interpretações é o produto cartesiano dos domínios dos atributos $dom(A_1) \times dom(A_2) \times \cdots \times dom(A_n)$. Para uma dada relação R, entretanto, esse espaço é limitado àquelas tuplas que são válidas de acordo com a semântica de R.

Exemplo 4.2: Seja o esquema de relação aproximada FILME(CODIGO, TITULO, ATOR_PRINC, GENERO) e considere a instância da relação aproximada FILME, apresentada na Figura 4.1.

FILME	CODIGO	TITULO	ATOR_PRINC	GENERO
	2	The Day World Ended	Richard Denning	{Ficção, Mistério}
	3	{The Boat, Das Boat}	{Jurgen Prochbow, J. Prochbow}	{Guerra, Romance}
	4	David and Bathsheba	Gregory Peck	Épico
	13	Evil Under The Sun	Peter Ustinov	Suspense
	15	The Exorcist	Max Von Sydow	{Terror, Suspense}
	18	A Bridge Too Far	James Chan	II Guerra Mundial
	21	The Guns of Navarone	Gregory G. Peck	II Guerra Mundial
	22	Platoon	Tom Berenger	Guerra do Vietnã

Figura 4.1: Uma instância da relação aproximada FILME.

Considere $t_1 \in$ FILME tal que t_1[CODIGO] = <‘15’>. As tuplas α_1 = <‘15’, ‘The Exorcist’, ‘Max Von Sydow’, ‘Terror’> e α_2 = <‘15’, ‘The Exorcist’, ‘Max Von Sydow’, ‘Suspense’> são interpretações de t_1.

Uma Base de Dados Relacional Aproximada tem sempre associada a ela uma relação de indiscernibilidade definida sobre o conjunto de todos os valores de atributos participantes da base. Essa relação é sempre notada por IND e é parte integrante da base de dados. Os autores do Modelo Relacional Aproximado não apresentaram uma definição formal da maneira como a IND é formada sobre os valores dos domínios pertencentes à base e, como contribuição deste trabalho de pesquisa ao modelo, é proposta uma definição para a IND a seguir.

Considere um conjunto de critérios $\{C_1, ..., C_n\}$ associados a cada um dos domínios $D_1, ..., D_n$ usados em uma Base de Dados Relacional Aproximada, respectivamente. Cada um desses critérios C_i (i = 1, ..., n) estabelece quando os valores de atributo do domínio correspondente D_i (i = 1, ..., n) são indiscerníveis.

Definição 4.5: Cada relação aproximada, da Base de Dados Relacional Aproximada, com esquema $R(A_1, A_2, ..., A_n)$ está associada à relação IND, chamada de relação de indiscernibilidade, definida no conjunto união VA dos valores dos atributos $A_1, A_2, ..., A_n$ por

$$<a_{ij}, a_{ik}> \in \text{IND} \Leftrightarrow a_{ij} \text{ e } a_{ik} \text{ são indiscerníveis de acordo com o critério } C_i.$$

A relação IND da Definição 4.5 é uma relação de equivalência e, conseqüentemente, (ver Definição A.5 do ANEXO A), IND induz uma partição no conjunto VA, em classes de equivalência. Cada uma destas classes de equivalência é caracterizada por um identificador.

Exemplo 4.3: Considere o esquema de relação FILME(CODIGO, TITULO, ATOR_PRINC, GENERO) e o conjunto de critérios $\{C_1, C_2, C_3, C_4\}$.

- O critério C_1, definido sobre o domínio CODIGO, estabelece que dois valores de atributo CODIGO pertencem à IND se e somente se forem iguais;
- O critério C_2, definido sobre o domínio TITULO, estabelece que dois valores de atributo TITULO pertencem à IND se forem iguais, se um for a tradução do outro ou se tiverem a grafia semelhante;
- O critério C_3, definido sobre o domínio ATOR_PRINC, estabelece que dois valores de atributo ATOR_PRINC pertencem à IND se forem iguais, se tiverem a grafia semelhante, se um for abreviação do outro ou se ambos se referem à mesma pessoa;
- O critério C_4, definido sobre o domínio GENERO, estabelece que dois valores de atributo GENERO pertencem à IND se forem iguais, se um for tradução do outro ou se forem sinônimos com relação a gêneros de filmes.

Considere agora o conjunto de todos os possíveis valores de atributo associados a todos os atributos que definem a Base de Dados Relacional Aproximada e, para cada um desses conjuntos de valores de atributo, considere o correspondente critério que estabelece a indiscernibilidade (ou seja, a relação IND restrita ao conjunto dos valores em questão) de seus valores como mostram Tabela 4.1, Tabela 4.2, Tabela 4.3 e Tabela 4.4. A cada grupo de valores indiscerníveis de atributo é atribuído um identificador (foi escolhido um identificador numérico neste exemplo).

Tabela 4.1: Valores do atributo CODIGO agrupados pelo critério C_1, com apenas um valor por grupo.

$Valores_{CODIGO}$	Identificador
2	5
3	6
4	7
13	8
15	9
18	12
21	13
22	14

Tabela 4.2: Valores do atributo TITULO agrupados pelo critério C_2. Note que os valores 'The Boat' e 'Das Boat' são indiscerníveis segundo C_2, pois formam um único grupo com um único identificador associado.

$Valores_{TITULO}$	Identificador
The Day World Ended	10
The Boat	11
David and Bathsheba	15
Evil Under The Sun	16
The Exorcist	17
A Bridge Too Far	18
The Guns of Navarone	19
Das Boat	11
Platoon	20

Tabela 4.3: Valores do atributo ATOR_PRINC agrupados pelo critério C_3. Note que os valores 'Gregory Peck' e 'Gregory G. Peck' são indiscerníveis segundo C_3, pois formam um único grupo com um único identificador associado.

$Valores_{ATOR\_PRINC}$	Identificador
Richard Denning	21
Jurgen Prochbow	22
J. Prochbow	22
Gregory Peck	23
Peter Ustinov	24
Max Von Sydow	25
James Chan	26
Gregory G. Peck	23
Tom Berenger	27

Tabela 4.4: Valores do atributo GENERO agrupados pelo critério C_4. Note que valores, como 'Ficção' e 'Sci-Fi', foram agrupados e possuem identificadores comuns e, portanto, são indiscerníveis segundo o critério C_4.

$Valores_{GENERO}$	Identificador
Romance	1
Guerra	2
Terror	3
Suspense	4
Amor	1
II Guerra Mundial	2
Guerra Civil	2
Medo	3
Guerra do Vietnã	2
Mistério	4
Horror	3
Ficção	28
Sci-Fi	28

Como definido anteriormente, VA = $Valores_{CODIGO} \cup Valores_{TITULO} \cup Valores_{ATOR\_PRINC} \cup Valores_{GENERO}$. A relação IND definida em VA é dada por:

IND = { <2, 2>, <3, 3>, <4, 4>, <13, 13>, <15, 15>, <18, 18>, <21, 21>, <22, 22>,
<The Day World Ended, The Day World Ended>, <The Boat, The Boat>,
<David and Bathsheba, David and Bathsheba>, <Platoon, Platoon>,
<Evil Under The Sun, Evil Under The Sun>, <The Exorcist, The Exorcist>,
<A Bridge Too Far, A Bridge Too Far >, <Das Boat, Das Boat>,
<The Guns of Navarone, The Guns of Navarone>, <The Boat, Das Boat >,
< Das Boat, The Boat >, < Richard Denning, Richard Denning>,
<Jurgen Prochbow, Jurgen Prochbow>, <J. Prochbow, J. Prochbow>,
<Gregory Peck, Gregory Peck>, <Peter Ustinov, Peter Ustinov>,
<Max Von Sydow, Max Von Sydow>, <James Chan, James Chan>,
<Gregory G. Peck, Gregory G. Peck>, <Tom Berenger, Tom Berenger>,
<Gregory Peck, Gregory G. Peck>, <Gregory G. Peck, Gregory Peck>,
<Jurgen Prochbow, J. Prochbow>, <J. Prochbow, Jurgen Prochbow>,
<Romance, Romance>, <Guerra, Guerra>, <Terror, Terror>, <Suspense, Suspense>,
<Amor, Amor>, <II Guerra Mundial, II Guerra Mundial>, <Medo, Medo>,
<Guerra Civil, Guerra Civil>, <Guerra do Vietnã, Guerra do Vietnã>,
<Mistério, Mistério>, <Horror, Horror>, <Ficção, Ficção>, <Sci-Fi, Sci-Fi>,
<Romance, Amor>, <Amor, Romance>, <Guerra, II Guerra Mundial>,
<Guerra, Guerra Civil>, <Guerra, Guerra do Vietnã>, <II Guerra Mundial, Guerra>,
<II Guerra Mundial, Guerra Civil>, <II Guerra Mundial, Guerra do Vietnã>,
<Guerra Civil, Guerra>, <Guerra Civil, II Guerra Mundial >,
<Guerra Civil, Guerra do Vietnã >, <Guerra do Vietnã, Guerra>,
<Guerra do Vietnã, II Guerra Mundial>, <Guerra do Vietnã, Guerra Civil>,
<Terror, Medo>, <Terror, Horror>, <Medo, Terror>, <Medo, Horror>,
<Horror, Terror>, <Horror, Medo>, <Suspense, Mistério>, <Mistério, Suspense>,
<Ficção, Sci-Fi>, <Sci-Fi, Ficção >
}

A relação IND é uma relação de equivalência (ou seja, reflexiva, simétrica e transitiva) e, como tal, induz uma partição no conjunto VA, como pode ser visualizado na Figura 4.2.

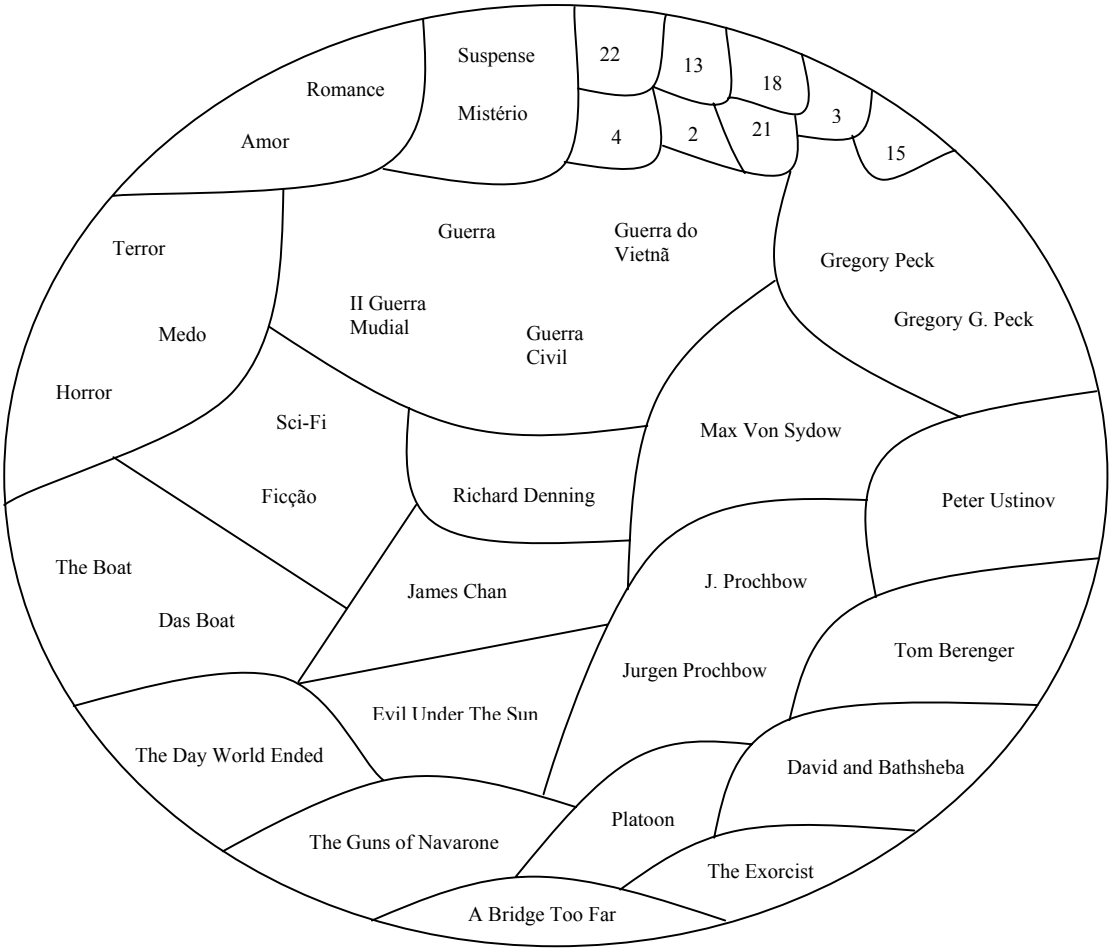


Figura 4.2: Classes de equivalência induzidas por IND.

As tuplas de uma relação aproximada, assim como as tuplas das relações do Modelo Relacional, devem ser distintas entre si, ou seja, não deve haver redundância de tuplas na relação aproximada. Apesar de não haver redundância entre as tuplas de uma relação aproximada, é possível e natural que haja mais de uma tupla com a mesma interpretação.

Definição 4.6: Duas tuplas $t_i = <d_{i1}, d_{i2}, ..., d_{in}>$ e $t_k = <d_{k1}, d_{k2}, ..., d_{kn}>$ são *redundantes* se $[d_{ij}] = [d_{kj}]$ para todo j = 1, ..., n.

Exemplo 4.4: Sejam t_1 = <‘3’, ‘The Boat’, ‘Jurgen Prochbow’, ‘Romance’> e t_2 = <‘3’, ‘Das Boat’, ‘J. Prochbow’, ‘Romance’> tuplas do esquema da relação aproximada FILME apresentado no Exemplo 4.2. Estas tuplas são redundantes, pois $[d_{11}] = [d_{21}] = \{3\}$, $[d_{12}] = [d_{22}] = \{\text{The Boat, Das Boat}\}$, $[d_{13}] = [d_{23}] = \{\text{Jurgen Prochbow, J. Prochbow}\}$ e $[d_{14}] = [d_{24}] = \{\text{Romance}\}$.

Exemplo 4.5: Seja o esquema de relação aproximada PESSOA(Nome, Idade) e a relação de indiscernibilidade IND da relação aproximada PESSOA definindo, sobre o atributo IDADE, as seguintes classes de equivalência: IND = {{Criança, Pré-Adolescente}, {Jovem, Adolescente}, {Adulto}, {Idoso, Senhor}}. Sejam também as tuplas t_1 = <‘José da Silva’, ‘Jovem’> e t_2 = <‘José da Silva’, ‘Adolescente’>. Estas tuplas são redundantes, pois $[d_{11}] = [d_{21}] = \{\text{José da Silva}\}$ e $[d_{12}] = [d_{22}] = \{\text{Jovem, Adolescente}\}$.

Exemplo 4.6: Sejam o esquema PESSOA e a relação de indiscernibilidade IND definidos no Exemplo 4.5 e as tuplas t_1 = <‘José da Silva’, {‘Jovem’, ‘Adulto’}> e t_2 = <‘José da Silva’, ‘Jovem’>. Estas tuplas, apesar de não serem redundantes, pois $[d_{11}] = [d_{21}] = \{\text{José da Silva}\}$ mas $[d_{12}] = \{\text{Jovem, Adolescente, Adulto}\}$ e $[d_{22}] = \{\text{Jovem, Adolescente}\}$, possuem uma interpretação idêntica entre elas: $\alpha_1 = \alpha_2$ = <‘José da Silva’, ‘Jovem’>.

4.3 Sobre as Consultas Aproximadas

Assim como a indiscernibilidade, os conceitos de aproximação inferior e de aproximação superior são partes integrantes de uma Base de Dados Relacional Aproximada. Estes três conceitos permitem o estabelecimento de um mecanismo de consultas à base que pode também ser caracterizado como aproximado.

Com base nos conceitos de aproximação inferior e aproximação superior, um mecanismo de consulta aproximada pode ser implementado em dois passos. O primeiro deles retorna os elementos da relação aproximada que pertencem à aproximação inferior ou seja, aqueles elementos que com certeza pertencem à relação aproximada resultante da consulta. Num segundo passo, retorna os elementos da aproximação superior ou seja, aqueles elementos que possivelmente pertencem à relação aproximada resultante da consulta. Toda operação de consulta

à Base de Dados Relacional Aproximada acessará, implicitamente, a relação de indiscernibilidade IND, além das relações aproximadas indicadas na consulta.

Note que, como a aproximação inferior está contida na aproximação superior (ver Definição 2.4), uma vez recuperados esses dois conjuntos de elementos da relação aproximada, é trivial identificar entre eles aqueles que pertencem à região duvidosa (ver Definição 2.5). Para efeito de implementação, durante uma consulta, com o objetivo de identificar quando uma tupla pertence à região duvidosa, um novo atributo, chamado DUV, é adicionado à descrição de cada tupla que pertence à base. Quando o valor desse atributo for '*' a tupla em questão pertence à região duvidosa, caso contrário, pertence à região positiva.

Exemplo 4.7: Seja a relação aproximada FILME apresentada na Figura 4.1. O usuário da base de dados quer retornar o nome dos filmes desta relação nos quais ATOR_PRINC = ['Gregory Peck']. No Modelo Relacional, apenas a tupla t_1[CODIGO, ATOR_PRINC] = <'4', 'Gregory Peck'> pertenceria à relação resultante da consulta enquanto que, no Modelo Relacional Aproximado, a tupla t_2[CODIGO, ATOR_PRINC] = <'21', 'Gregory G. Peck'> também é retornada. Esse acréscimo na recuperação de informação se deve aos diferentes mecanismos de consulta já que, o Modelo Relacional Aproximado utiliza equivalência como comparação dos valores e não a igualdade. Conforme pode ser visto na Figura 4.2, os valores 'Gregory Peck' e 'Gregory G. Peck' são equivalentes ou seja, ['Gregory Peck'] = ['Gregory G. Peck']. As aproximações inferior e superior, e a relação resultante da consulta estão representadas, respectivamente, na Figura 4.3, Figura 4.4 e Figura 4.5. A região duvidosa não está representada pois a mesma é vazia, para este exemplo.

INF	CODIGO	TITULO	ATOR_PRINC	GENERO
	4	David and Bathsheba	Gregory Peck	Épico
	21	The Guns of Navarone	Gregory G. Peck	II Guerra Mundial

Figura 4.3: Aproximação inferior da consulta do Exemplo 4.7.

SUP	CODIGO	TITULO	ATOR_PRINC	GENERO
	4	David and Bathsheba	Gregory Peck	Épico
	21	The Guns of Navarone	Gregory G. Peck	II Guerra Mundial

Figura 4.4: Aproximação superior da consulta do Exemplo 4.7.

TITULO	DUV
David and Bathsheba	Null
The Guns of Navarone	Null

Figura 4.5: O resultado da consulta do Exemplo 4.7.

Exemplo 4.8: Seja a relação aproximada FILME apresentada na Figura 4.1. O usuário da base de dados quer retornar o nome dos filmes desta relação nos quais GENERO = [‘Suspense’]. No Modelo Relacional apenas a tupla t_1[CODIGO, GENERO] = <‘13’, ‘Suspense’> seria retornada como pertencente à relação resultante da consulta, porém o Modelo Relacional Aproximado possui o mecanismo de consulta aproximado e, assim, as tuplas t_2[CODIGO, GENERO] = <‘15’, {‘Terror’, ‘Suspense’}> e t_3[CODIGO, GENERO] = <‘2’, {‘Ficção’, ‘Mistério’}> também são retornadas. Por não satisfazerem exatamente ao que foi solicitado pelo usuário, as tuplas t_2 e t_3 são classificadas como pertencentes à região duvidosa. As aproximações inferior e superior, a região duvidosa e a relação resultante da consulta estão representadas, respectivamente, na Figura 4.6, Figura 4.7, Figura 4.8 e Figura 4.9.

INF	CODIGO	TITULO	ATOR_PRINC	GENERO
	13	Evil Under The Sun	Peter Ustinov	Suspense

Figura 4.6: Aproximação inferior da consulta do Exemplo 4.8.

SUP	CODIGO	TITULO	ATOR_PRINC	GENERO
	2	The Day World Ended	Richard Denning	{Ficção, Mistério}
	13	Evil Under The Sun	Peter Ustinov	Suspense
	15	The Exorcist	Max Von Sydow	{Terror, Suspense}

Figura 4.7: Aproximação superior da consulta do Exemplo 4.8.

DUVI	CODIGO	TITULO	ATOR_PRINC	GENERO
	2	The Day World Ended	Richard Denning	{Ficção, Mistério}
	15	The Exorcist	Max Von Sydow	{Terror, Suspense}

Figura 4.8: A região duvidosa da consulta do Exemplo 4.8.

TITULO	DUV
Evil Under The Sun	Null
The Exorcist	*
The Day World Ended	*

Figura 4.9: O resultado da consulta do Exemplo 4.8.

4.4 Considerações Finais

Este capítulo apresentou e discutiu uma extensão do Modelo Relacional, chamada Modelo Relacional Aproximado, no qual foram incorporados conceitos da TCA e se definiu uma Base de Dados Relacional Aproximada, ressaltando suas vantagens sobre as Bases de Dados tradicionais, refinando a formalidade da teoria e contribuindo com a definição formal da maneira como a IND é construída. No próximo capítulo, são apresentados e discutidos os principais Operadores Relacionais Aproximados, que permitem a recuperação de informação da Base de Dados Relacional Aproximada. Todos eles são dependentes e estão fundamentados nos conceitos de relação de indiscernibilidade, de aproximação inferior e de aproximação superior, como tratados neste capítulo.

CAPÍTULO 5. OPERADORES RELACIONAIS APROXIMADOS

A funcionalidade de uma Base de Dados Relacional Aproximada é dependente do conjunto de Operadores Relacionais Aproximados disponibilizados. Este capítulo apresenta e discute vários deles, que foram propostos em [Beauboeuf e Petry 1994], apresentando a operacionalização de cada um por meio de sua descrição em pseudocódigos e comentários sobre a implementação, que são contribuições deste trabalho.

Os três operadores de atualização, isto é, *DELETE*, *INSERT* e *MODIFY*, do Modelo Relacional Aproximado são similares às do Modelo Relacional e por esta razão não são abordados. A única observação a ser feita é que o usuário deve lembrar que os atributos podem ser multivalorados e o foco dos operadores são as classes de equivalência às quais as tuplas ou os valores de seus atributos pertencem.

Os operadores relacionais aproximados são baseados nos operadores relacionais convencionais e foram desenvolvidos para trabalhar com as relações aproximadas[6]. As definições para operações de conjuntos em relações aproximadas são comparáveis àquelas definidas para o Modelo Relacional e, com exceção da junção, as operações binárias requerem que as relações envolvidas (argumentos da operação) sejam união compatíveis (ver Definição 3.19).

Para as definições que seguem, assume-se que as relações envolvidas são união compatíveis.

5.1 A União Aproximada

A *união aproximada* é uma operação binária entre duas relações aproximadas que resulta em uma outra relação aproximada. Como tal, ela é definida em termos de suas aproximações inferior e superior como pode ser visto em sua definição formal dada na Definição 5.1. A Figura 5.3 mostra o pseudocódigo da operação, que está exemplificada no Exemplo 5.4 na seção 5.7.

[6] Para a implementação dos novos operadores foi necessário o desenvolvimento de um mecanismo responsável pela sinalização das tuplas das relações aproximadas como pertencentes à aproximação inferior ou região duvidosa das relações sendo consideradas. Esse mecanismo também é uma contribuição deste trabalho de pesquisa já que a notação utilizada pelos autores (colocar o identificador da tupla entre parênteses quando esta pertence à região duvidosa) não é a mais adequada para a implementação.

Definição 5.1: Dadas duas relações aproximadas R_1 e R_2, a *união aproximada* de R_1 e R_2, denotada por $R_1 \cup R_2$, é uma relação aproximada, de mesmo esquema de R_1 e R_2, definida por suas aproximações inferior e superior, dadas por

$$A_{inf}(R_1 \cup R_2) = \{t \mid t \in A_{inf}(R_1) \cup A_{inf}(R_2)\}$$
$$A_{sup}(R_1 \cup R_2) = \{t \mid t \in A_{sup}(R_1) \cup A_{sup}(R_2)\}$$

A aproximação inferior da relação aproximada resultante $R_1 \cup R_2$ mantém aquelas tuplas que fazem parte da aproximação inferior de R_1 ou da aproximação inferior de R_2 ou de ambas. A aproximação superior da relação aproximada resultante $R_1 \cup R_2$ mantém aquelas tuplas que fazem parte da aproximação superior de R_1 ou da aproximação superior de R_2 ou de ambas.
Como ambas as aproximações são conjuntos, a implementação da união aproximada implicitamente trata a situação de tuplas redundantes, eliminando a ocorrência de tuplas que aparecem repetidas tanto na aproximação inferior quanto na aproximação superior. No caso de uma tupla que, por exemplo, pertence a $A_{inf}(R_1)$ e a $A_{sup}(R_2)$, essa tupla obviamente está tanto na $A_{inf}(R_1 \cup R_2)$ quanto na $A_{sup}(R_1 \cup R_2)$. Para efeito de implementação, entretanto, essa tupla é abordada como pertencente à aproximação inferior, ou seja, como certamente pertencente à relação.

Antes da apresentação do pseudocódigo do procedimento que realiza a união aproximada na Figura 5.3, são mostrados os pseudocódigos de dois procedimentos auxiliares monta_classe(lista, atrib) (Figura 5.1) para montar as classes de equivalência e seleciona_tupla_redundante(relac, tup) (Figura 5.2), utilizados por alguns dos procedimentos descritos neste capítulo.

```
Função monta_classe(lista, atrib) : Classe
  {Parâmetros de Entrada:
   - lista é a lista de valores dos quais será montada a classe
   - atrib é o atributo que representa o domínio dos valores em lista
   Parâmetro de Saída:
   - Classe é a classe de equivalência ou união de classes de equivalência
     resultante}

  Variáveis Locais
    listai {Um elemento de lista}
    z      {Conjunto temporário de valores}
    zk     {Um elemento de Z}
    Cl     {Identificador da classe de equivalência à qual um determinado
            listai pertence}
    kont   {contador de elementos da classe resultante}

Início
  kont ← 0
  Classe ← Null

  Para todo listai ∈ lista faça
  Início
    {Seleciona o valor do atributo CLASSE na tabela IND onde o
     atributo DOMINIO = domínio de atrib e o atributo VALOR = listai}
    Cl ← seleciona_identificador_classe(atrib, listai)

    {Seleciona todos os valores do atributo VALOR na tabela IND
     onde o atributo DOMINIO = domínio de atrib e o atributo CLASSE = Cl}
    z ← seleciona_valores(atrib, Cl)

    Para todo zk ∈ z faça
    Início
      Classe[kont] ← zk
      kont ← kont + 1
    Fim
  Fim
Fim
```

Figura 5.1: Pseudocódigo da função monta_classe(lista, atrib).

A função monta_classe(lista, atrib) tem como parâmetros de entrada uma lista de valores e o nome de um atributo e retorna uma classe de equivalência ou união de classes de equivalência à qual esses valores pertencem, dentro do domínio do atributo atrib. Isto é feito dentro de um *loop* que percorre todos os valores contidos em lista. Para cada valor:

1) é encontrado o identificador (Cl) de sua classe de equivalência, de acordo com a tabela IND;
2) são então selecionados todos os valores em IND que pertencem à classe Cl, dentro do domínio do atributo atrib. Todos esses valores são acumulados na variável de retorno Classe.

Ao final do procedimento, a variável Classe contém a classe de equivalência ou união de classes de equivalência à qual os valores fornecidos pertencem.

Como comentado anteriormente uma relação aproximada X é abordada, do ponto de vista da implementação, como dois conjuntos: um com as tuplas que com certeza pertencem à relação X (i.e., $A_{inf}(X)$) e outro com aquelas que possivelmente pertencem à X (i.e., duv(X)).

```
Função seleciona_tupla_redundante(relac, tup) : t
  {Parâmetros de Entrada:
   - relac é a relação onde será procurada a tupla redundante
   - tup é a tupla que será procurada na relação relac em busca de redundância
  {Parâmetro de Saída:
   - t é a tupla resultado e caso não exista tupla redundante t será Null}

  Variáveis Locais
    tx      {é uma tupla da relação relac}
    dxn     {Valor do enésimo atributo de uma tupla tx}
    An      {Enésimo atributo de tx}
    dyn     {Valor do enésimo atributo da tupla tup}
    Bn      {Enésimo atributo de tup}
    classex {Lista de classes de equivalência de uma tupla tx}
    classey {Lista de classes de equivalência da tupla tup}
    kont    {contador}

Início
  kont ← 0
  classey ← Null

  Para todo dyn ∈ tup faça
  Início
    classey[kont] ← monta_classe(dyn, Bn)
    kont ← kont + 1
  Fim

  Para todo tx ∈ relac faça
  Início
    kont ← 0
    classex ← Null

    Para todo dxn ∈ tx faça
    Início
      classex[kont] ← monta_classe(dxn, An)
      kont ← kont + 1
    Fim

    Se classex = classey então
      Retorna tx {Retorna a tupla redundante encontrada e interrompe a
                  a execução da função}
  Fim

  Retorna Null {Retorna Null pois não encontrou tupla redundante}
Fim
```

Figura 5.2: Pseudocódigo da função seleciona_tupla_redundante(relac, tup).

A função seleciona_tupla_redundante(relac, tup) tem como parâmetros de entrada uma relação aproximada e uma tupla. O procedimento busca pela ocorrência de tup em relac. Na eventualidade de ser bem sucedido, retorna a tupla em questão, caso contrário, retorna *Null*. Note-se que a busca por tuplas redundantes vai evidenciar as tuplas que estão na mesma classe de equivalência de tup, com relação a todos os atributos que as descrevem.

O primeiro *loop* do procedimento constrói a classe de equivalência ou união de classes de equivalência relativas a cada valor dos atributos que descrevem tup. O segundo *loop*, para cada tupla tx de relac:

1) constrói a classe de equivalência à qual tx pertence com relação a cada um de seus atributos;
2) compara, por igualdade, a classe de equivalência encontrada em 1) com aquela construída no primeiro *loop*. Em caso de igualdade tx é retornada e o procedimento é interrompido.

A função runion(X, Y), descrito na Figura 5.3, executa a operação união aproximada e tem como parâmetros de entrada duas relações aproximadas. O retorno é a relação aproximada T, especificada por meio dos conjuntos $A_{inf}(T)$ e duv(T). A função executa um *loop* que percorre todas as tuplas tx da primeira relação inserindo-as na relação resultado T, mantendo o valor do atributo DUV de cada tupla. Em seguida é executado um novo *loop*, agora sobre as tuplas de Y, que percorre todas as ty e, por meio da função seleciona_tupla_redundante(relac, tup), verifica a existência de uma tupla t redundante a ty na relação T. Se não existe redundância, ty é inserida em T, mantendo o valor do atributo DUV. Se existe uma t redundante a ty e ty[DUV] é igual à *Null* (isto é, $ty \in A_{inf}(Y)$) e t[DUV] é igual a "*" (isto é, $t \in duv(T)$), t é removida de T e ty é inserida em T. Esta substituição deve ser feita pois, neste caso, ty pertence à aproximação inferior enquanto que t pertence à região duvidosa. Com o término do *loop* de Y, é retornada a relação T.

```
Função runion(X, Y) : T
  {Parâmetros de Entrada:
   - X e Y são as relações alvo da operação
  {Parâmetro de Saída:
   - T é a relação resultante da operação expressa por A_inf(T) e duv(T)}

  Variáveis Globais
    tx   {Uma tupla de X e tx[A] representa o conjunto de valores de um atributo
          A numa determinada tupla tx}
    ty   {Uma tupla de Y e ty[B] representa o conjunto de valores de um atributo
          B numa determinada tupla ty}
    t    {Uma tupla de T e t[C] representa o conjunto de valores de um atributo
          C numa determinada tupla t}

Início
  Para todo tx ∈ X faça
  Início
    Insere(tx, T) {Insere a tupla tx na relação resultado T mantendo o valor do
                   atributo DUV}
  Fim

  Para todo ty ∈ Y faça
  Inicio
    t ← seleciona_tupla_redundante(T, ty) {Seleciona a tupla de T que é
                                           redundante à ty}
    Se t ≠ Null então
      Se (ty[DUV] = Null) e (t[DUV] = '*') então
        Início
          Remove(t, T) {Remove a tupla t da relação resultado T pois ela
                        pertence à região duvidosa}
          Insere(ty, T) {Insere a tupla ty na relação resultado T no lugar da
                         tupla t pois ty pertence à região positiva}
        Fim
    Senão
      Insere(ty, T) {Insere a tupla ty na relação resultado T mantendo o valor
                     do atributo DUV}
  Fim
Fim
```

Figura 5.3: Pseudocódigo da operação união aproximada.

5.2 A Intersecção Aproximada

A *intersecção aproximada* é uma operação binária entre duas relações aproximadas cujo resultado é, também, uma relação aproximada e está definida formalmente na Definição 5.2. A Figura 5.4 mostra o pseudocódigo da operação, que está exemplificada no Exemplo 5.5 na seção 5.7.

Definição 5.2: Sejam R_1 e R_2 duas relações aproximadas. A *intersecção aproximada* de R_1 e R_2, denotada por $R_1 \cap R_2$, é uma relação aproximada, de mesmo esquema de R_1 e R_2, definida por suas aproximações inferior e superior, dadas por

$$A_{inf}(R_1 \cap R_2) = \{t \mid t \in A_{inf}(R_1) \cap A_{inf}(R_2)\}$$
$$A_{sup}(R_1 \cap R_2) = \{t \mid t \in A_{sup}(R_1) \cap A_{sup}(R_2)\}$$

Na intersecção aproximada, a comparação das tuplas é baseada na redundância e não na igualdade como acontece na operação de mesmo nome relativa ao Modelo Relacional. Na eliminação de tuplas redundantes é sempre mantida a tupla pertencente à aproximação inferior.

A função rintersection(X, Y), que faz uso da função seleciona_tupla_redundante(relac, tup) descrita na Figura 5.2, executa a operação intersecção aproximada e tem como parâmetros de entrada duas relações aproximadas. O retorno é a relação aproximada T, especificada por meio dos conjuntos $A_{inf}(T)$ e duv(T). A função faz um *loop* que percorre todas as tx de X verificando se existe, por meio da função seleciona_tupla_redundante(relac, tup), uma tupla ty redundante à tx, na relação Y. Caso exista, duas situações podem acontecer:

1) tx[DUV] = ty[DUV] = *Null*, tx é inserida em T;
2) tx[DUV] = '*' ou ty[DUV] = '*', a tupla que tiver o valor do atributo DUV = '*' é inserida em T pois a intersecção acontece na região duvidosa.

```
Função rintersection (X, Y) : T
  {Parâmetros de Entrada:
   - X e Y são as relações alvo da operação
  {Parâmetro de Saída:
   - T é a relação resultante da operação expressa por A_inf(T) e duv(T)}

  Variáveis Globais
    tx    {Uma tupla de X e tx[A] representa o conjunto de valores de um
          atributo A numa determinada tupla tx}
    ty    {Uma tupla de Y e ty[B] representa o conjunto de valores de um
          atributo B numa determinada tupla ty}

Início
  Para todo tx ∈ X faça
  Inicio
    ty ← seleciona_tupla_redundante(Y, tx) {Seleciona a tupla de Y que é
                                            redundante à tx}

    Se ty ≠ Null então
    Início
      Se (tx[DUV] = Null) e (ty[DUV] = Null) então
        Insere(tx, T) {Insere a tupla tx na relação resultado T mantendo o
                       valor do atributo DUV = Null pois ambas pertencem à
                       aproximação inferior}
      Senão
        Se (tx[DUV] = '*') então
          Insere(tx, T) {Insere a tupla tx na relação resultado T mantendo o
                         valor do atributo DUV = '*' pois a intersecção
                         acontece na aproximação superior}
        Senão
          Insere(ty, T) {Insere a tupla ty na relação resultado T mantendo o
                         valor do atributo DUV = '*' pois ela pertence à região
                         duvidosa}
    Fim
  Fim
Fim
```

Figura 5.4: Pseudocódigo da operação intersecção aproximada.

5.3 A Diferença Aproximada

A *diferença aproximada* entre duas relações aproximadas é uma relação aproximada definida por aqueles elementos da primeira relação que não pertencem à segunda relação. A operação está definida formalmente na Definição 5.3 e exemplificada no Exemplo 5.6 na seção 5.7. A Figura 5.5 mostra o seu pseudocódigo.

Definição 5.3: Sejam R_1 e R_2 duas relações aproximadas. A *diferença aproximada* de R_1 e R_2, denotada por $R_1 - R_2$, é uma relação aproximada, de mesmo esquema de R_1 e R_2, definida por suas aproximações inferior e superior, dadas por

$$A_{inf}(R_1 - R_2) = \{t \mid t \in A_{inf}(R_1) \text{ e } t \notin A_{inf}(R_2)\}$$
$$A_{sup}(R_1 - R_2) = \{t \mid t \in A_{sup}(R_1) \text{ e } t \notin A_{sup}(R_2)\}$$

Na eliminação de tuplas redundantes é sempre mantida a tupla pertencente à aproximação inferior.

A função rdifference(X, Y), que faz uso da função seleciona_tupla_redundante(relac, tup) descrita na Figura 5.2, executa a operação diferença aproximada e tem como parâmetros de entrada duas relações aproximadas. O retorno é a relação aproximada T, especificada por meio dos conjuntos $A_{inf}(T)$ e duv(T). A função executa um *loop* que percorre todas as tuplas tx verificando, por meio da função seleciona_tupla_redundante(relac, tup), se existe uma tupla ty redundante à tx, na relação Y. Caso não exista ou, se tx[DUV] = *Null* e ty[DUV] = '*', insere-se tx em T. Esta verificação de tx[DUV] e ty[DUV] deve ser feita pois as tuplas de X somente são adicionadas a T em dois casos, a saber: quando não existir uma ty que seja redundante à tx ou quando, existindo redundância entre as tuplas, tx pertencer à aproximação inferior e ty pertencer à região duvidosa.

```
Função rdifference(X, Y) : T
  {Parâmetros de Entrada:
   - X e Y são as relações alvo da operação
  {Parâmetro de Saída:
   - T é a relação resultante da operação expressa por A_inf(T) e duv(T)}

  Variáveis Globais
    tx   {Uma tupla de X e tx[A] representa o conjunto de valores de um atributo
          A numa determinada tupla tx}
    ty   {Uma tupla de Y e ty[B] representa o conjunto de valores de um atributo
          B numa determinada tupla ty}

Início
  Para todo tx ∈ X faça
  Inicio
    ty ← seleciona_tupla_redundante(Y, tx) {Seleciona a tupla de Y que é
                                            redundante à tx}
    Se ty ≠ Null então
      Se (tx[DUV] = Null) e (ty[DUV] = '*') então
          Insere(tx, T) {Insere a tupla tx na relação resultado T mantendo o
                         valor do atributo DUV }
    Senão
      Insere(tx, T) {Insere a tupla tx na relação resultado T mantendo o valor
                     do atributo DUV}

  Fim
Fim
```

Figura 5.5: Pseudocódigo da operação diferença aproximada.

É importante observar que nas operações convencionais de intersecção e diferença não existe a possibilidade de tuplas redundantes na relação resultante mas, nas operações aproximadas sim. Isto se deve ao fato da aproximação inferior ser um subconjunto da aproximação superior.

5.4 A Seleção Aproximada

A operação *seleção aproximada* é uma operação unária, denotada por σ, que é aplicada a uma relação aproximada, gerando uma outra relação aproximada formada por um subconjunto de tuplas da primeira relação que satisfazem uma condição de seleção, baseada no valor de um ou mais atributos especificados. Esta operação é formalmente definida na Definição 5.4, está exemplificada no Exemplo 5.2 na seção 5.7 e seu pseudocódigo é mostrado na Figura 5.6.

Definição 5.4: Seja R um esquema de relação aproximada e R_1 uma relação aproximada no esquema R. A *seleção aproximada*, $\sigma_{A = \mathbf{a}}(R_1)$, das tuplas de R_1 é uma relação aproximada T, que tem o esquema R onde A é um atributo de R, $\mathbf{a} = \{a_i\}$ e a_i e $b_j \in dom(A)$, $\cup_x$ denota "a união sobre todo x", t[A] denota o valor do atributo A na tupla t e T é definida por suas aproximações inferior e superior dadas por

$$A_{inf}(T) = \{t \in R_1 \mid \cup_i[a_i] = \cup_j[b_j]\},\ a_i \in \mathbf{a},\ b_j \in t[A]$$
$$A_{sup}(T) = \{t \in R_1 \mid \cup_i[a_i] \subseteq \cup_j[b_j]\},\ a_i \in \mathbf{a},\ b_j \in t[A]$$

A função rselect(X, A, a), que faz uso da função monta_classe(lista, atrib) descrita na Figura 5.1, executa a operação seleção aproximada e tem como parâmetros de entrada uma relação aproximada, um atributo e uma lista de valores, sendo que os dois últimos formam a condição de seleção. Primeiramente é construída a classe de equivalência relativa aos valores de a (ou seja, de **a** da Definição 5.4) usando a função monta_classe(lista, atrib) e é armazenada na variável CEa. Em seguida a função executa um *loop* que percorre todas as tx e, para cada uma, constrói a classe de equivalência dos valores do atributo A em tx (CEb). As classes de equivalência (CEa e CEb) são comparadas e se:

1) CEa = CEb, tx é inserida em T com valor *Null* para o atributo DUV pois a tupla pertence à aproximação inferior;
2) CEa ⊂ CEb, tx é inserida em T com valor '*' para o atributo DUV pois a tupla pertence à região duvidosa;

```
Função rselect(X, A, a) : T
  {Parâmetros de Entrada:
   - X é a relação alvo da operação
   - A é o atributo escolhido para a comparação na condição de seleção
   - a é o conjunto de valores a serem comparados na condição de seleção}
  {Parâmetro de Saída:
   - T é a relação resultante da operação expressa por A_inf(T) e duv(T)}

  Variáveis Globais
    CEa  {Classe de equivalência de a}
    tx   {Uma tupla de X e tx[A] representa o conjunto de valores do
          atributo A numa determinada tupla tx}
    CEb  {Classe de equivalência de tx[A]}

Início
  CEa ← monta_classe(a, A)

  Para todo tx ∈ X faça
  Início
    CEb ← monta_classe(tx[A], A)

    Se CEa = CEb então
       Insere(tx, Null, T) {Insere a tupla t_x na relação resultado T com
                            valor Null para o atributo DUV}
    Senão
      Se CEa ⊂ CEb então
         Insere(tx, '*', T) {Insere a tupla t_x na relação resultado T com
                             valor '*' para o atributo DUV}
  Fim
Fim
```

Figura 5.6: Pseudocódigo da operação seleção aproximada.

5.5 A Projeção Aproximada

A operação *projeção aproximada* é uma operação unária, denotada pelo símbolo π, que é aplicada a uma relação aproximada retornando uma nova relação aproximada contendo todas as tuplas da relação aproximada argumento da operação, projetadas sobre um subconjunto de atributos especificados, da relação origem. A operação é especificada formalmente na Definição 5.5, está exemplificada no Exemplo 5.3 na seção 5.7 e seu pseudocódigo é mostrado na Figura 5.7.

Definição 5.5: Seja R_1 uma relação aproximada de esquema R. A operação $\pi_B(R_1)$, retornará uma relação aproximada T de esquema B que é um subconjunto de R, onde

$$T = \{t[B] \mid t \in R_1\}$$

Todas as tuplas em T são representadas somente pelos atributos que definem B. A projeção aproximada de uma relação R_1 sobre um conjunto de atributos B mantém as tuplas que definem R_1 (a menos que ocorram redundâncias, que devem ser tratadas). A projeção não altera a pertinência de uma tupla aos conjuntos aproximação inferior e região duvidosa. Na eliminação de tuplas redundantes, se ambas pertencem à mesma aproximação, inferior ou superior, qualquer uma pode ser eliminada. Se uma delas pertence à aproximação inferior e a outra à região duvidosa, a primeira deve ser mantida.

```
Função rproject(X, LA) : T
  {Parâmetros de Entrada:
   - X é a relação alvo da operação
   - LA é a lista de atributos sobre os quais a relação resultado será projetada
  {Parâmetro de Saída:
   - T é a relação resultante da operação expressa por A_inf(T) e duv(T)}

  Variáveis Globais
    tx    {Uma tupla de X e tx[A] representa o conjunto de valores do atributo
           A numa determinada tupla tx}
    t     {Uma tupla de T e t[C] representa o conjunto de valores do atributo
           C numa determinada tupla t}

Início
  Para todo tx ∈ X faça
  Início
    t ← seleciona_tupla_redundante(T, tx[LA]) {Seleciona a tupla de T que é
                                               redundante à tx[LA]}
    Se t ≠ Null então
      Se tx[DUV] = Null e t[DUV] = '*' então
        Início
          Remove(t, T) {Remove a tupla t da relação resultado T pois ela
                        pertence à região duvidosa}
          Insere(tx[LA], T) {Insere os valores dos atributos especificados em
                             LA na relação resultado T no lugar da tupla t
                             pois tx[LA] pertence à região positiva}
        Fim
    Senão
      Insere(tx[LA], T) {Insere os valores dos atributos especificados em LA na
                         relação resultado T mantendo o valor do atributo DUV}
  Fim
Fim
```

Figura 5.7: Pseudocódigo da operação projeção aproximada.

A função rproject(X, LA), que faz uso da função seleciona_tupla_redundante(relac, tup) descrita na Figura 5.2, executa a operação projeção aproximada e recebe como parâmetros de entrada uma relação aproximada e uma lista com os atributos sobre os quais X será projetada. A função faz um *loop* que percorre todas as tuplas tx, verificando se existe, por meio da função seleciona_tupla_redundante(relac, tup), uma tupla t redundante à tx[LA], na relação T. Se:

1) não existir redundância, tx[LA] é inserida em T, mantendo o valor de tx[DUV];
2) existir redundância, tx[DUV] = *Null* e t[DUV] = "*", t é removida e tx[LA] é inserida em T, com valor *Null* para DUV. Esta substituição em T deve ser feita pois, neste caso, tx pertence à aproximação inferior enquanto que t pertence à região duvidosa.

5.6 A Junção Aproximada

A operação *junção aproximada* é uma operação binária denotada pelo símbolo ⋈ que é aplicada a duas relações aproximadas. O retorno é uma nova relação aproximada onde suas tuplas são combinações de tuplas, em tuplas únicas, das relações argumentos da operação que satisfazem à condição de junção expressa na operação. Esse relacionamento entre as tuplas é feito por meio de atributos comuns às duas relações aproximadas. A operação é definida formalmente na Definição 5.6, está exemplificada no Exemplo 5.7 na seção 5.7 e seu pseudocódigo é mostrado na Figura 5.9.

Definição 5.6: Seja $R_1(A_1, A_2, ..., A_m)$ e $R_2(B_1, B_2, ..., B_n)$, duas relações aproximadas com m e n atributos, respectivamente. O resultado da *junção aproximada* entre R_1 e R_2, denotada por $R_1 \bowtie_{<condição>} R_2$, é a relação aproximada $T(C_1, C_2, ..., C_{m+n})$ de esquema C = AB. A é o conjunto dos atributos de R_1, B é o conjunto dos atributos de R_2, <condição> é uma conjunção de uma ou mais condições na forma **A** = **B**, para **A** ∈ A e **B** ∈ B (**A** é um atributo específico de A e **B** é um atributo específico de B), e t, t_1 e t_2 são tuplas das relações aproximadas T, R_1, R_2, respectivamente. T é definida por suas aproximações inferior e superior dadas por

$$T = \{t \mid \exists\, t_1 \in R_1, t_2 \in R_2 \text{ para } t_1 = t[A], t_2 = t[B]\}$$

$$A_{inf}(T) = \{t \in T \mid t_1[\mathbf{A}] = t_2[\mathbf{B}]\}$$

$$A_{sup}(T) = \{t \in T \mid t_1[\mathbf{A}] \subseteq t_2[\mathbf{B}] \text{ ou } t_2[\mathbf{B}] \subseteq t_1[\mathbf{A}]\}$$

A Definição 5.6 é uma reescrita da definição proposta em [Beauboeuf e Petry 1994], de acordo com a qual, sendo $X(A_1, A_2, ..., A_m)$ e $Y(B_1, B_2, ..., B_n)$ relações aproximadas com m e n atributos, respectivamente, e AB = C, o esquema da relação aproximada resultante T. A junção aproximada, $X \bowtie_{<condição>} Y$, de duas relações X e Y, é uma relação $T(C_1, C_2, ..., C_{m+n})$ onde

$T = \{t \mid \exists\, t_X \in X, t_y \in Y \text{ para } t_x = t(A), t_Y = t(B)\}$, e onde
$t_X(A \cap B) = t_Y(A \cap B)$, para $A_{inf}(T)$
$t_X(A \cap B) \subseteq t_Y(A \cap B)$ ou $t_Y(A \cap B) \subseteq t_X(A \cap B)$ para $A_{sup}(T)$
<condição> é uma conjunção de uma ou mais condições na forma **A** = **B**.

Como pode ser evidenciado, uma vez que a autora não esclarece a notação utilizada, tal definição pode ter várias interpretações. A sua reescrita mostrada na Definição 5.6 tem apenas uma interpretação, e traduz a idéia intuitiva do operador *join* para Bases de Dados Relacionais Aproximadas e é a adotada neste trabalho.

Exemplo 5.1: Sejam os esquemas de relações $A(A_1, A_2, A_3, A_4)$ e $B(B_1, B_2, B_3)$ e as relações R_1 e R_2 de esquemas A e B, respectivamente. A relação aproximada resultante da operação $R_1 \bowtie_{A_3 = B_2} R_2$ terá como esquema a concatenação de A e B, ou seja, o esquema $C(A_1, A_2, A_3, A_4, B_1, B_2, B_3)$. Suas tuplas são resultados da junção das tuplas t_1 e t_2, pertencentes às relações R_1 e R_2, respectivamente, conforme mostrado na Figura 5.8. É importante lembrar que apenas aquelas tuplas que satisfizerem à condição de junção, no caso do exemplo $A_3 = B_2$, é que participam da junção que resulta nas tuplas da relação resultante.

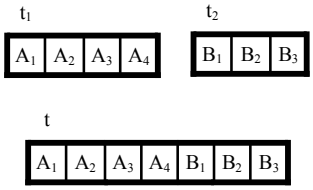


Figura 5.8: A tupla t é resultado da junção das tuplas t_1 e t_2.

A função rjoin(X, Y, A, B), que faz uso da função monta_classe(lista, atrib) descrita na Figura 5.1, executa a operação junção aproximada e tem como parâmetros de entrada duas relações aproximadas e os respectivos atributos para condição de junção. A função faz um *loop* que percorre todas as tuplas tx e para cada uma:

1) constrói a classe de equivalência dos valores do atributo A em tx (CEa);
2) para cada ty:
    a. constrói a classe de equivalência dos valores do atributo B em ty (CEb);
    b. compara as classes de equivalência (CEa e CEb) e se:
        i. CEa = CEb, tx e ty são concatenadas e inseridas em T, com valor *Null* para o atributo DUV, pois a tupla pertence à aproximação inferior;
        ii. CEa $\subset$ CEb ou CEb $\subset$ CEa, tx e ty são concatenadas e inseridas em T, com valor "*" para o atributo DUV, pois a tupla pertence à região duvidosa.

```
Função rjoin(X, Y, A, B) : T
  {Parâmetros de Entrada:
   - X e Y são as relações alvo da operação
   - A e B são os atributos de X e Y, respectivamente, escolhidos para a
     comparação na condição de junção}
  {Parâmetro de Saída:
   - T é a relação resultante da operação expressa por A_inf(T) e duv(T)}

  Variáveis Globais
    CEa  {Classe de equivalência de tx[A]}
    tx    {Uma tupla de X e tx[A] representa o conjunto de valores do atributo A
           numa determinada tupla tx}
    CEb  {Classe de equivalência de ty[B]}
    ty    {Uma tupla de Y e ty[B] representa o conjunto de valores do atributo B
           numa determinada tupla ty}

Início
  Para todo tx ∈ X faça
  Início
    CEa ← monta_classe(tx[A], A)

    Para todo ty ∈ Y faça
    Início
      CEb ← monta_classe(ty[B], B)

      Se CEa = CEb então
         Insere(tx, ty, Null, T) {Concatena as duas tuplas e insere na relação
                                  resultado T com valor Null para o atributo
                                  DUV}
      Senão
        Se CEa ⊂ CEb ou CEb ⊂ CEa então
           Insere(tx, ty, '*', T) {Concatena as duas tuplas e insere na
                                   relação resultado T com valor '*' para
                                   o atributo DUV}
    Fim
  Fim
Fim
```

Figura 5.9: Pseudocódigo da operação junção aproximada.

5.7 Um Exemplo de Uso dos Operadores Relacionais Aproximados

Para exemplificar os Operadores Relacionais Aproximados foi utilizada a Base de Dados Relacional Aproximada LOCADORA, que está representada na Figura 5.10. A sua relação de indiscernibilidade (IND) está representada na Figura 5.11 de maneira simplificada, pois aparecem apenas os valores de atributos de domínios da relação aproximada FILME, e não de toda a base. Essa simplificação se justifica pois os valores de atributos de domínios ausentes na IND não são utilizados nos exemplos e/ou não possuem elementos equivalentes (no caso da Base de Dados Relacional Aproximada LOCADORA), ou seja, sua classe de equivalência é o próprio valor. É importante lembrar que, no caso do atributo GEN_PREFERIDO da relação CLIENTE, as classes de equivalência dos seus valores são as mesmas dos valores do atributo GENERO da relação

FILME, pois possuem o mesmo domínio. As relações aproximadas da base possuem dados referentes a filmes, clientes e locações de filmes e todos os exemplos a seguir fazem uso delas, assim como também da IND.

FILME	CODIGO	TITULO	ATOR_PRINC	GENERO
	2	The Day World Ended	Richard Denning	{Ficção, Mistério}
	3	{The Boat, Das Boat}	{Jurgen Prochbow, J. Prochbow}	{Guerra, Romance}
	4	David and Bathsheba	Gregory Peck	Épico
	13	Evil Under The Sun	Peter Ustinov	Suspense
	15	The Exorcist	Max Von Sydow	{Terror, Suspense}
	18	A Bridge Too Far	James Chan	II Guerra Mundial
	21	The Guns of Navarone	Gregory G. Peck	II Guerra Mundial
	22	Platoon	Tom Berenger	Guerra do Vietnã

CLIENTE	NOME	RG	CPF	GEN PREFERIDO	ENDERECO
	José da Silva	111111111	222222222	{Guerra, Terror}	Rua Barão do Rio Branco, 12
	Maria Cristina de Abreu	333333333	444444444	Romance	Av. São Carlos, 34
	João Carlos Rodrigues	555555555	666666666	Terror	Av. 9 de Julho, 56
	Cláudia Martins	777777777	888888888	{Guerra Civil, Guerra do Vietnã}	Rua Aprígio de Araújo, 78
	Rosana Moreira	999999999	909090909	{Suspense, Terror}	Rua Santa Úrsula, 90

LOCACAO	CODIGO	RG CLIENTE	COD FILME	DATA
	4	111111111	18	09/11/2003
	5	333333333	3	09/11/2003
	6	111111111	21	12/11/2003
	7	777777777	18	12/11/2003
	8	111111111	22	12/11/2003
	9	999999999	15	13/11/2003
	10	333333333	3	15/11/2003

Figura 5.10: Uma instância da Base de Dados Relacional Aproximada LOCADORA.

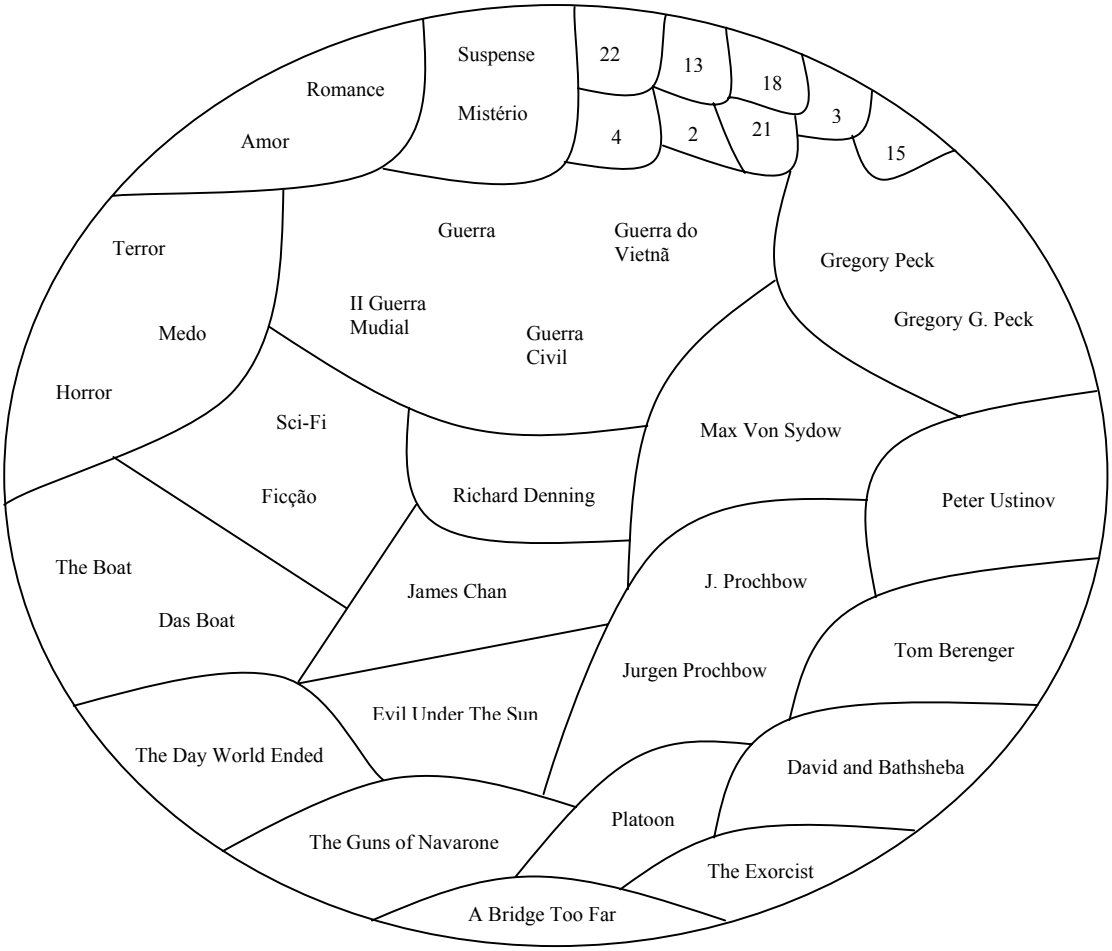


Figura 5.11: Representação simplificada da relação de indiscernibilidade IND associada à Base de Dados Relacional Aproximada LOCADORA.

Os itens Exemplo 5.2 até Exemplo 5.7 mostram situações de uso dos diversos operadores relacionais discutidos nas seções anteriores.

Exemplo 5.2: A operação $\sigma_{GENERO\ =\ [\text{'Suspense'}]}(FILME)$ selecionará todas as tuplas da relação FILME cujo atributo GENERO seja equivalente a 'Suspense' e seu resultado está representado na Figura 5.12.

FILME	CODIGO	TITULO	ATOR_PRINC	GENERO	DUV
	2	The Day World Ended	Richard Denning	{Ficção, Mistério}	*
	13	Evil Under The Sun	Peter Ustinov	Suspense	Null
	15	The Exorcist	Max Von Sydow	{Terror, Suspense}	*

Figura 5.12: O resultado da operação $\sigma_{GENERO\ =\ [\text{'Suspense'}]}(FILME)$.

Conforme comentado anteriormente, por meio do atributo DUV podemos verificar quais tuplas pertencem à aproximação inferior (t[DUV] = *Null*) e quais pertencem à região duvidosa (t[DUV] = <'*'>). No Exemplo 5.2 apenas a tupla t_1[CODIGO] = <13> pertence à aproximação inferior, enquanto que as tuplas t_2[CODIGO] = <2> e t_3[CODIGO] = <15> pertencem à região duvidosa. A tupla t_1 pertence à aproximação inferior porque t_1[GENERO] = <'Suspense'> e, segundo a relação IND, ['Suspense'] = ['Suspense']. Já a tupla t_2, na qual t_2[GENERO] = <{'Ficção', 'Mistério'}>, pertence à região duvidosa pois ['Suspense'] ≠ [{'Ficção', 'Mistério'}] mas ['Suspense'] ⊂ [{'Ficção', 'Mistério'}]. O mesmo ocorre para a tupla t_3.

Exemplo 5.3: Seja a operação $R_1 = \sigma_{GENERO = ['Suspense']}(FILME)$, apresentada no Exemplo 5.2. A operação $\pi_{TITULO,\ ATOR\_PRINC}(R_1)$ projetará as tuplas da relação aproximada R_1 sobre os atributos TITULO e ATOR_PRINC e seu resultado está representado na Figura 5.13.

R_1	TITULO	ATOR_PRINC	DUV
	The Day World Ended	Richard Denning	*
	Evil Under The Sun	Peter Ustinov	Null
	The Exorcist	Max Von Sydow	*

Figura 5.13: O resultado da operação $\pi_{TITULO,\ ATOR\_PRINC}(R_1)$.

A relação aproximada resultante da projeção aproximada, no Exemplo 5.3, contém todas as tuplas da relação aproximada origem pois, com a remoção dos atributos não selecionados para a projeção, não ocorreram redundâncias. Porém, caso ocorressem, estas deveriam ser removidas, sempre mantendo aquelas tuplas que pertencem à aproximação inferior.

Exemplo 5.4: Sejam as operações $R_1 = \sigma_{GENERO = ['Guerra']}(FILME)$ e $R_2 = \sigma_{ATOR\_PRINC = ['Gregory\ G.\ Peck'] \text{ OR } ATOR\_PRINC = ['Jurgen\ Prochbow']}(FILME)$. O resultado da operação $T = R_1 \cup R_2$ está representado na Figura 5.14.

R_1

CODIGO	TITULO	ATOR_PRINC	GENERO	DUV
3	{The Boat, Das Boat}	{Jurgen Prochbow, J. Prochbow}	{Guerra, Romance}	*
18	A Bridge Too Far	James Chan	II Guerra Mundial	Null
21	The Guns of Navarone	Gregory G. Peck	II Guerra Mundial	Null
22	Platoon	Tom Berenger	Guerra do Vietnã	Null

R_2

CODIGO	TITULO	ATOR_PRINC	GENERO	DUV
3	{The Boat, Das Boat}	{Jurgen Prochbow, J. Prochbow}	{Guerra, Romance}	Null
4	David and Bathsheba	Gregory Peck	Épico	Null
21	The Guns of Navarone	Gregory G. Peck	II Guerra Mundial	Null

T

CODIGO	TITULO	ATOR_PRINC	GENERO	DUV
3	{The Boat, Das Boat}	{Jurgen Prochbow, J. Prochbow}	{Guerra, Romance}	Null
18	A Bridge Too Far	James Chan	II Guerra Mundial	Null
21	The Guns of Navarone	Gregory G. Peck	II Guerra Mundial	Null
22	Platoon	Tom Berenger	Guerra do Vietnã	Null
4	David and BathSheba	Gregory Peck	Épico	Null

Figura 5.14: O resultado da operação $T = R_1 \cup R_2$.

No Exemplo 5.4 é importante observar que ocorreu redundância entre as tuplas t_1 = <3, {'The Boat', 'Das Boat'}, {'Jurgen Prochbow', 'J. Prochbow'}, {'Guerra', 'Romance'}, '*'>, de R_1, e t_2 = <3, {'The Boat', 'Das Boat'}, {'Jurgen Prochbow', 'J. Prochbow'}, {'Guerra', 'Romance'}, *Null*>, de R_2, e entre as tuplas t_3 = <21, 'The Guns of Navarone', 'Gregory G. Peck', 'II Guerra Mundial', *Null*>, de R_1, e t_4 = <21, 'The Guns of Navarone', 'Gregory G. Peck', 'II Guerra Mundial', *Null*>, de R_2. No caso das tuplas t_3 e t_4, qualquer uma pode ser mantida, já que ambas pertencem à mesma região, no caso a região positiva ou aproximação inferior. No entanto, dentre as tuplas t_1 e t_2, foi mantida a t_2 pois ela pertence à aproximação inferior enquanto que a t_1 pertence à região duvidosa.

Exemplo 5.5: Sejam as operações $R_1 = \sigma_{GENERO = ['Guerra']}(FILME)$ e $R_2 = \sigma_{ATOR\_PRINC = ['Gregory\ G.\ Peck'] \ OR \ ATOR\_PRINC = ['Jurgen\ Prochbow']}(FILME)$. O resultado da operação $T = R_1 \cap R_2$ está representado na Figura 5.15.

R_1

CODIGO	TITULO	ATOR_PRINC	GENERO	DUV
3	{The Boat, Das Boat}	{Jurgen Prochbow, J. Prochbow}	{Guerra, Romance}	*
18	A Bridge Too Far	James Chan	II Guerra Mundial	Null
21	The Guns of Navarone	Gregory G. Peck	II Guerra Mundial	Null
22	Platoon	Tom Berenger	Guerra do Vietnã	Null

R_2

CODIGO	TITULO	ATOR_PRINC	GENERO	DUV
3	{The Boat, Das Boat}	{Jurgen Prochbow, J. Prochbow}	{Guerra, Romance}	Null
4	David and Bathsheba	Gregory Peck	Épico	Null
21	The Guns of Navarone	Gregory G. Peck	II Guerra Mundial	Null

T

CODIGO	TITULO	ATOR_PRINC	GENERO	DUV
21	The Guns of Navarone	Gregory G. Peck	II Guerra Mundial	Null
3	{The Boat, Das Boat}	{Jurgen Prochbow, J. Prochbow}	{Guerra, Romance}	*

Figura 5.15: O resultado da operação $T = R_1 \cap R_2$.

No Exemplo 5.5 é importante observar o valor do atributo DUV na tupla t = <3, {'The Boat', 'Das Boat'}, {'Jurgen Prochbow', 'J. Prochbow'}, {'Guerra', 'Romance'}, '*'>, que comparece nas duas relações aproximadas R_1 e R_2 (com valores diferentes para o atributo DUV) e por isso está na relação resultante. A tupla t pertence à região duvidosa pois, como também na relação R_1 ela pertence à região duvidosa, t aparece em comum nas aproximações superiores de R_1 e R_2 e não nas aproximações inferiores.

Exemplo 5.6: Sejam as operações $R_1 = \sigma_{GENERO = ['Guerra']}(FILME)$ e $R_2 = \sigma_{ATOR\_PRINC = ['Gregory\ G.\ Peck'] \ OR \ ATOR\_PRINC = ['Jurgen\ Prochbow']}(FILME)$. O resultado das operações $T_1 = R_1 - R_2$ e $T_2 = R_2 - R_1$ estão representados na Figura 5.16.

No Exemplo 5.6 é importante observar que a tupla t[CODIGO, TITULO, ATOR_PRINC, GENERO] = <3, {'The Boat', 'Das Boat'}, {'Jurgen Prochbow', 'J. Prochbow'}, {'Guerra', 'Romance'}>, apesar de comparecer às duas relações, aparece na relação resultante T_2. Isso acontece pois t pertence à aproximação inferior de R_2 e não pertence à aproximação inferior de R_1. Já na relação resultante T_1, t não comparece pois ela pertence à aproximação superior de R_1 e também pertence à aproximação superior de R_2.

R_1

CODIGO	TITULO	ATOR_PRINC	GENERO	DUV
3	{The Boat, Das Boat}	{Jurgen Prochbow, J. Prochbow}	{Guerra, Romance}	*
18	A Bridge Too Far	James Chan	II Guerra Mundial	Null
21	The Guns of Navarone	Gregory G. Peck	II Guerra Mundial	Null
22	Platoon	Tom Berenger	Guerra do Vietnã	Null

R_2

CODIGO	TITULO	ATOR_PRINC	GENERO	DUV
3	{The Boat, Das Boat}	{Jurgen Prochbow, J. Prochbow}	{Guerra, Romance}	Null
4	David and Bathsheba	Gregory Peck	Épico	Null
21	The Guns of Navarone	Gregory G. Peck	II Guerra Mundial	Null

T_1

CODIGO	TITULO	ATOR_PRINC	GENERO	DUV
18	A Bridge Too Far	James Chan	II Guerra Mundial	Null
22	Platoon	Tom Berenger	Guerra do Vietnã	Null

T_2

CODIGO	TITULO	ATOR_PRINC	GENERO	DUV
3	{The Boat, Das Boat}	{Jurgen Prochbow, J. Prochbow}	{Guerra, Romance}	Null
4	David and BathSheba	Gregory Peck	Épico	Null

Figura 5.16: O resultado da operação $T_1 = R_1 - R_2$ e $T_2 = R_2 - R_1$.

Exemplo 5.7: Sejam as operações $R_1 = \pi_{\text{NOME, GEN\_PREFERIDO}}(\text{CLIENTE})$ e $R_2 = \pi_{\text{TITULO, GENERO}}(\text{FILME})$ representadas pela Figura 5.17. O resultado da operação $T = R_1 \bowtie_{\text{GEN\_PREFERIDO} = \text{GENERO}} R_2$ está representado na Figura 5.18.

As tuplas, nas quais as classes de equivalência dos valores dos atributos sendo comparados são iguais, dão origem a tuplas pertencentes à aproximação inferior na relação resultante da junção. Isso acontece, no Exemplo 5.7, com as tuplas t_1 = <'Cláudia Martins', {'Guerra Civil', 'Guerra do Vietnã'}>, de R_1, e t_2 = <'A Bridge Too Far', 'II Guerra Mundial'>, de R_2, pois [{'Guerra Civil', 'Guerra do Vietnã'}] = ['II Guerra Mundial']. Já as tuplas, nas quais as classes de equivalência dos valores dos atributos sendo comparados são diferentes, mas uma das classes está contida na outra, dão origem a tuplas pertencentes à aproximação superior na relação resultante da junção. Isso acontece, por exemplo, com as tuplas t_3 = <'José da Silva', {'Guerra', 'Terror'}>, de R_1, e t_4 = <'A Bridge Too Far', 'II Guerra Mundial'>, de R_2, pois [{'Guerra', 'Terror'}] $\neq$ ['II Guerra Mundial'] mas ['II Guerra Mundial'] $\subset$ [{'Guerra', 'Terror'}].

R_1

NOME	GEN_PREFERIDO
José da Silva	{Guerra, Terror}
Maria Cristina de Abreu	Romance
João Carlos Rodrigues	Terror
Cláudia Martins	{Guerra Civil, Guerra do Vietnã}
Rosana Moreira	{Suspense, Terror}

R_2

TITULO	GENERO
The Day World Ended	{Ficção, Mistério}
{The Boat, Das Boat}	{Guerra, Romance}
David and Bathsheba	Épico
Evil Under The Sun	Suspense
The Exorcist	{Terror, Suspense}
A Bridge Too Far	II Guerra Mundial
The Guns of Navarone	II Guerra Mundial
Platoon	Guerra do Vietnã

Figura 5.17: As relações aproximadas R_1 e R_2.

T

NOME	GEN_PREFERIDO	TITULO	GENERO	DUV
José da Silva	{Guerra, Terror}	A Bridge Too Far	II Guerra Mundial	*
José da Silva	{Guerra, Terror}	The Guns of Navarone	II Guerra Mundial	*
José da Silva	{Guerra, Terror}	Platoon	Guerra do Vietnã	*
Maria Cristina de Abreu	Romance	{The Boat, Das Boat}	{Guerra, Romance}	*
João Carlos Rodrigues	Terror	The Exorcist	{Terror, Suspense}	*
Cláudia Martins	{Guerra Civil, Guerra do Vietnã}	{The Boat, Das Boat}	{Guerra, Romance}	*
Cláudia Martins	{Guerra Civil, Guerra do Vietnã}	A Bridge Too Far	II Guerra Mundial	Null
Cláudia Martins	{Guerra Civil, Guerra do Vietnã}	The Guns of Navarone	II Guerra Mundial	Null
Cláudia Martins	{Guerra Civil, Guerra do Vietnã}	Platoon	Guerra do Vietnã	Null
Rosana Moreira	{Suspense, Terror}	Evil Under The Sun	Suspense	*
Rosana Moreira	{Suspense, Terror}	The Exorcist	{Terror, Suspense}	Null

Figura 5.18: O resultado da operação $T = R_1 \bowtie_{GEN\_PREFERIDO = GENERO} R_2$.

5.8 Considerações Finais

Neste capítulo foram apresentados os Operadores Relacionais Aproximados considerados mais relevantes para a recuperação de informação de uma Base de Dados Relacional Aproximada, por meio de suas definições e pseudocódigos, além de sua funcionalidade por meio de exemplos. No próximo capítulo são definidos formalmente os principais conceitos do Modelo Relacional Aproximado *Fuzzy*. Os Operadores Relacionais Aproximados *Fuzzy*, cujos pseudocódigos foram desenvolvidos e propostos como parte deste trabalho, são também definidos formalmente e têm suas funcionalidades exemplificadas.

CAPÍTULO 6. MODELO RELACIONAL APROXIMADO *FUZZY*

O Modelo Relacional Aproximado *Fuzzy*, proposto em [Beauboeuf et al. 1998] e [Beauboeuf 2004], é uma extensão do Modelo Relacional Aproximado, ao qual foram incorporados conceitos da Teoria de Conjuntos *Fuzzy* (TFC) (ver subseção 2.6.1) a fim de explorar as vantagens de ambas as teorias, i.e., TCA e TCF. Conforme visto na subseção 2.6.2, conjuntos aproximados podem ser representados por uma função de pertinência *fuzzy*, que caracteriza as regiões positiva, negativa e duvidosa (ver Definição 2.5), possibilitando, assim, quantificar a pertinência dos elementos de um conjunto aproximado a essas regiões.

O principal objetivo deste capítulo é apresentar os conceitos do Modelo Relacional Aproximado *Fuzzy* e os Operadores Relacionais Aproximados *Fuzzy*, investigando a contribuição da TCF ao Modelo Relacional Aproximado. Este trabalho de pesquisa contribui para o modelo discutido neste capítulo refinando e padronizando o formalismo utilizado pelos autores, reescrevendo a definição de operadores buscando melhorar a compreensão de sua funcionalidade e evitar ambigüidades na interpretação, além também do desenvolvimento dos pseudocódigos dos operadores e suas implementações.

6.1 Conceitos do Modelo Relacional Aproximado *Fuzzy*

Assim como no Modelo Relacional Aproximado alguns conceitos no Modelo Relacional Aproximado *Fuzzy* apenas tiveram seus nomes alterados e terão o mesmo significado do modelo original quando nada em contrário for mencionado.

Por se tratar de uma extensão, o novo modelo também representa os dados como uma coleção de relações (tabelas) contendo tuplas, no qual as relações são conjuntos de tuplas. Como as tuplas são elementos de um conjunto elas não comparecem duplicadas e nem ordenadas.

Uma tupla t_i de uma Base de Dados Relacional Aproximada *Fuzzy* tem a forma $<d_{i1}, d_{i2}, ..., d_{in}, d_{i\mu}>$, onde d_{ij} é um valor de um determinado domínio $dom(A_j)$, A_j $(1 \leq j \leq n)$ é um atributo que nomeia um papel desempenhado pelo domínio D_j, e $d_{i\mu} \in D_\mu$, onde D_μ é o contradomínio para função de pertinência aproximada (ver subseção 2.6.2), ou seja, o intervalo [0,1]. Uma tupla em uma Base de Dados Relacional Aproximada *Fuzzy* é, pois, uma tupla de uma Base de Dados Relacional Aproximada à qual foi associado um valor de pertinência que representa "o quanto"

essa tupla pertence à relação considerada. No que segue são apresentados os principais conceitos de uma Base de Dados Relacional Aproximada *Fuzzy*.

Definição 6.1: Uma *relação aproximada fuzzy* R é um subconjunto de $P(dom(A_1)) \times P(dom(A_2)) \times \ldots \times P(dom(A_n)) \times D_\mu$.

Definição 6.2: Uma tupla $t_i = <d_{i1}, d_{i2}, ..., d_{in}, d_{i\mu}>$ é chamada de *tupla arbitrária fuzzy* se $t_i \in P(dom(A_1)) \times P(dom(A_2)) \times \ldots \times P(dom(A_n)) \times D_\mu$ e, portanto, $d_{ij} \subseteq dom(A_j)$, j = 1, ..., n, e $d_{i\mu} \in D_\mu = [0, 1]$.

Definição 6.3: Uma tupla $t_i = <d_{i1}, d_{i2}, ..., d_{in}, d_{i\mu}>$ é chamada de *tupla aproximada fuzzy* se $t_i \in R$ e portanto $\in P(dom(A_1)) \times P(dom(A_2)) \times \ldots \times P(dom(A_n)) \times D_\mu$. Cada $d_{ij} \subseteq dom(A_j)$, j = 1, ..., n, e $d_{i\mu} \in D_\mu = [0, 1]$.

A definição de interpretação de uma tupla aproximada *fuzzy* proposta em [Beauboeuf et al. 1998] tem inconsistência, uma vez que estabelece que:

> "uma interpretação $\alpha = <a_1, a_2, ..., a_n, a_\mu>$ de uma tupla aproximada *fuzzy* $t_i = <d_{i1}, d_{i2}, ..., d_{in}, d_{i\mu}>$ é qualquer atribuição de valor tal que $a_j \in d_{ij}$ para todo j."

A inconsistência surge do fato de $d_{i\mu}$ ser um valor e, consequentemente, dizer que $a_\mu \in d_{i\mu}$ está incorreto. Neste trabalho propomos, então, que a definição de interpretação de uma tupla aproximada *fuzzy* seja reescrita como:

Definição 6.4: Uma interpretação $\alpha = <a_1, a_2, ..., a_n, a_\mu>$ de uma tupla aproximada *fuzzy* $t_i = <d_{i1}, d_{i2}, ..., d_{in}, d_{i\mu}>$ é qualquer atribuição de valor tal que $a_j \in d_{ij}$, j = 1, ..., n e $a_\mu = d_{i\mu}$.

O espaço das interpretações é o produto cartesiano dos domínios dos atributos $dom(A_1) \times dom(A_2) \times \ldots \times dom(A_n) \times D_\mu$ porém é limitado, para uma dada relação aproximada *fuzzy* R, ao conjunto das tuplas que são válidas de acordo com a semântica de R.

Exemplo 6.1: Seja o esquema da relação aproximada *fuzzy* FILME(CODIGO, TITULO, ATOR_PRINC, GENERO, MU), onde MU é o atributo que representa o valor de pertinência da tupla, e sejam também a instância da relação aproximada *fuzzy* FILME, apresentada na Figura 6.1, e t_1 uma tupla de FILME para t_1[CODIGO] = <'2'>. As tuplas α_1 = <'2', 'The Day World Ended', 'Richard Denning', 'Ficção', 1> e α_2 = <'2', 'The Day World Ended', 'Richard Denning', 'Mistério', 1> são interpretações de t_1.

FILME	CODIGO	TITULO	ATOR_PRINC	GENERO	MU
	2	The Day World Ended	Richard Denning	{Ficção, Mistério}	1
	3	{The Boat, Das Boat}	{Jurgen Prochbow, J. Prochbow}	{Guerra, Romance}	1
	4	David and Bathsheba	Gregory Peck	Épico	1
	5	O Auto da Compadecida	{Matheus Natchergaele, Selton Mello}	Comédia	0.8
	7	A Sauna	Bruce Gomlevsky	Ficção	0.5
	13	Evil Under The Sun	Peter Ustinov	Suspense	1
	15	The Exorcist	Max Von Sydow	{Terror, Suspense}	1
	18	A Bridge Too Far	James Chan	II Guerra Mundial	1
	21	The Guns of Navarone	Gregory G. Peck	II Guerra Mundial	1
	22	Platoon	Tom Berenger	Guerra do Vietnã	1
	23	A Casa das Sete Mulheres	{Thiago Lacerda, Giovanna Antonelli}	Épico	0.8

Figura 6.1: Uma instância da relação aproximada *fuzzy* FILME.

Para uma determinada relação R, a pertinência é determinada semanticamente. No caso da relação FILME, apresentada na Figura 6.1, o valor de MU, quando diferente de 1, caracteriza as tuplas como "não sendo exatamente filmes". Para uma tupla com valor de MU igual a 0.8, por exemplo, esta tupla representa uma mini-série e para MU igual a 0.5 a tupla representa um curta-metragem.

Conforme mencionado anteriormente, $[d_{ij}]$ denota a classe de equivalência à qual d_{ij} pertence e, se d_{ij} é um conjunto de valores, então sua classe de equivalência é formada pela união das classes de equivalência dos membros de d_{ij}, ou seja, se $d_{ij} = \{c_1, c_2, ..., c_k\}$, então $[d_{ij}] = [c_1] \cup [c_2] \cup \cdots \cup [c_k]$.

Definição 6.5: Duas tuplas aproximadas *fuzzy* $t_i = <d_{i1}, d_{i2}, ..., d_{in}, d_{i\mu}>$ e $t_k = <d_{k1}, d_{k2}, ..., d_{kn}, d_{k\mu}>$, pertencentes a r, são *redundantes* se $[d_{ij}] = [d_{kj}]$ para todo j = 1, ..., n.

Note que a definição de tuplas redundantes não leva em consideração o valor de pertinência associado a cada uma das tuplas da relação.

Ainda em [Beauboeuf et al. 1998] é comentado que: "se uma relação contém apenas aquelas tuplas da aproximação inferior, i.e., aquelas tuplas tendo o valor de pertinência igual a 1, a interpretação α da tupla é única".

A afirmação acima, entretanto, não é sempre verdadeira; só será verdadeira se os valores que descrevem cada um dos atributos multivalorados pertencerem à mesma classe de equivalência. Isso é mostrado no Exemplo 6.2.

Exemplo 6.2: Seja a tupla aproximada *fuzzy* t_1 = <{'a'}, {'b', 'c'}, 1> e suas interpretações α_1 = <'a', 'b', 1> e α_2 = <'a', 'c', 1>. Se os valores de atributo b e c pertencerem à mesma classe de equivalência, [b] = [c] e como obviamente [a] = [a], consequentemente α_1 e α_2 são redundantes e, portanto, t_1 tem apenas uma interpretação. Já se b e c pertencerem a classes de equivalência diferentes, i.e., [b] $\neq$ [c], α_1 e α_2 não são redundantes e, portanto, não existe uma única interpretação para t_1.

Exemplo 6.3: Seja o esquema da relação aproximada *fuzzy* PESSOA(Nome, Idade, MU) e a relação de indiscernibilidade IND da relação aproximada *fuzzy* PESSOA definindo, sobre o atributo IDADE, as seguintes classes de equivalência: IND = {{Criança, Pré-Adolescente}, {Jovem, Adolescente}, {Adulto}, {Idoso, Senhor}}. Sejam também as tuplas t_1 = <'José da Silva', 'Jovem', 1> e t_2 = <'José da Silva', 'Adolescente', 1>. Estas tuplas são redundantes, pois $[d_{11}] = [d_{21}]$ = {José da Silva} e $[d_{12}] = [d_{22}]$ = {Jovem, Adolescente}.

Exemplo 6.4: Seja o esquema PESSOA e a relação de indiscernibilidade IND definidos no Exemplo 6.3 e as tuplas t_1 = <'José da Silva', {'Jovem', 'Adulto'}, 1> e t_2 = <'José da Silva', 'Jovem', 1>. Estas tuplas, apesar de não serem redundantes, pois $[d_{11}] = [d_{21}]$ = {José da Silva} mas $[d_{12}]$ = {Jovem, Adolescente, Adulto} e $[d_{22}]$ = {Jovem, Adolescente}, possuem uma interpretação idêntica entre elas: $\alpha_1 = \alpha_2$ = <'José da Silva', 'Jovem', 1>.

O mecanismo de consulta do Modelo Relacional Aproximado *Fuzzy*, por herança do modelo origem, utiliza nas comparações de valores de atributos a indiscernibilidade ao invés da igualdade

e, para isso, acessa implicitamente a relação de indiscernibilidade IND, da Base de Dados Relacional Aproximada *Fuzzy*. Uma relação aproximada *fuzzy* também é composta por dois conjuntos de tuplas: a aproximação inferior, cujos elementos certamente pertencem à relação, e a aproximação superior, cujos elementos possivelmente pertencem à relação.

Na implementação do Modelo Relacional Aproximado *Fuzzy*, o atributo DUV, usado na implementação do Modelo Relacional Aproximado para identificar a pertinência de uma tupla à região duvidosa, não é mais necessário, uma vez que existe o atributo MU. Por meio de seu valor, no intervalo [0, 1], a região à qual a tupla pertence é identificada: região negativa se MU igual a 0; região positiva se MU igual a 1; e região duvidosa se $0 < MU < 1$.

6.2 Operadores Relacionais Aproximados *Fuzzy*

Os operadores do Modelo Relacional Aproximado *Fuzzy* que se originaram de operações da teoria clássica de conjuntos, no caso a União, a Intersecção e a Diferença, requerem que as relações envolvidas sejam união compatíveis (ver Definição 3.19).

Além dos operadores encontrados no Modelo Relacional Aproximado, redefinidos para o Modelo Relacional Aproximado *Fuzzy*, o novo modelo define alguns operadores alternativos e, para utilizá-los, é necessária uma definição alternativa de redundância, apresentada formalmente na Definição 6.6.

Definição 6.6: Duas sub-tuplas $X = (d_{x1}, d_{x2}, ..., d_{xm})$ e $Y = (d_{y1}, d_{y2}, ..., d_{ym})$ são *aproximadamente redundantes*, $\approx_R$, se para algum $[p] \subseteq [d_{xj}]$ e $[q] \subseteq [d_{yj}]$, $[p] = [q]$ para todo $j = 1, ..., m$.

Exemplo 6.5: Seja o esquema PESSOA e a relação de indiscernibilidade IND definidos no Exemplo 6.3, as tuplas $t_1 =$ <'José da Silva', {'Jovem', 'Adulto'}, 1>, $t_2 =$ <'José da Silva', 'Adolescente', 1> e os valores de atributos a_{11} = 'José da Silva', a_{21} = 'José da Silva', a_{12} = 'Jovem' e a_{22} = 'Adolescente'. Estas tuplas são aproximadamente redundantes, pois:

- $[a_{11}] \subseteq [d_{11}]$, $[a_{21}] \subseteq [d_{21}]$ e $[a_{11}] = [a_{21}] =$ {José da Silva} e
- $[a_{12}] \subseteq [d_{12}]$, $[a_{22}] \subseteq [d_{22}]$ e $[a_{12}] = [a_{22}] =$ {Jovem, Adolescente}.

Os Operadores Relacionais Aproximados *Fuzzy* utilizam os procedimentos auxiliares monta_classe(lista, atrib) (ver Figura 5.1), utilizado para montar classes de equivalência, e seleciona_tupla_redundante(relac, tup) (ver Figura 5.2), utilizado para encontrar uma tupla redundante em uma relação. Estes são exatamente os mesmos procedimentos utilizados pelo Modelo Relacional Aproximado e, portanto, não necessitam ser apresentados novamente.

6.2.1 A União Aproximada *Fuzzy*

A *união aproximada fuzzy* é uma operação binária entre duas relações aproximadas *fuzzy* que resulta em uma outra relação aproximada *fuzzy*, contendo a união das tuplas das duas relações envolvidas e está formalmente definida na Definição 6.7. A Figura 6.2 mostra o pseudocódigo da operação, que está exemplificada no Exemplo 6.8 na subseção 6.2.7.

Seja uma relação aproximada *fuzzy* R e uma tupla t_i pertencente a R. O grau de pertinência de t_i à relação R é denotado por $\mu_R(t_i)$.

Definição 6.7: Dadas duas relações aproximadas *fuzzy* R_1 e R_2, a *união aproximada fuzzy* destas, denotada por $R_1 \cup R_2$, é uma nova relação aproximada *fuzzy*, de mesmo esquema de R_1 e R_2, definida por

$$R_1 \cup R_2 = \{t \mid t \in R_1 \text{ ou } t \in R_2\} \text{ e } \mu_T(t) = \text{MAX}[\mu_{R_1}(t), \mu_{R_2}(t)]$$

A relação aproximada *fuzzy* $R_1 \cup R_2$ resultante contém todas as tuplas pertencentes a R_1 ou R_2 ou a ambas. Caso exista uma tupla de R_1 redundante a uma tupla de R_2, permanece em $R_1 \cup R_2$ aquela com maior valor de pertinência.

A função frunion(X, Y), descrita na Figura 6.2, executa a operação união aproximada *fuzzy* e tem como parâmetros de entrada duas relações aproximadas *fuzzy*. O retorno é a relação aproximada *fuzzy* T, especificada por meio dos conjuntos $A_{inf}(T)$ e duv(T). A função executa um *loop* que percorre todas as tuplas tx da primeira relação inserindo-as na relação resultado T, mantendo o valor do atributo MU de cada tupla. Em seguida é executado um novo *loop*, agora sobre as tuplas de Y, que percorre todas as ty e, por meio da função seleciona_tupla_redundante(relac, tup), verifica a existência de uma tupla t redundante à ty na relação T. Se não existe redundância, ty é inserida em T, mantendo o valor do atributo MU. Se

existe uma t redundante à ty e ty[MU] > t[MU], t é removida de T e ty é inserida em T. Com o término do *loop* de Y, é retornada a relação T.

```
Função frunion(X, Y) : T
  {Parâmetros de Entrada:
   - X e Y são as relações alvo da operação
  {Parâmetro de Saída:
   - T é a relação resultante da operação expressa por A_inf(T) e duv(T)}

  Variáveis Globais
    tx   {Uma tupla de X e tx[A] representa o conjunto de valores de um atributo
          A numa determinada tupla tx}
    ty   {Uma tupla de Y e ty[B] representa o conjunto de valores de um atributo
          B numa determinada tupla ty}
    t    {Uma tupla de T e t[C] representa o conjunto de valores de um atributo
          C numa determinada tupla t}

Início
  Para todo tx ∈ X faça
  Início
    Insere(tx, T) {Insere a tupla tx na relação resultado T mantendo o valor do
                   atributo MU}
  Fim

  Para todo ty ∈ Y faça
  Inicio
    t ← seleciona_tupla_redundante(T, ty) {Seleciona a tupla de T que é
                                           redundante à ty}
    Se t ≠ Null então
      Se (ty[MU] > t[MU]) então
        Início
          Remove(t, T) {Remove a tupla t da relação resultado T pois ela
                        tem grau de pertinência menor que ty}
          Insere(ty, T) {Insere a tupla ty na relação resultado T no lugar da
                         tupla t pois ty tem grau de pertinência maior que t}
        Fim
    Senão
      Insere(ty, T) {Insere a tupla ty na relação resultado}
  Fim
Fim
```

Figura 6.2: Pseudocódigo da operação união aproximada *fuzzy*.

6.2.2 A Intersecção Aproximada *Fuzzy*

A *intersecção aproximada fuzzy* é uma operação binária entre duas relações aproximadas *fuzzy* que resulta em uma nova relação aproximada *fuzzy* contendo as tuplas comuns às duas relações e está formalmente definida na Definição 6.8. A Figura 6.3 mostra o pseudocódigo da operação, que está exemplificada no Exemplo 6.10 na subseção 6.2.7.

Definição 6.8: Dadas duas relações aproximadas *fuzzy* R_1 e R_2, a *intersecção aproximada fuzzy* destas, denotada por $R_1 \cap R_2$, é uma nova relação aproximada *fuzzy*, de mesmo esquema de R_1 e R_2, definida por

$$R_1 \cap R_2 = \{t \mid t \in R_1 \text{ e } t \in R_2\} \text{ e } \mu(t) = \text{MIN}[\mu_{R_1}(t), \mu_{R_2}(t)]$$

A relação aproximada *fuzzy* $R_1 \cap R_2$ resultante contém aquelas tuplas que pertencem a R_1 e também a R_2, sendo que permanece o menor valor de pertinência. Na intersecção aproximada *fuzzy* a comparação das tuplas é baseada na redundância e não na igualdade.

```
Função frintersection (X, Y) : T
  {Parâmetros de Entrada:
   - X e Y são as relações alvo da operação
  {Parâmetro de Saída:
   - T é a relação resultante da operação expressa por A_inf(T) e duv(T)}

  Variáveis Globais
    tx    {Uma tupla de X e tx[A] representa o conjunto de valores de um
           atributo A numa determinada tupla tx}
    ty    {Uma tupla de Y e ty[B] representa o conjunto de valores de um
           atributo B numa determinada tupla ty}

Início
  Para todo tx ∈ X faça
  Inicio
    ty ← seleciona_tupla_redundante(Y, tx) {Seleciona a tupla de Y que é
                                            redundante à tx}

    Se ty ≠ Null então
    Início
      Se (tx[MU] <= ty[MU]) então
        Insere(tx, T) {Insere a tupla tx na relação resultado T pois tx tem
                       menor grau de pertinência}
      Senão
        Insere(ty, T) {Insere a tupla ty na relação resultado T pois ty tem
                       menor grau de pertinência }
    Fim
  Fim
Fim
```

Figura 6.3: Pseudocódigo da operação intersecção aproximada *fuzzy*.

A função frintersection(X, Y), que faz uso da função seleciona_tupla_redundante(relac, tup) descrita na Figura 5.2, executa a operação intersecção aproximada *fuzzy* e tem como parâmetros de entrada duas relações aproximadas *fuzzy*. O retorno é a relação aproximada *fuzzy* T, especificada por meio dos conjuntos $A_{inf}(T)$ e duv(T). A função faz um *loop* que percorre todas

as tx de X verificando se existe, por meio da função seleciona_tupla_redundante(relac, tup), uma tupla ty redundante à tx, na relação Y. Caso exista, o valor do atributo MU é testado: se tx[MU] <= ty[MU] tx é inserida em T, caso contrário ty é inserida em T.

Existem casos nos quais alguma informação pode ser perdida, ou melhor, pode não ser retornada na intersecção aproximada *fuzzy*, pois as tuplas comparadas não são "exatamente" redundantes e por isso foi proposta em [Beauboeuf 2004] uma definição alternativa para a operação, definida formalmente na Definição 6.9.

Definição 6.9: Dadas duas relações aproximadas *fuzzy* R_1 e R_2, a *intersecção aproximada fuzzy* destas, denotada por $R_1 \cap_A R_2$, é uma nova relação aproximada *fuzzy*, de mesmo esquema de R_1 e R_2, definida por

$$R_1 \cap_A R_2 = \{t_1 \mid t_1 \in R_1, e \exists\, t_2 \in R_2 \mid t_1 \approx_R t_2\} \cup \{t_2 \mid t_2 \in R_2, e \exists\, t_1 \in R_1 \mid t_2 \approx_R t_1\} \text{ e}$$
$$\mu(t) = \text{MIN}[\mu_{R_1}(t_1), \mu_{R_2}(t_2)]$$

Nesse caso, a relação aproximada *fuzzy* $R_1 \cap_A R_2$ resultante contém todas as tuplas de R_1 que possuem tuplas aproximadamente redundantes (ver Definição 6.6) em R_2, unidas a todas as tuplas de R_2 que possuem tuplas aproximadamente redundantes em R_1. As redundâncias entre as tuplas, na relação resultante, devem ser removidas, sendo que estas são encontradas por meio do conceito de redundância definido na Definição 6.5 e não do conceito de redundância aproximada.

Para a execução dessa operação é necessário utilizar uma função para selecionar tuplas aproximadamente redundantes, conforme o conceito mostrado na Definição 6.6. A Figura 6.4 apresenta o pseudocódigo da função seleciona_tupla_aproxredundante(relac, tup) e a Figura 6.5 mostra o pseudocódigo da operação alternativa para intersecção aproximada *fuzzy*.

A função frintersection_a(X, Y), que faz uso da função seleciona_tupla_aproxredundante(relac, tup), executa a operação alternativa para intersecção aproximada *fuzzy* e tem como parâmetros de entrada duas relações aproximadas *fuzzy*. O retorno é a relação aproximada *fuzzy* T, especificada por meio dos conjuntos $A_{inf}(T)$ e duv(T). A função faz um *loop* que percorre todas as tx de X verificando se existe, por meio da função seleciona_tupla_aproxredundante(relac, tup), uma tupla ty aproximadamente redundante à tx, na relação Y. Caso exista, tx é inserida em T. A seguir, é feito um *loop* que percorre todas as ty de Y

verificando se existe uma tupla tx aproximadamente redundante à ty, na relação X. Caso exista, é necessário verificar se não existe uma tupla t redundante, mas não aproximadamente redundante, à ty na relação T. Essa verificação é necessária pois, se duas tuplas são redundantes, elas são também aproximadamente redundantes e, assim, poderia gerar redundância nas tuplas da relação resultante T. Se não houver uma tupla t redundante à ty, esta é inserida em T; se houver mas ty[MU] < t[MU], t é removida de T e ty é inserida em T.

```
Função seleciona_tupla_aproxredundante(relac, tup) : t
  {Parâmetros de Entrada:
   - relac é a relação onde será procurada a tupla redundante
   - tup é a tupla que será procurada na relação relac em busca de redundância
  {Parâmetro de Saída:
   - t é a tupla resultado e caso não exista tupla redundante t será Null}

  Variáveis Locais
    tx        {é uma tupla da relação relac}
    An        {Enésimo atributo de tx}
    dxn       {Valor do enésimo atributo de uma tupla tx}
    axi       {iésimo valor de dxn}
    dyn       {Valor do enésimo atributo de uma tupla tup}
    Bn        {Enésimo atributo de tup}
    classeaxi {Classe de equivalência de um valor de dxn}
    classey   {Lista de classes de equivalência de uma tupla tup}
    kont      {contador}
    Achou     {Variável booleana}

Início
  kont ← 0
  classey ← Null

  Para todo dyn ∈ tup faça
  Início
    classey[kont] ← monta_classe(dyn, Bn)
    kont ← kont + 1
  Fim

  Para todo tx ∈ relac faça
  Início
    kont ← 0

    Para todo dxn ∈ tx faça
    Início
      Para todo axi ∈ dxn faça
      Início
        Classeaxi ← monta_classe(axi, An)

        Se Classeaxi ⊆ classey[kont] então
        Início
          Achou ← Verdadeiro
          Interrompe {Interrompe o loop mais interno}
        Fim
        Senão
          Achou ← Falso
      Fim
      Se Achou = Falso então
        Interrompe

      kont ← kont + 1
    Fim

    Se Achou = Verdadeiro então
      Retorna tx {Retorna a tupla redundante encontrada e interrompe a função}
  Fim

  Retorna Null {Retorna Null pois não encontrou tupla redundante}
Fim
```

Figura 6.4: Pseudocódigo da função seleciona_tupla_aproxredundante(relac, tup).

```
Função frintersection_a (X, Y) : T
  {Parâmetros de Entrada:
   - X e Y são as relações alvo da operação
  {Parâmetro de Saída:
   - T é a relação resultante da operação expressa por A_inf(T) e duv(T)}

  Variáveis Globais
    tx    {Uma tupla de X e tx[A] representa o conjunto de valores de um
          atributo A numa determinada tupla tx}
    ty    {Uma tupla de Y e ty[B] representa o conjunto de valores de um
          atributo B numa determinada tupla ty}
    t     {Uma tupla de T e t[C] representa o conjunto de valores de um
          atributo C numa determinada tupla t}

Início
  Para todo tx ∈ X faça
  Inicio
    ty ← seleciona_tupla_aproxredundante(Y, tx){Seleciona a tupla de Y que é
                                                aproximadamente redundante à tx}

    Se ty ≠ Null então
       Insere(tx, T) {Insere a tupla tx na relação resultado T}
  Fim

  Para todo ty ∈ Y faça
  Inicio
    tx ← seleciona_tupla_aproxredundante(X, ty){Seleciona a tupla de X que é
                                                aproximadamente redundante à ty}

    Se tx ≠ Null então
    Início
      t ← seleciona_tupla_redundante(T, ty){Seleciona a tupla de T que é
                                           redundante à ty}

      Se ( t = Null ) então
        Insere(ty, T) {Insere a tupla ty na relação resultado T}
      Senão
        Se ( ty[MU] < t[MU] ) então
        Início
          Remove(t, T) {Remove a tupla t da relação resultado T pois ela
                        tem grau de pertinência maior que ty}

          Insere(ty, T) {Insere a tupla ty na relação resultado T}
        Fim
    Fim
  Fim
Fim
```

Figura 6.5: Pseudocódigo da operação alternativa para a intersecção aproximada *fuzzy*.

6.2.3 A Diferença Aproximada *Fuzzy*

A *diferença aproximada fuzzy* é uma operação binária entre duas relações aproximadas *fuzzy* que resulta em uma nova relação aproximada *fuzzy* contendo as tuplas da primeira relação que não pertencem à segunda relação e está formalmente definida na Definição 6.10. A Figura 6.6 mostra o pseudocódigo da operação, que está exemplificada no Exemplo 6.9 na subseção 6.2.7.

Definição 6.10: Dadas duas relações aproximadas *fuzzy* R_1 e R_2, a *diferença aproximada fuzzy* destas, denotada por $R_1 - R_2$, é uma nova relação aproximada *fuzzy* $R_1 - R_2$, de mesmo esquema de R_1 e R_2, definida por

$$R_1 - R_2 = \{t \mid t \in R_1 \text{ e } t \notin R_2\} \cup \{t \mid t \in R_1 \text{ e } t \in R_2 \text{ e } \mu_{R_1}(t) > \mu_{R_2}(t)\}$$

A relação aproximada *fuzzy* $R_1 - R_2$ resultante contém aquelas tuplas que pertencem a R_1 e que não pertencem a R_2, unidas com aquelas que pertencem a R_1 e também a R_2 onde $\mu_{R_1}(t) > \mu_{R_2}(t)$ sendo que sempre permanece na relação resultado a tupla de R_1. Na diferença aproximada *fuzzy* a comparação das tuplas também é baseada na redundância e não na igualdade.

```
Função frdifference(X, Y) : T
  {Parâmetros de Entrada:
   - X e Y são as relações alvo da operação
  {Parâmetro de Saída:
   - T é a relação resultante da operação expressa por A_inf(T) e duv(T)}

  Variáveis Globais
    tx   {Uma tupla de X e tx[A] representa o conjunto de valores de um atributo
          A numa determinada tupla tx}
    ty   {Uma tupla de Y e ty[B] representa o conjunto de valores de um atributo
          B numa determinada tupla ty}

Início
  Para todo tx ∈ X faça
  Inicio
    ty ← seleciona_tupla_redundante(Y, tx) {Seleciona a tupla de Y que é
                                            redundante à tx}
    Se ty ≠ Null então
      Se (tx[MU] > ty[MU]) então
          Insere(tx, T) {Insere a tupla tx na relação resultado T}
    Senão
      Insere(tx, T) {Insere a tupla tx na relação resultado T}
  Fim
Fim
```

Figura 6.6: Pseudocódigo da operação diferença aproximada *fuzzy*.

A função frdifference(X, Y), que faz uso da função seleciona_tupla_redundante(relac, tup) descrita na Figura 5.2, executa a operação diferença aproximada *fuzzy* e tem como parâmetros de entrada duas relações aproximadas *fuzzy*. O retorno é a relação aproximada *fuzzy* T, especificada

por meio dos conjuntos $A_{inf}(T)$ e duv(T). A função executa um *loop* que percorre todas as tuplas tx verificando, por meio da função seleciona_tupla_redundante(relac, tup), se existe uma tupla ty redundante à tx, na relação Y. Caso não exista ou, se existir, tx[MU] > ty[MU], tx é inserida em T.

6.2.4 A Seleção Aproximada *Fuzzy*

A operação *seleção aproximada fuzzy* é uma operação unária, denotada por σ, que é aplicada a uma relação aproximada *fuzzy*, gerando uma outra relação aproximada *fuzzy* formada por um subconjunto de tuplas da relação alvo que satisfazem a uma condição de seleção, baseada no valor de um atributo especificado. Esta operação é formalmente definida na Definição 6.11, está exemplificada no Exemplo 6.6 na subseção 6.2.7 e seu pseudocódigo é mostrado na Figura 6.7.

Definição 6.11: Seja R um esquema de relação aproximada *fuzzy* e R_1 uma relação aproximada *fuzzy* no esquema R. A *seleção aproximada fuzzy*, $\sigma_{A = \mathbf{a}}(R_1)$, das tuplas de R_1 é uma relação aproximada *fuzzy* T, que tem o mesmo esquema de R_1 onde A é um atributo de R, $\mathbf{a} = \{a_i\}$ e $\mathbf{b} = \{b_j\}$, a_i e $b_j \in$ dom(A), $\cup_x$ denota "a união sobre todo x", t[A] denota o valor do atributo A na tupla t e T é definida por

$$T = \{t \in R_1 \mid \cup_i [a_i] \subseteq \cup_j [b_j]\} \ , \ a_i \in \mathbf{a}, \ b_j \in t[A]$$

onde o valor de pertinência é calculado pela multiplicação do valor de pertinência original por

$$\text{card}(\mathbf{a}) \ / \ \text{card}(\mathbf{b})$$

onde *card*(*x*) é a cardinalidade de *x*.

A relação aproximada *fuzzy* T resultante contém aquelas tuplas de R_1 em que a classe de equivalência de **a** for igual ou estiver contida na classe de equivalência de t[A].

A função frselect(X, A, a), que faz uso da função monta_classe(lista, atrib) descrita na Figura 5.1, executa a operação seleção aproximada *fuzzy* e tem como parâmetros de entrada uma relação aproximada *fuzzy*, um atributo e uma lista de valores, sendo que os dois últimos formam a condição de seleção. Primeiramente é construída a classe de equivalência relativa aos valores de a

(ou seja, de **a** da Definição 6.11) usando a função monta_classe(lista, atrib) e é armazenada na variável CEa. Em seguida a função executa um *loop* que percorre todas as tx e, para cada uma, constrói a classe de equivalência dos valores do atributo A em tx (CEb). As classes de equivalência (CEa e CEb) são comparadas e, se CEa $\subseteq$ CEb, AuxMU é calculado e tx é inserida em T com o valor de AuxMU para o atributo MU.

```
Função frselect(X, A, a) : T
  {Parâmetros de Entrada:
   - X é a relação alvo da operação
   - A é o atributo escolhido para a comparação na condição de seleção
   - a é o conjunto de valores a serem comparados na condição de seleção}
  {Parâmetro de Saída:
   - T é a relação resultante da operação expressa por A_inf(T) e duv(T)}

  Variáveis Globais
    CEa   {Classe de equivalência de a}
    tx    {Uma tupla de X e tx[A] representa o conjunto de valores do
           atributo A numa determinada tupla tx}
    CEb   {Classe de equivalência de tx[A]}
    AuxMU {Variável auxiliar para cálculo de MU}

Início
  CEa ← monta_classe(a, A)

  Para todo tx ∈ X faça
  Início
    CEb ← monta_classe(tx[A], A)

    Se (CEa ⊆ CEb) então
    Início
      AuxMU ← tx[MU]*( card(CEa)/card(CEb) )

      Insere(tx, AuxMU, T) {Insere a tupla t_x na relação resultado T com valor
                            AuxMU para o atributo MU}
    Fim
  Fim
Fim
```

Figura 6.7: Pseudocódigo da operação seleção aproximada *fuzzy*.

6.2.5 A Projeção Aproximada *Fuzzy*

A operação *projeção aproximada fuzzy* é uma operação unária, denotada pelo símbolo π, que é aplicada a uma relação aproximada *fuzzy* retornando uma nova relação aproximada *fuzzy* contendo todas as tuplas da relação argumento da operação, projetadas sobre um subconjunto de atributos especificados. A operação é especificada formalmente na Definição 6.12, está exemplificada no Exemplo 6.7 na subseção 6.2.7 e seu pseudocódigo é mostrado na Figura 6.8.

Definição 6.12: Seja R_1 uma relação aproximada *fuzzy* de esquema R. A operação $\pi_B(R_1)$, retornará uma relação aproximada *fuzzy* T de esquema B, que é um subconjunto de R, onde

$$T = \{t[B] \mid t \in R_1\}$$

Todas as tuplas em T são representadas somente pelos atributos que definem B. A projeção aproximada *fuzzy* de uma relação R_1 sobre um conjunto de atributos B mantém as tuplas que definem R_1 (a menos que ocorram redundâncias, que devem ser tratadas) e não altera o valor de pertinência das tuplas. Na eliminação de tuplas redundantes, é mantida aquela que tiver o maior valor de pertinência.

```
Função frproject(X, LA) : T
  {Parâmetros de Entrada:
   - X é a relação alvo da operação
   - LA é a lista de atributos sobre os quais a relação resultado será projetada
  {Parâmetro de Saída:
   - T é a relação resultante da operação expressa por A_inf(T) e duv(T)}

  Variáveis Globais
    tx    {Uma tupla de X e tx[A] representa o conjunto de valores do atributo
           A numa determinada tupla tx}
    t     {Uma tupla de T e t[C] representa o conjunto de valores do atributo
           C numa determinada tupla t}

Início
  Para todo tx ∈ X faça
  Início
    t ← seleciona_tupla_redundante(T, tx[LA]) {Seleciona a tupla de T que é
                                               redundante à tx[LA]}
    Se t ≠ Null então
      Se (tx[MU] > t[MU]) então
        Início
          Remove(t, T) {Remove a tupla t da relação resultado T pois ela
                        tem menor valor de pertinência}
          Insere(tx[LA], T) {Insere os valores dos atributos especificados em
                             LA na relação resultado T no lugar da tupla t
                             pois tx[LA] possui maior valor de pertinência}
        Fim
    Senão
      Insere(tx[LA], T) {Insere os valores dos atributos especificados em LA na
                         relação resultado T}
  Fim
Fim
```

Figura 6.8: Pseudocódigo da operação projeção aproximada *fuzzy*.

A função frproject(X, LA), que faz uso da função seleciona_tupla_redundante(relac, tup) descrita na Figura 5.2, executa a operação projeção aproximada *fuzzy* e recebe como parâmetros de entrada uma relação aproximada *fuzzy* e uma lista com os atributos sobre os quais X será projetada. A função faz um *loop* que percorre todas as tuplas tx, verificando se existe, por meio da função seleciona_tupla_redundante(relac, tup), uma tupla t redundante à tx[LA], na relação T. Se:

1) não existir redundância, tx[LA] é inserida em T, mantendo o valor de tx[MU];
2) existir redundância, tx[MU] > t[MU], t é removida e tx[LA] é inserida em T. Esta substituição em T deve ser feita pois, neste caso, tx possui maior valor de pertinência que a tupla t.

6.2.6 A Junção Aproximada *Fuzzy*

A operação *junção aproximada fuzzy* é uma operação binária denotada pelo símbolo ⋈ que é aplicada a duas relações aproximadas *fuzzy*. O retorno é uma nova relação aproximada *fuzzy* na qual suas tuplas são combinações de tuplas, em tuplas únicas, das relações argumentos da operação que satisfazem à condição de junção expressa na operação. Esse relacionamento entre as tuplas é feito por meio de atributos comuns às duas relações aproximadas *fuzzy*. A operação é definida formalmente na Definição 6.13, está exemplificada no Exemplo 6.12 na subseção 6.2.7 e seu pseudocódigo é mostrado na Figura 6.9.

Definição 6.13: Seja $R_1(A_1, A_2, ..., A_m)$ e $R_2(B_1, B_2, ..., B_n)$, duas relações aproximadas *fuzzy* com m e n atributos, respectivamente. O resultado da *junção aproximada fuzzy* entre R_1 e R_2, denotada por $R_1 \bowtie_{<condição>} R_2$, é a relação aproximada *fuzzy* $T(C_1, C_2, ..., C_{m+n})$ de esquema C = AB. A é o conjunto dos atributos de R_1, B é o conjunto dos atributos de R_2, <condição> é uma conjunção de uma ou mais condições na forma **A** = **B**, para **A** ∈ A e **B** ∈ B (**A** é um atributo específico de A e **B** é um atributo específico de B), e t, t_1 e t_2 são tuplas das relações aproximadas *fuzzy* T, R_1, R_2, respectivamente, onde T é definida por

$T = \{t \mid \exists\ t_1 \in R_1, t_2 \in R_2$ para $t_1 = t[A], t_2 = t[B]\}$ e onde

(1) $\mu_T(t) = 1$ se $t_1[\mathbf{A}] = t_2[\mathbf{B}]$ ou

(2) $\mu_T(t) = MIN[\mu_{R_1}(t_1), \mu_{R_2}(t_2)]$ se $t_1[\mathbf{A}] \subset t_2[\mathbf{B}]$ ou $t_2[\mathbf{B}] \subset t_1[\mathbf{A}]$

A relação aproximada *fuzzy* resultante T, contém tuplas formadas pela junção de tuplas da primeira relação com tuplas da segunda relação que satisfazem à condição de seleção. Quando o valor do atributo, especificado na condição de seleção, da primeira relação for equivalente ao valor do atributo, também especificado na condição de seleção, da segunda relação, a tupla resultante terá valor de pertinência igual a 1. Se a classe de equivalência dos valores do atributo da primeira relação estiver contida na classe de equivalência dos valores do atributo da segunda relação, ou vice-versa, o valor de pertinência da tupla resultante será o menor entre os valores das duas relações que deram origem a ela.

```
Função frjoin(X, Y, A, B) : T
  {Parâmetros de Entrada:
   - X e Y são as relações alvo da operação
   - A e B são os atributos de X e Y, respectivamente, escolhidos para a
     comparação na condição de junção}
  {Parâmetro de Saída:
   - T é a relação resultante da operação expressa por A_inf(T) e duv(T)}

  Variáveis Globais
    CEa  {Classe de equivalência de tx[A]}
    tx    {Uma tupla de X e tx[A] representa o conjunto de valores do atributo A
           numa determinada tupla tx}
    CEb  {Classe de equivalência de ty[B]}
    ty    {Uma tupla de Y e ty[B] representa o conjunto de valores do atributo B
           numa determinada tupla ty}

Início
  Para todo tx ∈ X faça
  Início
    CEa ← monta_classe(tx[A], A)

    Para todo ty ∈ Y faça
    Início
      CEb ← monta_classe(ty[B], B)

      Se CEa = CEb então
         Insere(tx, ty, 1, T) {Concatena as duas tuplas e insere na relação
                               resultado T com valor 1 para o atributo MU}
      Senão
        Se CEa ⊂ CEb ou CEb ⊂ CEa então
           Insere(tx, ty, min(tx[MU],ty[MU]), T) {Concatena as duas tuplas e
                                                  insere na relação resultado T
                                                  com o menor valor entre tx[MU]
                                                  e ty[MU] para o atributo MU}
    Fim
  Fim
Fim
```

Figura 6.9: Pseudocódigo da operação junção aproximada *fuzzy*.

A função frjoin(X, Y, A, B), que faz uso da função monta_classe(lista, atrib) descrita na Figura 5.1, executa a operação junção aproximada *fuzzy* e tem como parâmetros de entrada duas relações aproximadas *fuzzy* e os respectivos atributos para condição de junção. A função faz um *loop* que percorre todas as tuplas tx e para cada uma:

1) constrói a classe de equivalência dos valores do atributo A em tx (CEa);
2) para cada ty:
    a. constrói a classe de equivalência dos valores do atributo B em ty (CEb);
    b. compara as classes de equivalência (CEa e CEb) e se:
        i. CEa = CEb, tx e ty são concatenadas e inseridas em T, com valor 1 para o atributo MU;
        ii. CEa $\subset$ CEb ou CEb $\subset$ CEa, tx e ty são concatenadas e inseridas em T, com o menor valor entre tx[MU] e ty[MU] para o atributo MU.

Assim como a operação intersecção aproximada *fuzzy*, a junção aproximada *fuzzy*, da maneira como está definida, também pode deixar de retornar informações que estão relacionadas com o que foi solicitado na consulta porque os valores dos atributos comparados não são "exatamente" equivalentes. Para contornar esse problema foi proposta também em [Beauboeuf 2004] uma definição alternativa para a operação, definida formalmente na Definição 6.14. Seu pseudocódigo está descrito na Figura 6.10.

Definição 6.14: Seja R_1 (A_1, A_2, ..., A_m) e R_2(B_1, B_2, ..., B_n), duas relações aproximadas *fuzzy* com m e n atributos, respectivamente. O resultado da *junção aproximada fuzzy* entre R_1 e R_2, denotada por $R_1 \bowtie_{A\ <condição>} R_2$, é a relação aproximada *fuzzy* T(C_1, C_2, ...,C_{m+n}) de esquema C = AB. A é o conjunto dos atributos de R_1, B é o conjunto dos atributos de R_2, <condição> é uma conjunção de uma ou mais condições na forma **A** = **B**, para **A** $\in$ A e **B** $\in$ B (**A** é um atributo específico de A e **B** é um atributo específico de B), e t, t_1 e t_2 são tuplas das relações aproximadas *fuzzy* T, R_1, R_2, respectivamente, onde T é definida por

$T = \{t \mid \exists\ t_1 \in R_1, t_2 \in R_2$ para $t_1 = t[A], t_2 = t[B]\}$ e onde

(1) $\mu_T(t) = 1$ se $t_1[\mathbf{A}] = t_2[\mathbf{B}]$ ou

(2) $\mu_T(t) = MIN[\mu_{R_1}(t_1), \mu_{R_2}(t_2)]$ se t_1[**A**] é aproximadamente redundante a t_2[**B**]

A relação aproximada *fuzzy* resultante T, contém tuplas formadas pela junção de tuplas da primeira relação com tuplas da segunda relação que satisfizeram à condição de seleção. Quando o valor do atributo da primeira relação, especificado na condição de seleção, for equivalente ao valor do atributo da segunda relação, também especificado na condição de seleção, a tupla resultante terá valor de pertinência igual a 1. Se a sub-tupla da primeira relação, formada pelo atributo da primeira relação na condição de seleção, for aproximadamente redundante à sub-tupla da segunda relação, formada pelo atributo da segunda relação na condição de seleção, o valor da pertinência da tupla resultante será o menor entre os valores das duas relações que deram origem a ela.

A função frjoin_a(X, Y, A, B), que faz uso da função monta_classe(lista, atrib) descrita na Figura 5.1 e da função seleciona_tupla_aproxredundante(relac, tup) descrita na Figura 6.4, executa a operação alternativa para a junção aproximada *fuzzy* e tem como parâmetros de entrada duas relações aproximadas *fuzzy* e os respectivos atributos para condição de junção. A função faz um *loop* que percorre todas as tuplas tx e, para cada uma:

1) constrói a classe de equivalência dos valores do atributo A em tx (CEa);
2) faz um *loop* percorrendo todas as tuplas ty e, para cada uma:
    a. constrói a classe de equivalência dos valores do atributo B em ty (CEb);
    b. compara as classes de equivalência (CEa e CEb) e:
        i. se CEa = CEb, tx e ty são concatenadas e inseridas em T, com valor 1 para o atributo MU;
        ii. se CEa ≠ CEb, faz um *loop* percorrendo cada um dos valores de CEa (axi). Caso algum deles esteja contido em CEb (sendo assim CEa aproximadamente redundante à CEb ou CEb aproximadamente redundante à CEa), tx e ty são concatenadas e inseridas em T, com o menor valor entre tx[MU] e ty[MU] para o atributo MU e o *loop* que percorre os valores de CEa é interrompido.

```
Função frjoin_a(X, Y, A, B) : T
  {Parâmetros de Entrada:
   - X e Y são as relações alvo da operação
   - A e B são os atributos de X e Y, respectivamente, escolhidos para a
     comparação na condição de junção}
  {Parâmetro de Saída:
   - T é a relação resultante da operação expressa por A_inf(T) e duv(T)}

  Variáveis Globais
    tx         {Uma tupla de X e tx[A] representa o conjunto de valores do
                atributo A numa determinada tupla tx}
    CEa        {Classe de equivalência de tx[A]}
    axi        {iésimo valor de CEa}
    ty         {Uma tupla de Y e ty[B] representa o conjunto de valores do
                atributo B numa determinada tupla ty}
    CEb        {Classe de equivalência de ty[B]}

Início
  Para todo tx ∈ X faça
  Início
    CEa ← monta_classe(tx[A], A)

    Para todo ty ∈ Y faça
    Início
      CEb ← monta_classe(ty[B], B)

      Se CEa = CEb então
        Insere(tx, ty, 1, T) {Concatena as duas tuplas e insere na relação
                              resultado T com valor 1 para o atributo MU}
      Senão
      Início
        Para todo axi ∈ CEa faça
          Se axi ⊆ CEb então
          Início
            Insere(tx, ty, min(tx[MU],ty[MU]), T) {Concatena as duas tuplas e
                                                   insere na relação resultado T
                                                   com o menor valor entre tx[MU]
                                                   e ty[MU] para o atributo MU}
            Interrompe {Interrompe o loop mais interno}
          Fim
      Fim
    Fim
  Fim
Fim
```

Figura 6.10: Pseudocódigo da operação alternativa para a junção aproximada *fuzzy*.

É importante observar que, da maneira como estão definidas as operações junção aproximada *fuzzy* e sua versão alternativa, podem ocorrer casos onde tuplas da relação resultante recebem valor de pertinência igual a 1 mesmo que as tuplas de origem não satisfaçam exatamente ao que foi solicitado na condição de seleção. Isso ocorre pois o operador MIN é utilizado para o cálculo do valor de pertinência das tuplas resultantes, quando a condição de seleção não é exatamente satisfeita, e, se as tuplas de origem possuem valor de pertinência igual a 1, as resultantes também o terão, já que MIN[1, 1] = 1.

6.2.7 Um Exemplo de Uso dos Operadores Relacionais Aproximados *Fuzzy*

Para exemplificar os Operadores Relacionais Aproximados *Fuzzy* foi utilizada a Base de Dados Relacional Aproximada *Fuzzy* LOCADORA, baseada na Base de Dados Relacional Aproximada LOCADORA do CAPÍTULO 5. A base de dados está representada na Figura 6.11 e sua relação de indiscernibilidade (IND) está representada na Figura 6.12 de maneira simplificada.

FILME

CODIGO	TITULO	ATOR_PRINC	GENERO	MU
2	The Day World Ended	Richard Denning	{Ficção, Mistério}	1
3	{The Boat, Das Boat}	{Jurgen Prochbow, J. Prochbow}	{Guerra, Romance}	1
4	David and Bathsheba	Gregory Peck	Épico	1
5	O Auto da Compadecida	{Matheus Natchergaele, Selton Mello}	Comédia	0.8
7	A Sauna	Bruce Gomlevsky	Ficção	0.5
13	Evil Under The Sun	Peter Ustinov	Suspense	1
15	The Exorcist	Max Von Sydow	{Terror, Suspense}	1
18	A Bridge Too Far	James Chan	II Guerra Mundial	1
21	The Guns of Navarone	Gregory G. Peck	II Guerra Mundial	1
22	Platoon	Tom Berenger	Guerra do Vietnã	1
23	A Casa das Sete Mulheres	{Thiago Lacerda, Giovanna Antonelli}	Épico	0.8

CLIENTE

NOME	RG	CPF	GEN_PREFERIDO	ENDERECO	MU
José da Silva	111111111	222222222	{Guerra, Terror}	Rua Barão do Rio Branco, 12	1
Maria Cristina de Abreu	333333333	444444444	Romance	Av. São Carlos, 34	1
João Carlos Rodrigues	555555555	666666666	Terror	Av. 9 de Julho, 56	1
Cláudia Martins	777777777	888888888	{Guerra Civil, Guerra do Vietnã}	Rua Aprígio de Araújo, 78	1
Rosana Moreira	999999999	909090909	{Suspense, Terror}	Rua Santa Úrsula, 90	1

LOCACAO

CODIGO	RG_CLIENTE	COD_FILME	DATA	MU
4	111111111	18	09/11/2003	1
5	333333333	3	09/11/2003	1
6	111111111	21	12/11/2003	1
7	777777777	18	12/11/2003	1
8	111111111	22	12/11/2003	1
9	999999999	15	13/11/2003	1
10	333333333	3	15/11/2003	1

Figura 6.11: Uma instância da Base de Dados Relacional Aproximada *Fuzzy* LOCADORA.

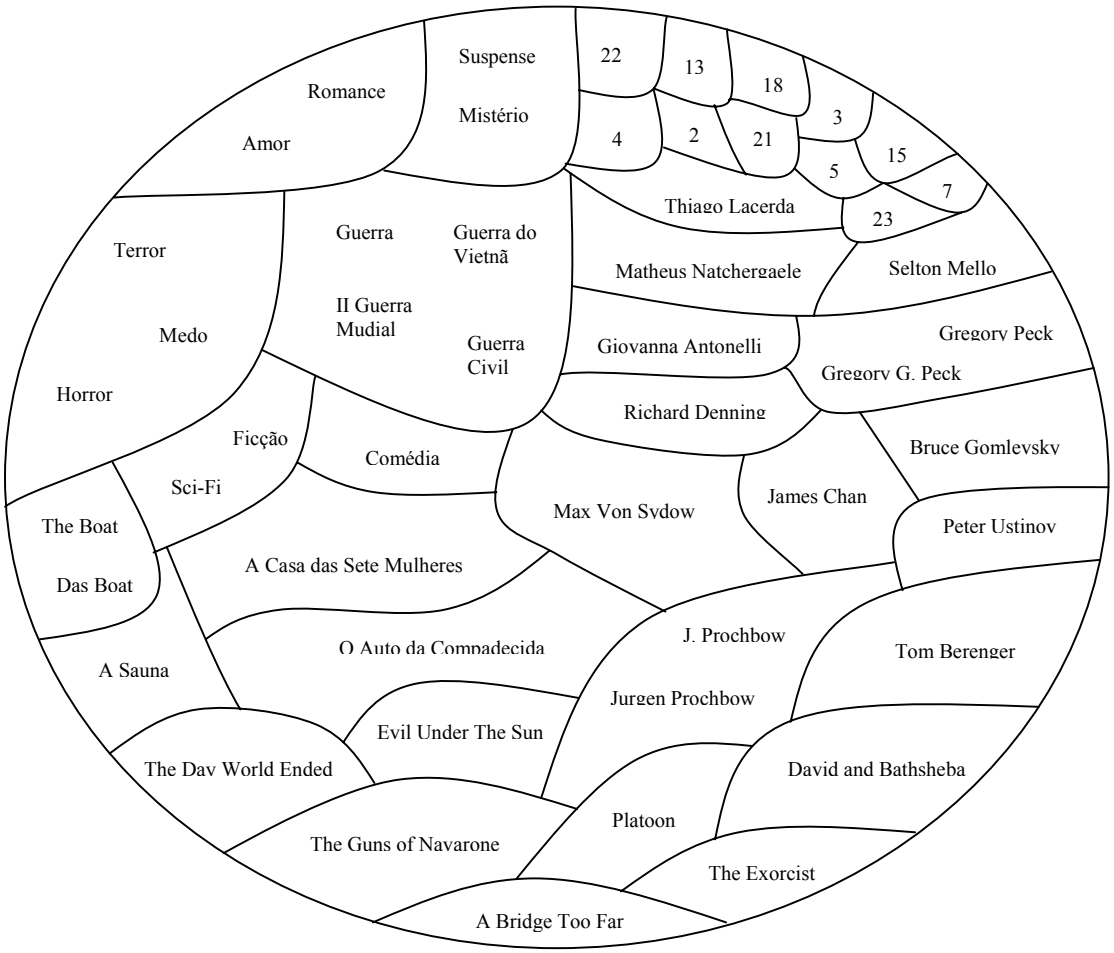


Figura 6.12: Representação simplificada da relação de indiscernibilidade IND da Base de Dados Relacional Aproximada *Fuzzy* LOCADORA.

Exemplo 6.6: A operação $\sigma_{GENERO\ =\ [\text{'Suspense'}]}(FILME)$ selecionará todas as tuplas da relação FILME cujo atributo GENERO seja equivalente à 'Suspense'. O resultado desta operação está representado na Figura 6.13.

FILME	CODIGO	TITULO	ATOR_PRINC	GENERO	MU
	2	The Day World Ended	Richard Denning	{Ficção, Mistério}	0.5
	13	Evil Under The Sun	Peter Ustinov	Suspense	1
	15	The Exorcist	Max Von Sydow	{Terror, Suspense}	0.4

Figura 6.13: O resultado da operação $\sigma_{GENERO\ =\ [\text{'Suspense'}]}(FILME)$.

O cálculo de MU, como no Exemplo 6.6, é feito multiplicando o valor de pertinência original da tupla pela divisão da cardinalidade de 'Suspense' pela cardinalidade dos valores do atributo

GENERO de cada tupla. Sendo t uma tupla de R_1, com t['CODIGO'] = 2 e t[GENERO] = {'Ficção', 'Mistério'}, então card(GENERO) = 2. Assim, como card('Suspense') = 1, t[MU] = 1 * 1/2 = 0.5.

Exemplo 6.7: Seja a operação $R_1 = \sigma_{GENERO = ['Suspense']}(FILME)$, apresentada no Exemplo 6.6. A operação $\pi_{TITULO,\ ATOR\_PRINC}(R_1)$ projetará as tuplas da relação aproximada *fuzzy* R_1 sobre os atributos TITULO e ATOR_PRINC. O resultado desta operação está representado na Figura 6.14.

R_1	TITULO	ATOR_PRINC	MU
	The Day World Ended	Richard Denning	0.5
	Evil Under The Sun	Peter Ustinov	1
	The Exorcist	Max Von Sydow	0.4

Figura 6.14: O resultado da operação $\pi_{TITULO,\ ATOR\_PRINC}(R_1)$.

No caso do Exemplo 6.7 não ocorreu redundância de tuplas devido à remoção dos atributos que não foram selecionados na projeção, mas caso tivesse ocorrido, na eliminação, as tuplas com maior valor de pertinência são sempre mantidas.

Para os exemplos descritos em Exemplo 6.8, Exemplo 6.9 e Exemplo 6.10 serão utilizadas as relações $R_1 = \sigma_{GENERO = ['Guerra']}(FILME)$ e $R_2 = \sigma_{ATOR\_PRINC = ['Gregory\ G.\ Peck'] \text{ OR } ATOR\_PRINC = ['Jurgen\ Prochbow']}(FILME)$, apresentadas na Figura 6.15.

R_1	CODIGO	TITULO	ATOR_PRINC	GENERO	MU
	3	{The Boat, Das Boat}	{Jurgen Prochbow, J. Prochbow}	{Guerra, Romance}	0.67
	18	A Bridge Too Far	James Chan	II Guerra Mundial	1
	21	The Guns of Navarone	Gregory G. Peck	II Guerra Mundial	1
	22	Platoon	Tom Berenger	Guerra do Vietnã	1

R_2	CODIGO	TITULO	ATOR_PRINC	GENERO	MU
	3	{The Boat, Das Boat}	{Jurgen Prochbow, J. Prochbow}	{Guerra, Romance}	1
	4	David and Bathsheba	Gregory Peck	Épico	1
	21	The Guns of Navarone	Gregory G. Peck	II Guerra Mundial	1

Figura 6.15: As relações aproximadas *fuzzy*: $R_1 = \sigma_{GENERO = ['Guerra']}(FILME)$ e $R_2 = \sigma_{ATOR\_PRINC = ['Gregory\ G.\ Peck'] \text{ OR } ATOR\_PRINC = ['Jurgen\ Prochbow']}(FILME)$.

Exemplo 6.8: Sejam as relações R_1 e R_2. O resultado da operação $T = R_1 \cup R_2$ está representado na Figura 6.16.

T	CODIGO	TITULO	ATOR_PRINC	GENERO	MU
	3	{The Boat, Das Boat}	{Jurgen Prochbow, J. Prochbow}	{Guerra, Romance}	1
	18	A Bridge Too Far	James Chan	II Guerra Mundial	1
	21	The Guns of Navarone	Gregory G. Peck	II Guerra Mundial	1
	22	Platoon	Tom Berenger	Guerra do Vietnã	1
	4	David and Bathsheba	Gregory Peck	Épico	1

Figura 6.16: O resultado da operação $T = R_1 \cup R_2$.

É importante observar que na eliminação de redundância, permanece a tupla com maior valor de pertinência, como é o caso, no Exemplo 6.8, das tuplas t_1 = <{'The Boat', 'Das Boat'}, {'Jurgen Prochbow', 'J. Prochbow'}, {'Guerra', 'Romance'}, 0.67>, de R_1 e t_2 = <{'The Boat', 'Das Boat'}, {'Jurgen Prochbow', 'J. Prochbow'}, {'Guerra', 'Romance'}, 1>, de R_2. Na relação resultante, permaneceu a tupla t_1.

Exemplo 6.9: Sejam as relações R_1 e R_2. O resultado das operações $T_1 = R_1 - R_2$ e $T_2 = R_2 - R_1$ estão representados na Figura 6.17.

T_1	CODIGO	TITULO	ATOR_PRINC	GENERO	MU
	18	A Bridge Too Far	James Chan	II Guerra Mundial	1
	22	Platoon	Tom Berenger	Guerra do Vietnã	1

T_2	CODIGO	TITULO	ATOR_PRINC	GENERO	MU
	4	David and Bathsheba	Gregory Peck	Épico	1
	3	{The Boat, Das Boat}	{Jurgen Prochbow, J. Prochbow}	{Guerra, Romance}	1

Figura 6.17: O resultado das operações $T_1 = R_1 - R_2$ e $T_2 = R_2 - R_1$.

É importante observar no Exemplo 6.9, que a tupla t_1 = <{'The Boat', 'Das Boat'}, {'Jurgen Prochbow', 'J. Prochbow'}, {'Guerra', 'Romance'}, 0.67>, de R_1, não comparece na relação resultante T_1 enquanto que a tupla t_2 = <{'The Boat', 'Das Boat'}, {'Jurgen Prochbow', 'J. Prochbow'}, {'Guerra', 'Romance'}, 1>, de R_2, comparece na relação resultante T_2. Isso acontece, pois t_2 possui um valor de pertinência maior do que t_1.

Exemplo 6.10: Sejam as relações R_1 e R_2. O resultado da operação $T = R_1 \cap R_2$ está representado na Figura 6.18.

T	CODIGO	TITULO	ATOR_PRINC	GENERO	MU
	3	{The Boat, Das Boat}	{Jurgen Prochbow, J. Prochbow}	{Guerra, Romance}	0.67
	21	The Guns of Navarone	Gregory G. Peck	II Guerra Mundial	1

Figura 6.18: O resultado da operação $T = R_1 \cap R_2$.

No Exemplo 6.10, é importante observar que permanece na relação resultante as tuplas com menor valor de pertinência, devido ao uso do operador MIN.

Exemplo 6.11: Suponha que o proprietário da locadora de filmes que utiliza a Base de Dados Relacional Aproximada *Fuzzy*, cuja instância está apresentada na Figura 6.11, adquiriu uma antiga concorrente. Os dados do sistema de sua nova locadora foram importados para a base de dados da antiga, mantendo as relações de cada uma em separado. Seja $R_1 = \pi_{NOME, GEN\_PREFERIDO}(CLIENTE)$, onde CLIENTE pertence originalmente à base, e seja CLIENTE_2 a relação de clientes da nova locadora. Os clientes que possuíam cadastro em ambas as locadoras podem ser retornados através das operações $T_1 = R_1 \cap CLIENTE\_2$ e $T_2 = R_1 \cap_A CLIENTE\_2$. As relações envolvidas nessa operação e seu resultado estão representados na Figura 6.19.

Observando os resultados do Exemplo 6.11 é visível o ganho na recuperação de informação com o uso da operação alternativa, como as tuplas $t_1 =$ < 'José da Silva', {'Guerra', 'Terror'}, 1>, de R_1, e $t_2 =$ < 'José da Silva', 'Terror', 0.3>, de CLIENTE_2, que somente comparecem na relação resultante da operação alternativa. Apesar de representarem o mesmo cliente, seriam excluídas do retorno de informação ao usuário por causa de uma pequena diferença em um dos atributos.

R_1	NOME	GEN_PREFERIDO	MU
	José da Silva	{Guerra, Terror}	1
	Maria Cristina de Abreu	Romance	1
	João Carlos Rodrigues	Terror	1
	Cláudia Martins	{Guerra Civil, Guerra do Vietnã}	1
	Rosana Moreira	{Suspense, Terror}	1

CLIENTE_2	NOME	GEN_PREFERIDO	MU
	Claudecir Aragão	{Romance, Ficção}	1
	José da Silva	Terror	0.3
	Rosana Moreira	{Suspense, Terror}	0.6
	Douglas Scheloto	Ficção	1
	João Carlos Rodrigues	Terror	1

T_1	NOME	GEN_PREFERIDO	MU
	João Carlos Rodrigues	Terror	1
	Rosana Moreira	{Suspense, Terror}	0.6

T_2	NOME	GEN_PREFERIDO	MU
	José da Silva	{Guerra, Terror}	0.3
	João Carlos Rodrigues	Terror	1
	Rosana Moreira	{Suspense, Terror}	0.6
	José da Silva	Terror	0.3

Figura 6.19: As relações R1, CLIENTE_2 e o resultado da operação $R_1 \cap_A$ CLIENTE_2.

Exemplo 6.12: Sejam as operações $R_1 = \pi_{NOME,\ GEN\_PREFERIDO}(CLIENTE)$ e $R_2 = \pi_{TITULO,\ GENERO}\ (\ \sigma_{GENERO\ =\ ['Suspense'] \text{ OR } GENERO\ =\ ['Guerra']}(FILME)\ )$ representadas pela Figura 6.20. O resultado das operações $T_1 = R_1 \bowtie_{GEN\_PREFERIDO\ =\ GENERO} R_2$ e $T_2 = R_1 \bowtie_{A\ GEN\_PREFERIDO\ =\ GENERO} R_2$ estão representados na Figura 6.21 e na Figura 6.22, respectivamente.

R_1	NOME	GEN_PREFERIDO	MU
	José da Silva	{Guerra, Terror}	1
	Maria Cristina de Abreu	Romance	1
	João Carlos Rodrigues	Terror	1
	Cláudia Martins	{Guerra Civil, Guerra do Vietnã}	1
	Rosana Moreira	{Suspense, Terror}	1

R_2	TITULO	GENERO	MU
	The Day World Ended	{Ficção, Mistério}	0.5
	{The Boat, Das Boat}	{Guerra, Romance}	0.67
	Evil Under The Sun	Suspense	1
	The Exorcist	{Terror, Suspense}	0.4
	A Bridge Too Far	II Guerra Mundial	1
	The Guns of Navarone	II Guerra Mundial	1
	Platoon	Guerra do Vietnã	1

Figura 6.20: As relações $R_1 = \pi_{NOME,\ GEN\_PREFERIDO}(CLIENTE)$ e $R_2 = \pi_{TITULO,\ GENERO}(\sigma_{GENERO\ =\ ['Suspense'] \text{ OR } GENERO\ =\ ['Guerra']}FILME)$.

T_1	NOME	GEN_PREFERIDO	TITULO	GENERO	MU
	José da Silva	{Guerra, Terror}	A Bridge Too Far	II Guerra Mundial	1
	José da Silva	{Guerra, Terror}	The Guns of Navarone	II Guerra Mundial	1
	José da Silva	{Guerra, Terror}	Platoon	Guerra do Vietnã	1
	Maria Cristina de Abreu	Romance	{The Boat, Das Boat}	{Guerra, Romance}	0.67
	João Carlos Rodrigues	Terror	The Exorcist	{Terror, Suspense}	0.4
	Cláudia Martins	{Guerra Civil, Guerra do Vietnã}	{The Boat, Das Boat}	{Guerra, Romance}	0.67
	Cláudia Martins	{Guerra Civil, Guerra do Vietnã}	A Bridge Too Far	II Guerra Mundial	1
	Cláudia Martins	{Guerra Civil, Guerra do Vietnã}	The Guns of Navarone	II Guerra Mundial	1
	Cláudia Martins	{Guerra Civil, Guerra do Vietnã}	Platoon	Guerra do Vietnã	1
	Rosana Moreira	{Suspense, Terror}	Evil Under The Sun	Suspense	1
	Rosana Moreira	{Suspense, Terror}	The Exorcist	{Terror, Suspense}	1

Figura 6.21: O resultado da operação $T_1 = R_1 \bowtie_{GEN\_PREFERIDO = GENERO} R_2$.

T_2	NOME	GEN_PREFERIDO	TITULO	GENERO	MU
	José da Silva	{Guerra, Terror}	{The Boat, Das Boat}	{Guerra, Romance}	0.67
	José da Silva	{Guerra, Terror}	The Exorcist	{Terror, Suspense}	0.4
	José da Silva	{Guerra, Terror}	A Bridge Too Far	II Guerra Mundial	1
	José da Silva	{Guerra, Terror}	The Guns of Navarone	II Guerra Mundial	1
	José da Silva	{Guerra, Terror}	Platoon	Guerra do Vietnã	1
	Maria Cristina de Abreu	Romance	{The Boat, Das Boat}	{Guerra, Romance}	0.67
	João Carlos Rodrigues	Terror	The Exorcist	{Terror, Suspense}	0.4
	Cláudia Martins	{Guerra Civil, Guerra do Vietnã}	{The Boat, Das Boat}	{Guerra, Romance}	0.67
	Cláudia Martins	{Guerra Civil, Guerra do Vietnã}	A Bridge Too Far	II Guerra Mundial	1
	Cláudia Martins	{Guerra Civil, Guerra do Vietnã}	The Guns of Navarone	II Guerra Mundial	1
	Cláudia Martins	{Guerra Civil, Guerra do Vietnã}	Platoon	Guerra do Vietnã	1
	Rosana Moreira	{Suspense, Terror}	The Day World Ended	{Ficção, Mistério}	0.5
	Rosana Moreira	{Suspense, Terror}	Evil Under The Sun	Suspense	1
	Rosana Moreira	{Suspense, Terror}	The Exorcist	{Terror, Suspense}	1

Figura 6.22: O resultado da operação $T_2 = R_1 \bowtie_{A\ GEN\_PREFERIDO = GENERO} R_2$.

O primeiro fato a se observar no Exemplo 6.12 é a ocorrência da situação citada anteriormente, na qual a condição de seleção não é exatamente satisfeita e, ainda assim, o grau de pertinência da tupla resultante é igual a 1. Isso pode ser observado na junção das tuplas $t_1 = <$‘ Rosana Moreira’, {‘Suspense’, ‘Terror’}, 1>, de R_1, e $t_2 = <$‘ Evil Under The Sun’, ‘Suspense’, 1>, de R_2, já que [{‘Suspense’, ‘Terror’}] $\neq$ [‘Suspense’] mas [‘Suspense’] $\subset$ [{‘Suspense’, ‘Terror’}].

Outro fato a se observar é, novamente, o ganho na recuperação de informação com a utilização da operação alternativa. Esse ganho é visível pela comparação na quantidade de tuplas retornadas pelas duas operações, a junção aproximada *fuzzy* e sua versão alternativa.

6.3 Considerações Finais

Este capítulo apresentou o Modelo Relacional Aproximado *Fuzzy* e seus conceitos fundamentais, além das definições, pseudocódigos e exemplos dos Operadores Relacionais Aproximados *Fuzzy* considerados mais relevantes. No próximo capítulo é proposto um sistema híbrido, chamado ROUGH-ID3, formado por um sistema que implementa os Operadores Relacionais Aproximados e Aproximados *Fuzzy* combinados a um sistema que implementa o método simbólico de aprendizado ID3, visando a extração de conhecimento em Bases de Dados Relacionais Aproximadas e Aproximadas *Fuzzy*.

CAPÍTULO 7. BASE DE DADOS RELACIONAL APROXIMADA E EXTRAÇÃO DE CONHECIMENTO

O objetivo principal deste capítulo é apresentar um sistema híbrido formado pelos Operadores Relacionais Aproximados e Aproximados *Fuzzy*, e o método simbólico de aprendizado conhecido como ID3, buscando investigar a colaboração dessa combinação de recursos para a extração de conhecimento a partir de Bases de Dados Relacionais Aproximadas e Aproximadas *Fuzzy*.

Antes, porém, de apresentar este sistema híbrido, chamado ROUGH-ID3, é fornecido o embasamento teórico sobre o ID3 procurando exemplificar seu funcionamento.

7.1 O método ID3

O ID3 [Quinlan 1986] é um algoritmo de aprendizado de máquina da família de algoritmos conhecida como TDIDT (*Top Down Induction of Decision Trees*). Assim como os outros elementos dessa família, o ID3 utiliza a técnica de indução de árvores de decisão a partir de um conjunto de instâncias; é o algoritmo mais conhecido e difundido para a indução de árvores de decisão e muitos dos algoritmos da família TDIDT são baseados nele. O ID3 representa o conceito induzido na forma de árvores de decisão.

7.1.1 Árvores de Decisão como representação de conhecimento

Conforme citado em [Quinlan 1986] uma árvore de decisão é uma forma relativamente simples de representação de conhecimento, que não possui o poder expressivo de uma rede semântica, por exemplo, mas que é capaz de representar soluções de problemas complexos. Pela sua simplicidade, as metodologias de aprendizado utilizadas pela família TDIDT são consideravelmente menos complexas do que aquelas utilizadas em sistemas que podem expressar seus resultados em uma linguagem mais poderosa.

Uma árvore de decisão é composta por nós folha (ou nós resposta) – cada um deste tipo de nó representa uma classe – e nós raiz (ou nós de decisão) – cada um deste tipo de nó representa um atributo de teste, com um ramo para cada possível valor desse atributo ligando-o a outro nó (folha ou raiz).

Exemplo 7.1: O exemplo que segue foi retirado de [Mitchell 1997] e utiliza como base o conjunto de treinamento apresentado na Tabela 7.1, cujos atributos caracterizam duas classes (sim e não) associadas ao conceito "dia adequado para jogar tênis". Os atributos e seus possíveis valores são:

- APARENCIA – ensolarado, nublado e chuvoso;
- TEMPERATURA – baixa, média e alta;
- UMIDADE – alta e normal;
- VENTO – fraco e forte;
- JOGAR_TENIS (atributo classe) – sim e não.

Uma das possíveis árvores de decisão que classificam corretamente cada instância no conjunto de treinamento da Tabela 7.1 é apresentada na Figura 7.1.

Tabela 7.1: Conjunto de treinamento do conceito "dia adequado para jogar tênis".

Nº	APARÊNCIA	TEMPERATURA	UMIDADE	VENTO	JOGAR TÊNIS
1	ensolarado	alta	alta	fraco	não
2	ensolarado	alta	alta	forte	não
3	nublado	alta	alta	fraco	sim
4	chuvoso	média	alta	fraco	sim
5	chuvoso	baixa	normal	fraco	sim
6	chuvoso	baixa	normal	forte	não
7	nublado	baixa	normal	forte	sim
8	ensolarado	média	alta	fraco	não
9	ensolarado	baixa	normal	fraco	sim
10	chuvoso	média	normal	fraco	sim
11	ensolarado	média	normal	forte	sim
12	nublado	média	alta	forte	sim
13	nublado	alta	normal	fraco	sim
14	chuvoso	média	alta	forte	não

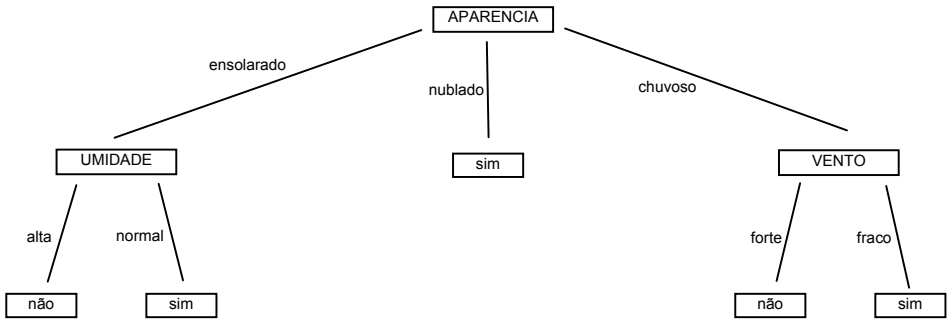


Figura 7.1: Uma árvore de decisão que classifica corretamente as instâncias do conjunto de treinamento da Tabela 7.1.

A classificação de uma instância, por meio de uma árvore de decisão da família TDIDT, começa da raiz da árvore, no caso da Figura 7.1 o atributo APARENCIA, e segue o ramo apropriado, de acordo com o valor do atributo, até um outro nó. O processo continua até que um nó folha seja encontrado, indicando a classe à qual a instância pertence. Por exemplo, se o valor do atributo APARENCIA for igual a "chuvoso" verifica-se o valor de VENTO. Se for "fraco" a classe dessa instância é "sim". Observa-se que o atributo TEMPERATURA não aparece na árvore, apesar de descrever as instâncias. Isso acontece devido ao método de avaliação ou função de avaliação utilizado que, durante a construção da árvore, pode descartar um atributo.

A construção de uma árvore da família TDIDT é um processo iterativo que parte da raiz para as folhas (por isso o nome *Top Down*) e a cada passo é necessário determinar o atributo mais relevante (aquele que melhor agrupa as instâncias segundo o valor das suas respectivas classes) do conjunto de treinamento, sendo que isso é feito por meio de uma função de avaliação. A função de avaliação é uma das diferenças entre os algoritmos de aprendizado de árvores de decisão; a função de avaliação utilizada pelo ID3 é baseada na entropia. O conceito de entropia e os passos para se determinar o atributo mais relevante de um conjunto de instâncias são descritos a seguir (na subseção 7.1.2). A subseção 7.1.3 descreve o processo de construção de uma árvore de decisão utilizando algoritmo do ID3, por meio do seu pseudocódigo.

7.1.2 Entropia e Ganho de Informação

A entropia é uma medida utilizada em teoria da informação e é interpretada como a quantidade de informação contida numa mensagem; quanto maior o valor da entropia, maior a incerteza com relação ao conteúdo da mensagem [Shannon 1948]. Assim, a entropia de um conjunto de instâncias é a quantidade de informação necessária para a classificação de uma instância qualquer desse conjunto com relação à sua pertinência a uma determinada classe. Ela atinge o valor máximo, 1, quando as classes têm a mesma probabilidade de ocorrer e atinge o valor mínimo, 0, quando todas as instâncias pertencem à mesma classe. Sendo assim, pode-se interpretar que quanto maior a entropia de um determinado conjunto de instâncias maior a incerteza na classificação de uma instância desse conjunto com relação à sua pertinência a uma determinada classe.

Seja C uma coleção de instâncias, C_1, C_2, ..., C_n as classes em que uma instância pode ser classificada e $p(C_i)$ a probabilidade de ocorrência de uma instância na classe C_i, para $1 \leq i \leq n$. A entropia ou H(C), utilizada pelo ID3, é dada por

$$H(C) = -\sum_{i=1}^{n} p(C_i)\log_2 p(C_i) \tag{6.1}$$

Exemplo 7.2: A entropia do conjunto da Tabela 7.1 é calculada da seguinte forma:

$$\begin{aligned} H(C) &= -p(\text{sim}) * \log_2 p(\text{sim}) - p(\text{não}) * \log_2 p(\text{não}) \\ &= -(9/14) * \log_2 (9/14) - (5/14) * \log_2 (5/14) \\ &= -(0{,}6429) * (-0{,}6374) - (0{,}3571) * (-1{,}4854) \\ &= 0{,}4098 + 0{,}5305 \\ &= 0{,}9403 \end{aligned}$$

O valor encontrado revela uma grande incerteza na classificação das instâncias do conjunto avaliado, no caso o conjunto de treinamento da Tabela 7.1, visto que a quantidade de informação necessária para a sua classificação ou entropia é muito alta (bem próxima do valor máximo 1).

Para construir uma árvore de decisão o ID3 busca, dentre os atributos do conjunto de treinamento, aquele que tiver o menor valor de entropia. Para calcular a entropia relativa a um determinado atributo é necessário dividir o conjunto de treinamento em subconjuntos, agrupando as instâncias que possuem o mesmo valor para o atributo em questão, conforme mostra a Tabela 7.2.

Tabela 7.2: Subconjuntos do conjunto de treinamento da **Tabela 7.1** para o atributo VENTO.

VENTO	JOGAR TÊNIS
forte	não
forte	não
forte	sim
forte	sim
forte	sim
forte	não
fraco	não
fraco	sim
fraco	sim
fraco	sim
fraco	não
fraco	sim
fraco	sim
fraco	sim

Seja A um atributo e a_1, a_2, ..., a_m, os valores de A, o cálculo da entropia de cada um dos subconjuntos do conjunto de treinamento, em relação ao atributo A é dado por

$$H(C \mid A = a_j) = -\sum_{i=1}^{n} p(C_i \mid A = a_j) \log_2 p(C_i \mid A = a_j) \tag{6.2}$$

onde $p(C_i \mid A = a_j)$ é a probabilidade do valor da classe ser C_i quando A é igual a a_j.

A entropia de todo o conjunto de instâncias relativo ao atributo A, é calculada pela expressão

$$H(C \mid A) = \sum_{j=1}^{m} p(a_j) H(C \mid A = a_j) \tag{6.3}$$

Exemplo 7.3: Seja o atributo VENTO, cujos subconjuntos de treinamento estão representados na Tabela 7.2. Calcula-se a entropia relativa ao atributo VENTO da forma a seguir:

- Cálculo da entropia de cada subconjunto de instâncias do atributo

$$\begin{aligned} H(C \mid \text{VENTO} = \text{forte}) &= -p(\text{sim} \mid \text{VENTO} = \text{forte}) * \log_2 p(\text{sim} \mid \text{VENTO} = \text{forte}) \\ &\quad - p(\text{não} \mid \text{VENTO} = \text{forte}) * \log_2 p(\text{não} \mid \text{VENTO} = \text{forte}) \\ &= -(3/6) * \log_2 (3/6) - (3/6) * \log_2 (3/6) \\ &= -(0{,}5) * (-1) - (0{,}5) * (-1) \\ &= 0{,}5 + 0{,}5 \\ &= 1 \end{aligned}$$

$$\begin{aligned} H(C \mid \text{VENTO} = \text{fraco}) &= -p(\text{sim} \mid \text{VENTO} = \text{fraco}) * \log_2 p(\text{sim} \mid \text{VENTO} = \text{fraco}) \\ &\quad - p(\text{não} \mid \text{VENTO} = \text{fraco}) * \log_2 p(\text{não} \mid \text{VENTO} = \text{fraco}) \\ &= -(6/8) * \log_2 (6/8) - (2/8) * \log_2 (2/8) \\ &= -(0{,}75) * (-0{,}415) - (0{,}25) * (-2) \\ &= 0{,}3113 + 0{,}5 \\ &= 0{,}8113 \end{aligned}$$

- Cálculo da entropia total relativa ao atributo

$$\begin{aligned} H(C \mid \text{VENTO}) &= (6/14) * (1) + (8/14) * (0{,}8113) \\ &= 0{,}8922 \end{aligned}$$

O valor da entropia total para o atributo VENTO mostra que ele não parece ser o atributo mais adequado para classificar as instâncias do conjunto de treinamento, já que sua entropia está

próxima de 1 porém, é necessário que esse cálculo seja realizado para todos os atributos para se verificar qual possui o menor valor de entropia.

O ganho de informação é a função de avaliação do algoritmo ID3 e utiliza a entropia para avaliar cada atributo do conjunto na classificação das instâncias. Isso é feito medindo-se a redução da entropia, que acontece quando o conjunto é particionado usando como critério os valores de um determinado atributo. O ganho de informação é calculado pela equação (6.4), na qual H(C) é descrito pela equação (6.1) e H(C | A) pela equação (6.3).

$$\text{Ganho}(C \mid A) = H(C) - H(C \mid A) \tag{6.4}$$

Exemplo 7.4: O ganho do conjunto de treinamento representado na Tabela 7.1, para a classe JOGAR_TENIS e o atributo VENTO é calculado como mostrado a seguir:

$$\begin{aligned} \text{Ganho}(C \mid \text{VENTO}) &= H(C) - H(C \mid \text{VENTO}) \\ &= 0{,}9403 - 0{,}8922 \\ &= 0{,}0481 \end{aligned}$$

Esse cálculo é feito para cada um dos atributos do conjunto de treinamento e aquele que possuir o menor valor de entropia e, consequentemente, o maior ganho de informação, é eleito o atributo mais relevante do conjunto.

7.1.3 Pseudocódigo do ID3

O pseudocódigo do algoritmo ID3, apresentado na Figura 7.2, foi retirado de [Mitchell 1997]. A função ID3(Exemplos, Atributo_Alvo, Atributos) recebe como parâmetros o conjunto de treinamento, o atributo alvo (normalmente o atributo classe, mas também pode ser qualquer atributo que participe da definição das instâncias de treinamento) e a lista de atributos existentes no conjunto de treinamento. Primeiramente são verificadas as condições de parada, que são os casos nos quais todos os exemplos pertencem a uma mesma classe (sim ou não) – retorna um nó com o rótulo do valor da classe – ou ainda quando a lista de atributos terminou – retorna um nó com rótulo igual ao valor de Atributo_Alvo mais comum em Exemplos. Se as condições de parada não forem satisfeitas o ID3 escolhe o atributo, pertencente a Atributos, que melhor classifica Exemplos e cria um nó (Node) para ele. Essa escolha é baseada no ganho de

informação e o atributo que possui o maior valor é identificado. Faz-se um *loop* (laço) que percorre todos os possíveis valores (ai) do atributo escolhido (A). A cada ciclo é criado um novo ramo a partir de Node com o rótulo igual ao valor do ai atual e um subconjunto (Exemplos_ai) de Exemplos, em que o atributo A é igual a ai, é criado. Se Exemplos_ai está vazio então o algoritmo retorna o nó com rótulo igual ao valor de Atributo_Alvo mais comum em Exemplos, senão, adiciona ao novo ramo uma árvore de decisão, encontrada recursivamente, passando como parâmetros Exemplos_ai, como conjunto de exemplos, Atributo_Alvo, e Atributos menos o atributo A, para que este não compareça nos ramos a partir do seu próprio nó de decisão.

```
ID3(Exemplos, Atributo_Alvo,Atributos)
  {Exemplos é o conjunto de treinamento. Atributo_Alvo pode ser o atributo de
   classe (por exemplo "Jogar Tênis"), ou qualquer outro atributo que
   participa da definição das instâncias de treinamento. Atributos é a lista de
   atributos existentes no conjunto de treinamento.}

Se todos os Exemplos possuem o valor de Atributo_Alvo = sim
  Então Retorne um nó com o rótulo = sim
Se todos os Exemplos possuem o valor de Atributo_Alvo = não
  Então Retorne um nó com o rótulo = não
Se Atributos está vazio
  Então Retorne um nó com o rótulo = valor de Atributo_Alvo mais comum em
        Exemplos
Senão
  A ← o atributo pertencente a Atributos que melhor classifica Exemplos
  Crie um nó raiz Node ← A { Node é um nó de decisão para o atributo A }
  Para cada possível valor, ai, de A
    Crie um novo ramo a partir de Node com o rótulo = ai
    Seja Exemplos_ai o subconjunto de Exemplos que têm o atributo A = ai
    Se Exemplos_ai está vazio
      Então adicione ao ramo em questão um nó com o rótulo = valor de
            Atributo_Alvo mais comum em Exemplos.
    Senão adicione ao novo ramo a seguinte árvore
      ID3(Exemplos_ai,Atributo_Alvo,Atributos-{A})

Retorne Node
```

Figura 7.2: Pseudocódigo do ID3.

7.2 O Sistema Híbrido ROUGH-ID3

Esta seção descreve as principais características do sistema híbrido ROUGH-ID3, que foi proposto e desenvolvido durante este trabalho de pesquisa com vistas à extração de conhecimento a partir de uma Base de Dados Relacional Aproximada ou Aproximada *Fuzzy*. O ROUGH-ID3 implementa uma articulação entre os Operadores Relacionais Aproximados (ver CAPÍTULO 5) e Aproximados *Fuzzy* (ver CAPÍTULO 6) e um sistema de aprendizado simbólico baseado no ID3.

Conforme pode ser visto em sua arquitetura, mostrada na Figura 7.3, ele é composto pelo Sistema RSQ (*Rough SQL Query*) e pelo Sistema Simbólico ID3 PX.

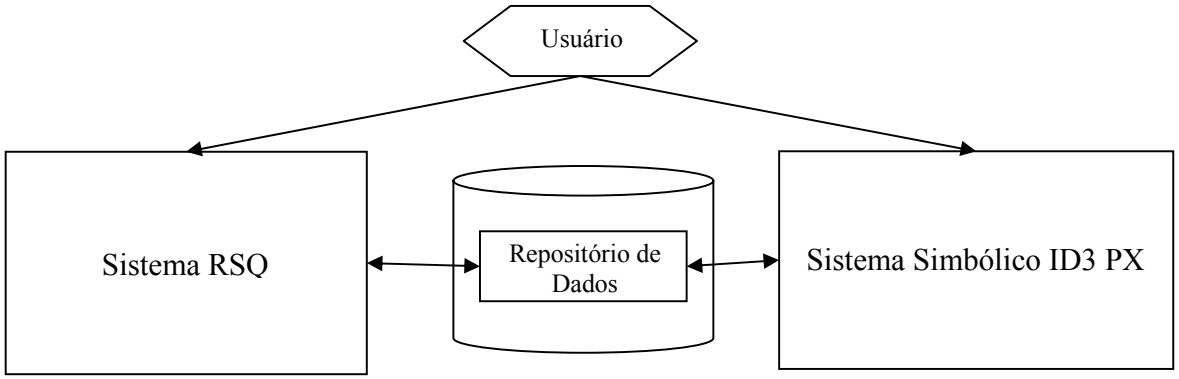


Figura 7.3: Arquitetura do sistema híbrido ROUGH-ID3.

7.2.1 Sistema RSQ

O Sistema RSQ, desenvolvido durante este trabalho de pesquisa, disponibiliza os Operadores Relacionais Aproximados e Aproximados *Fuzzy* para realizar consultas à Base de Dados Relacional Aproximada e Aproximada *Fuzzy*, como apresentados no CAPÍTULO 5 e CAPÍTULO 6. Foi desenvolvido para a plataforma Windows, usando como Sistema Gerenciador de Bases de Dados (SGBD) o Oracle 8i [Loney e Koch 2000] e como interface de desenvolvimento o Borland Delphi 7 [Cantú 2003]. Adotou-se o SGBD Oracle 8i por ele implementar um Modelo Relacional Estendido [Elmasri e Navathe 2003] que possibilita a representação de atributos multivalorados, usados pelo Modelo Relacional Aproximado e Aproximado *Fuzzy*. Os Operadores Relacionais Aproximados *Fuzzy*, apesar de implementados e funcionando devidamente na Base de Dados Relacional Aproximada *Fuzzy*, ainda não estão integrados ao RSQ. Mais detalhes de implementação e das funcionalidades do RSQ estão descritos no ANEXO B.

Conforme pode ser visto em sua arquitetura, apresentada na Figura 7.4, o Sistema RSQ é composto pelos módulos descritos a seguir:

- Interface: faz a ligação do Usuário com os demais módulos do Sistema RSQ;
- Módulo RSQ: se comunica com a Base de Dados Relacional Aproximada para executar consultas solicitadas pelo Usuário aos dados da base, por meio dos Operadores

Relacionais Aproximados. Os algoritmos apresentados no CAPÍTULO 5 descrevem a lógica utilizada na implementação desses operadores;

- Módulo RFSQ: se comunica com a Base de Dados Relacional Aproximada *Fuzzy* para executar consultas solicitadas pelo Usuário aos dados da base, por meio dos Operadores Relacionais Aproximados *Fuzzy*. Os algoritmos apresentados no CAPÍTULO 6 descrevem a lógica utilizada na implementação desses operadores;
- Customizador e Gerador de Arquivos de Bases de Dados: exporta o resultado das consultas executadas pelo Módulo RSQ e pelo Módulo RFSQ para arquivos em formato texto, representados na arquitetura pelo módulo Arquivos de Bases de Dados;
- Base de Dados Relacional Aproximada: contém as relações aproximadas e os Operadores Relacionais Aproximados que são utilizados pelo Módulo RSQ;
- Base de Dados Relacional Aproximada *Fuzzy*: contém as relações aproximadas *fuzzy* e os Operadores Relacionais Aproximados *Fuzzy* que são utilizados pelo Módulo RFSQ;
- Arquivos de Bases de Dados: representam os arquivos texto que foram exportados pelo módulo Customizador e Gerador de Arquivos de Bases de Dados. Tais arquivos estão no formato CSV (*comma separated values*) e seguem as especificações de entrada para o Sistema Simbólico ID3 PX.

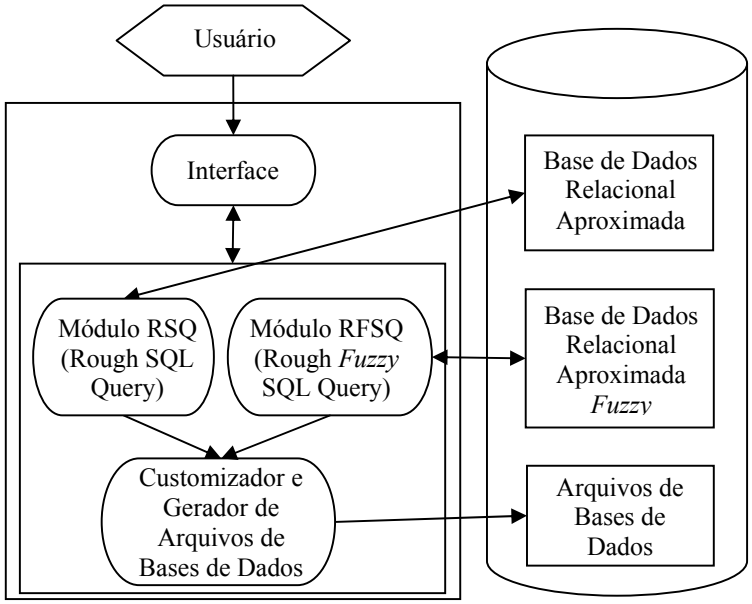


Figura 7.4: Arquitetura do Sistema RSQ.

7.2.2 Sistema Simbólico ID3 PX

O software que implementa o algoritmo ID3, chamado ID3 PX, foi implementado em C++, também para a plataforma Windows, e é utilizado pelo ROUGH-ID3 como uma caixa preta (detalhes sobre os pseudocódigos deste sistema podem ser vistos em [Figueira 2004]).

Esse sistema faz a indução de conhecimento e representação do conceito induzido por meio de árvores de decisão. Sua arquitetura é apresentada na Figura 7.5 e seus módulos são descritos a seguir:

- Interface: faz a ligação do Usuário com os demais módulos do Sistema Simbólico ID3 PX;
- ID3: implementa o algoritmo do ID3;
- Arquivo de Treinamento: arquivo de dados utilizado para a indução do conceito;
- Arquivo de Teste: arquivo de dados utilizado para avaliar a precisão de classificação do conceito induzido;
- Avaliação do Conceito: utiliza o Arquivo de Teste e a Árvore Induzida para avaliar a precisão de classificação do conceito induzido;
- Árvore Induzida: representa a árvore de decisão, que é armazenada em um arquivo texto, gerada pelo Sistema Simbólico ID3 PX baseada no Arquivo de Treinamento.

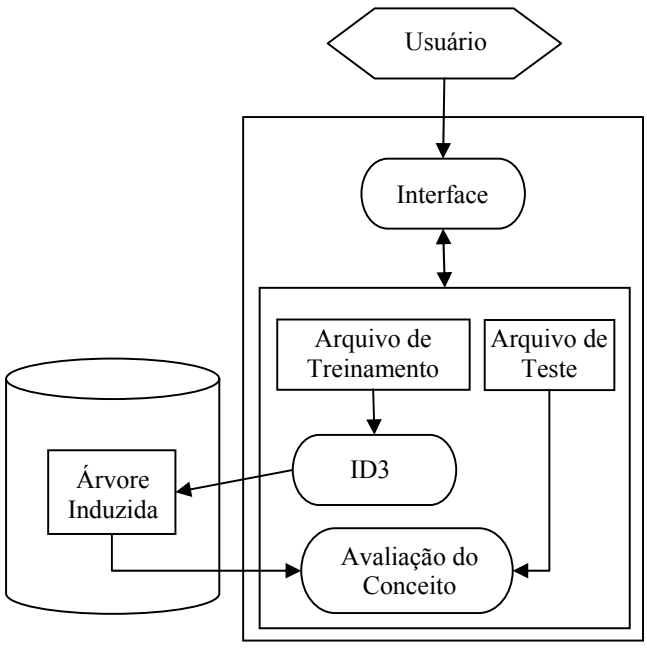


Figura 7.5: Arquitetura do Sistema Simbólico ID3 PX.

7.2.3 Sobre a Extração de Conhecimento Utilizando o Sistema Híbrido ROUGH-ID3

Focalizando a arquitetura do sistema ROUGH-ID3 (Figura 7.3), do Sistema RSQ (Figura 7.4) e do Sistema Simbólico ID3 PX (Figura 7.5), o processo de aquisição de conhecimento é inicializado por meio de uma consulta à Base de Dados Relacional Aproximada (utilizando o Módulo RSQ) ou Aproximada *Fuzzy* (utilizando o Módulo RFSQ) feita pelo Usuário, via Interface do Sistema RSQ. A consulta é processada pelo módulo escolhido pelo Usuário (Módulo RSQ ou Módulo RFSQ) que consulta a base de dados, correspondente ao módulo utilizado, e recupera os dados que satisfazem aos critérios da consulta.

Via de regra, as respostas à consulta são recuperadas da Base de Dados Relacional Aproximada e Aproximada *Fuzzy* na forma de dois conjuntos: os exemplos que pertencem à aproximação inferior e os exemplos que pertencem à região duvidosa do conceito a ser recuperado. O Usuário solicita, então, por meio do módulo Customizador e Gerador de Arquivos de Bases de Dados, a exportação das tuplas resultantes, onde os conjuntos de tuplas (aproximação inferior e região duvidosa) dão origem a dois arquivos textos, representados na arquitetura do sistema ROUGH-ID3 pelos Arquivos de Bases de Dados. Por meio da Interface do Sistema Simbólico ID3 PX o Usuário escolhe os Arquivos de Bases de Dados que deseja utilizar, como Arquivo de Treinamento e Arquivo de Teste. O ID3, mediante solicitação do usuário, induz a expressão do conceito representado pelas instâncias do Arquivo de Treinamento, na forma de uma árvore de decisão, e então a armazena, em formato de arquivo texto. Utilizando o Arquivo de Teste, a Árvore Induzida é avaliada.

Os arquivos textos gerados com os resultados das consultas do Sistema RSQ estão de acordo com a sintaxe exigida pelo ID3 PX. Como este sistema não suporta atributos multivalorados, é necessário, no momento da geração dos arquivos, que cada atributo possua apenas valores atômicos. Sendo assim, cada tupla[7] da relação aproximada ou aproximada *fuzzy* resultante da consulta dá lugar, no arquivo gerado, a todas as suas possíveis interpretações (ver Definição 4.4 para relações aproximadas e Definição 6.4 para aproximadas *fuzzy*). Portanto, cada tupla resultante da consulta dá origem a *n* instâncias no arquivo destino, onde *n* é igual ao número de interpretações da tupla, sendo que estas têm os mesmos valores para cada atributo, com exceção

[7] Os termos tupla e instância são sinônimos dentro do contexto de Bases de Dados, porém, neste capítulo do trabalho, utiliza-se o termo tupla quando os dados estão armazenados em uma relação de uma base de dados e o termo instância caso contrário. Essa convenção foi adota para evitar confusão no uso destes termos ao longo deste capítulo.

dos atributos multivalorados que têm os seus *n* valores distintos distribuídos em cada uma das *n* instâncias geradas. Os valores dos atributos no algoritmo do ID3 devem ser discretos e, portanto, os valores pertencentes a domínios contínuos devem ser discretizados.

Uma consulta à Base de Dados Relacional Aproximada ou Aproximada *Fuzzy* recupera instâncias que satisfazem certamente (aproximação inferior) e instâncias que satisfazem à consulta com certo grau de incerteza (região duvidosa). Esses dois conjuntos de instâncias, então, podem ser submetidos a um processo de aprendizado indutivo usando o Sistema Simbólico ID3 PX. A recuperação de instâncias que satisfazem determinado(s) critério(s) e a sua generalização caracterizam um processo híbrido de extração de conhecimento. O processo de tradução de uma árvore de decisão em um conjunto de regras é trivial – sendo geradas tantas regras quantas forem as folhas da árvore.

7.3 Um Exemplo de Utilização do Sistema ROUGH-ID3

Devido à dificuldade de encontrar bases de dados reais com atributos multivalorados, que representariam a situação ideal para validação do Modelo Relacional Aproximado implementado por meio dos operadores descritos no CAPÍTULO 5, optou-se por adaptar um domínio de dados real, com atributos monovalorados, inserindo nele um atributo multivalorado. Conforme citado anteriormente, a implementação da integração dos Operadores Relacionais Aproximados *Fuzzy* com o Sistema RSQ ainda não foi realizada e, portanto, o teste realizado envolveu apenas o uso da Base de Dados Relacional Aproximada e seus operadores.

Os experimentos realizados e descritos nesta seção utilizaram o domínio de dados conhecido como *Wisconsin Breast Cancer Database* (WBC) [Wolberg e Mangasarian 1990] [Mangasarian e Wolberg 1990], extraído do UCI Machine Learning Repository [Blake e Merz 1998]. Os dados deste domínio têm sido utilizados em inúmeros experimentos relacionados a Aprendizado de Máquina (ver por exemplo [Khare e Yao 2002] e [Tan et al. 2003]).

O arquivo de dados WBC possui 699 instâncias, cada uma delas descritas por 10 atributos e uma classe associada. A Tabela 7.3 nomeia cada um dos atributos e especifica o conjunto de possíveis valores que cada um deles pode assumir. A Tabela 7.4 mostra a distribuição de instâncias entre as classes.

Tabela 7.3: Lista de atributos do domínio WBC**.**

ATRIBUTO	CONJUNTO DE VALORES
Código da Amostra	número inteiro
Densidade da Massa Informe	{1, 2, ..., 10}
Uniformidade do Tamanho da Célula	{1, 2, ..., 10}
Uniformidade da Forma da Célula	{1, 2, ..., 10}
Aderência Marginal	{1, 2, ..., 10}
Tamanho da Célula Epitelial	{1, 2, ..., 10}
Núcleo Reduzido	{1, 2, ..., 10}
Cromatina Suave	{1, 2, ..., 10}
Nucléolo Normal	{1, 2, ..., 10}
Mitose	{1, 2, ..., 10}
Classe	2 – benigno 4 – maligno

Tabela 7.4: Distribuição de classes do domínio WBC**.**

CLASSE	FREQUÊNCIA	%
Benigno	458	65.5
Maligno	241	34.5

O WBC foi 'transformado' em um domínio com atributo multivalorado por meio da criação e inserção de um atributo extra chamado FEBRE, cujo conjunto de possíveis valores está no intervalo [34.5, 41.5].

Da WBC original foram importadas 677 instâncias (todas sem atributos com valores ausentes). Essas 677 instâncias foram então expandidas com a introdução do atributo FEBRE e constituem a relação aproximada BCANCER, de esquema BCANCER(COD, DENSID, TAMANHO, FORMA, ADERENCIA, EPITELIAL, NUCLEO_RED, CROMATINA, NUCLEOLO, MITOSE, FEBRE, CLASSE), como mostra a instanciação na Figura 7.6.

BCANCER	COD	DENSID	TAMANHO	FORMA	ADERENCIA	EPITELIAL	NUCLEO_RED	CROMATINA	NUCLEOLO	MITOSE	FEBRE	CLASSE
	1143978	5	2	1	1	2	1	3	1	1	{35.5, 36.1, 35.8}	2
	1133041	5	3	1	2	2	1	2	1	1	{36.4, 36.7, 36.5, 37.4}	2
	1017023	4	1	1	3	2	1	3	1	1	{36.5, 36.5, 36, 37, 36.5}	2
	1017122	8	10	10	8	7	10	9	7	1	{40, 39.5, 40, 39.5, 39.5}	4
	1041801	5	3	3	3	2	3	4	4	1	{37, 37.5, 38, 37.5, 37}	4

Figura 7.6: Uma instância da relação aproximada BCANCER.

O atributo FEBRE foi implementado com um máximo de dez possíveis valores, que representam medidas de temperatura tiradas dos pacientes ao longo de um período de tempo fixo. O atributo COD representa a identificação associada a cada instância e foi mantido da base original, porém não é utilizado no processo de aquisição de conhecimento.

O atributo FEBRE, ao contrário dos outros atributos, possui um domínio de valores contínuos e, portanto, deve ser discretizado. Isso foi feito utilizando as classes de equivalência induzidas pela relação de indiscernibilidade sobre esse atributo, ou seja, cada classe de equivalência dá origem a um valor discreto para o atributo FEBRE.

Para mostrar o processo de aquisição de conhecimento, um exemplo completo foi executado. É importante enfatizar que esse exemplo de uso foi criado artificialmente para poder explorar as funcionalidades do Sistema RSQ e mostrar a execução do ROUGH-ID3 como um todo. A Figura 7.7 mostra a consulta feita ao Sistema RSQ, via Interface, recuperando informações sobre as tuplas da relação BCANCER que possuem o valor do atributo FEBRE igual ou aproximadamente igual a 38.5 e 39.5. No que segue, os resultados intermediários e final são discutidos.

```
RSELECT BCANCER.DENSID,
        BCANCER.TAMANHO,
        BCANCER.FORMA,
        BCANCER.ADERENCIA,
        BCANCER.EPITELIAL,
        BCANCER.NUCLEO_RED,
        BCANCER.CROMATINA,
        BCANCER.NUCLEOLO,
        BCANCER.MITOSE,
        BCANCER.FEBRE,
        BCANCER.CLASSE

FROM BCANCER

WHERE BCANCER.FEBRE = ( '38.5', '39.5')
```

Figura 7.7: Consulta solicitada ao RSQ.

A consulta mostrada na Figura 7.7 retornou 98 tuplas, sendo 50 pertencentes à aproximação inferior e 48 pertencentes à região duvidosa. Cada um destes conjuntos foi exportado para um arquivo seguindo a sintaxe de entrada exigida pelo ID3 PX. Devido à presença do atributo multivalorado FEBRE, a aproximação inferior, com 50 tuplas, deu origem a um Arquivo de Base de Dados com 290 instâncias, monovaloradas. Pelas mesmas razões, a região duvidosa, com 48 tuplas, deu origem a um arquivo com 268 instâncias.

O conhecimento induzido referente à aproximação inferior está representado por meio da árvore de decisão mostrada na Figura 7.8 e o referente à região duvidosa, por meio da árvore de decisão mostrada na Figura 7.9.

As árvores (em ambos os casos) foram induzidas usando 90% de instâncias dos respectivos conjuntos e avaliadas nos respectivos conjuntos restantes de instâncias, com precisão de 100%

para a aproximação inferior e 99.47% para a região duvidosa na avaliação dos conceitos induzidos.

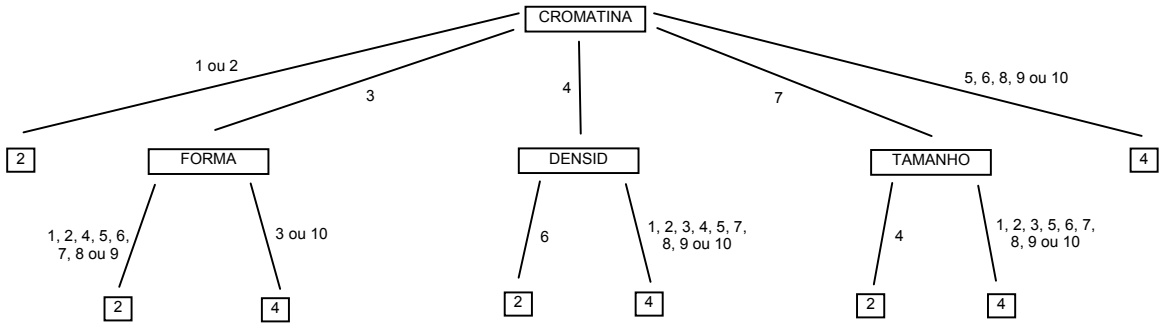


Figura 7.8: Árvore de decisão induzida com instâncias da aproximação inferior.

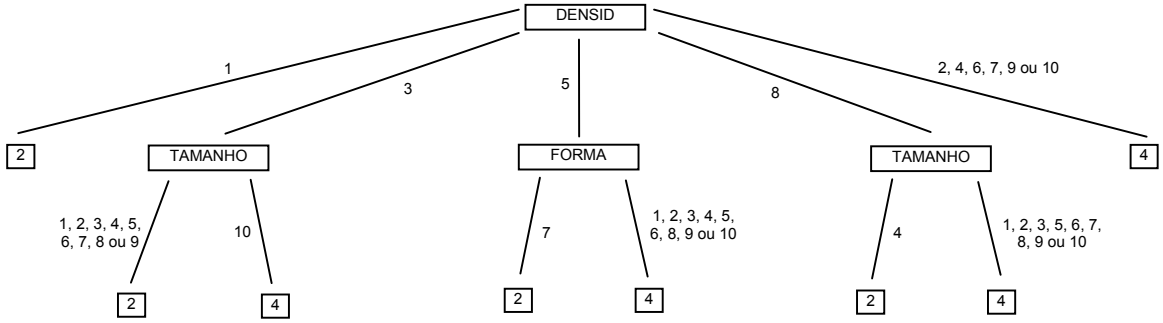


Figura 7.9: Árvore de decisão induzida com instâncias da região duvidosa.

Uma interpretação da árvore da Figura 7.8 seria: Os pacientes que têm febre igual a 38.5 e 39.5 graus e apresentam cromatina suave igual a 7 e tamanho da célula epitelial igual a 4 tem tumor benigno.

A Figura 7.9 indica, por exemplo, que os pacientes que têm febre próxima de 38.5 e 39.5 graus e que têm densidade da massa informe igual a 8 e cujo tamanho da célula epitelial pertence ao conjunto {1, 2, 3, 5, 6, 7, 8, 9, 10}, tem tumor maligno. Note-se que esse tipo de regra não seria extraído num processamento convencional de mineração de dados.

O objetivo desse exemplo de uso do sistema híbrido aproximado simbólico foi mostrar como o Sistema ROUGH-ID3 viabiliza a extração de conhecimento certo e conhecimento com um certo grau de incerteza, a partir de uma Base de Dados Relacional Aproximada. Como a base de

dados com atributo multivalorado foi construída artificialmente com o intuito único de exemplificar o processo, o aprendizado do conceito não seguiu a abordagem tradicional de sistemas de Aprendizado de Máquina (validação cruzada com k partições (ver [Dietterich 1998]), por exemplo).

7.4 Considerações Finais

Este capítulo focalizou a apresentação do sistema híbrido ROUGH-ID3, que combina os Operadores Relacionais Aproximados e Aproximados *Fuzzy* e o sistema simbólico de aprendizado ID3 visando a extração de conhecimento em Bases de Dados Relacionais Aproximadas e Aproximadas *Fuzzy*. No próximo capítulo são apresentadas as principais conclusões do trabalho desenvolvido e algumas possíveis linhas de pesquisa a serem seguidas.

CAPÍTULO 8. CONCLUSÕES

Este trabalho de pesquisa teve como principais objetivos: avaliar a integração de conceitos da TCA em uma Base de Dados Relacional visando aumentar a sua flexibilidade e sua versatilidade e implementar os operadores do Modelo Relacional Aproximado apresentado; avaliar a integração de conceitos *fuzzy* ao Modelo Relacional Aproximado e implementar seus operadores; e, por fim, avaliar a combinação de cada um dos modelos apresentados, por meio dos operadores implementados, com um método simbólico de aprendizado visando a extração de conhecimento.

Para atingir as metas propostas nesta dissertação, a pesquisa iniciou-se buscando familiarização com a TCA, a base mais importante para todo o trabalho. Em seguida foi necessário o mesmo estudo a respeito do Modelo Relacional e da Álgebra Relacional, outra importante base para este trabalho.

Com o embasamento necessário iniciou-se a pesquisa focando o Modelo Relacional Aproximado, desenvolvendo pseudocódigos para os operadores do modelo e implementando-os. O Modelo Relacional Aproximado *Fuzzy*, que tem como base o Modelo Relacional Aproximado, foi estudado em seguida e teve seus conceitos e operadores apresentados e discutidos. Os pseudocódigos dos operadores deste modelo, assim como os do Modelo Relacional Aproximado, também foram desenvolvidos e implementados.

Como não existiam, durante essa pesquisa, Sistemas Gerenciadores de Bases de Dados Relacionais Aproximadas, o sistema RSQ foi desenvolvido para emular tal base, permitindo a utilização dos Operadores Relacionais Aproximados implementados sobre uma Base de Dados Relacional. Foram encontradas dificuldades em obter uma Base de Dado Relacional Aproximada real e, portanto, utilizou-se para os testes uma base de dados artificial, construída a partir de dados de uma base de domínio público.

Para compor o sistema híbrido, juntamente com o Modelo Relacional Aproximado e o Aproximado *Fuzzy*, foi escolhido o ID3 como método simbólico de aprendizado, requerendo, assim, estudo e descrição de seus conceitos fundamentais. O sistema híbrido, chamado ROUGH-ID3, utilizou o sistema ID3 PX como implementação do algoritmo ID3, para induzir conhecimento sobre as informações retornadas pelo RSQ.

Durante a pesquisa sobre os modelos discutidos (Relacional Aproximado e Aproximado *Fuzzy*), notou-se uma falta de formalismo por parte dos autores no momento de definir seus conceitos, o que dificultou a compreensão de alguns de seus significados. Em vista disso, uma das preocupações durante o desenvolvimento dessa dissertação foi ser o mais formal e, consequentemente, o mais claro possível nas definições dos conceitos, tanto que alguns deles foram reescritos por essa razão. Isto pode ser observado, por exemplo, nas definições dos operadores junção aproximada e junção aproximada *fuzzy* (ver Definição 5.6 e Definição 6.14, respectivamente) e na definição do conceito de interpretação no Modelo Relacional Aproximado *Fuzzy* (ver Definição 6.4).

Como contribuições deste trabalho pode-se citar:

- O refinamento do formalismo utilizado na definição dos conceitos do Modelo Relacional Aproximado e Aproximado *Fuzzy*;
- O desenvolvimento dos pseudocódigos dos Operadores Relacionais Aproximados e Aproximados *Fuzzy*;
- A implementação dos Operadores Relacionais Aproximados e Aproximados *Fuzzy*;
- O desenvolvimento de um ambiente que emula uma Base de Dados Relacional Aproximada possibilitando o uso dos Operadores Relacionais Aproximados;
- A proposta de um modelo híbrido de extração de conhecimento, composto por um sistema que implementa os Operadores Relacionais Aproximados e Aproximados *Fuzzy* e um sistema que implementa o sistema simbólico de aprendizado ID3. Tal sistema permite a extração de conhecimento certo e conhecimento com certo grau de incerteza.

Como possíveis linhas de pesquisa na continuação deste trabalho seguem as sugestões:

- Avaliação dos códigos dos operadores implementados neste trabalho, visando a sua otimização, na tentativa de aumentar a velocidade das transações;
- Implementação da integração que possibilite o uso dos Operadores Relacionais Aproximados *Fuzzy* por meio de uma interface, assim como foi feito com os Operadores Relacionais Aproximados neste trabalho de pesquisa;
- Criação e implementação de operadores, tanto para o Modelo Relacional Aproximado quanto para o Modelo Relacional Aproximado *Fuzzy*, que auxiliem na manutenção e manipulação dos dados da tabela IND;

- Elaboração de um algoritmo que seja capaz de avaliar os dados da base e encontrar as relações entre os dados, encontrando as classes de equivalência e gerando automaticamente a tabela IND;
- Identificação de domínios de dados reais que sejam apropriados para a experimentação com o ROUGH-ID3;
- Avaliação de outros sistemas de aprendizado (particularmente o sistema CN2 [Clark e Niblett 1989] [Clark e Boswell 1991], devido à sua eficiência) em conjunto com os Operadores Relacionais Aproximados ou Aproximados *Fuzzy* com o objetivo de avaliar os resultados da agregação do Sistema RSQ a outros sistemas simbólicos.

REFERÊNCIAS BIBLIOGRÁFICAS

[Aho e Ullman 1992]
Aho, A. V. e Ullman, J. D. (1992). Foundations of Computer Science. New York, NY, Computer Science Press, Cap. 8: 387 - 434. 29[8].

[Alves et al. 2004]
Alves, R. T., Delgado, M. R., Lopes, H. S. e Freitas, A. A. (2004). An Artificial Immune System for Fuzzy-rule Induction in Data Mining. In *8th International Conference on Parallel Problem Solving from Nature*, Yao, X. (ed.), Birmingham, UK, p. 1011-1020. 3.

[Beauboeuf e Petry 1994]
Beauboeuf, T. e Petry, F. E. (1994). A Rough Set Model for Relational Databases. Rough Sets, Fuzzy Sets and Knowledge Discovery. Ziarko, W. P. (ed.), Springer-Verlag**:** 100-107. 3, 4, 55, 67, 80.

[Beauboeuf et al. 1998]
Beauboeuf, T., Petry, F. E. e Arora, G. (1998). Information Measures for Rough and Fuzzy Sets and Application to Uncertainty in Relational Databases. In *Rough-Fuzzy Hybridization: A New Trend in Decision-Making*, Pal, S. e Skowron, A. (ed.), Singapore, Springer-Verlag, p. 200-214. 4, 90, 91, 93.

[Beauboeuf 2004]
Beauboeuf, T. (2004). *Comunicação Pessoal via e-mail*. 4, 55, 90, 98, 108.

[Blake e Merz 1998]
Blake, C. L. e Merz, C. J. (1998). UCI Repository of Machine Learning Databases. Dept. of Information and Computer Sciences, University of California, Irvine, CA, http://www.ics.uci.edu/~mlearn/MLRepository.html. 131.

[Cantú 2003]
Cantú, M. (2003). Mastering Delphi 7. Alameda, Ca, SYBEX Inc. 126.

[Clark e Niblett 1989]
Clark, P. e Niblett, T. (1989). "The CN2 Induction Algorithm", Machine Learning, **3**(4): 261-283. 137.

[Clark e Boswell 1991]
Clark, P. e Boswell, R. (1991). Rule Induction with CN2: Some Recent Improvements. In *Machine Learning - EWSL-91*, Kodratoff, Y. (ed.), Berlin, Springer-Verlag, p. 151-163. 137.

[Codd 1970]
Codd, E. F. (1970). "A Relational Model for Large Shared Data Banks", Communications of the ACM, **13**(6): 377-387. 29.

[Deogun et al. 1994]
Deogun, J. S., Raghavan, V. V. e Sever, H. (1994). Rough Set Based Classification Methods and Extended Decision Tables. In *International Workshop on Rough Sets and Soft Computing*,(ed.), San Jose, California, p. 302-309. 3.

[8] Os números que seguem cada referência correspondem às páginas onde cada referência em questão foi citada, nesta dissertação.

[Deogun et al. 1997]
Deogun, J. S., Raghavan, V. V., Sarkar, A. e Sever, H. (1997). Data Mining: Trends in Research and Development. Rough Sets and Data Mining. Lin, T. Y. e Cercone, N. (ed.). Norwell, Massachusetts, Kluwer Academic Publishers**:** 9-45. 3.

[Dietterich 1998]
Dietterich (1998). "Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms", Neural Computation, **10**(7): 1895-1923. 134.

[Elmasri e Navathe 2003]
Elmasri, R. e Navathe, S. B. (2003). Fundamentals of Database Systems. Menlo Park, CA, Pearson Addison Wesley. 29, 126.

[Fernandez-Baizán et al. 1996]
Fernandez-Baizán, M. C., Ruiz, E. M. e Sánchez, J. M. P. (1996). Integrating RDMS and Data Mining Capabilities Using Rough Sets. In *Sixth International Conferences Information Processing and Management of Uncertainty in Knowledge Based Systems*,(ed.), Granada (Spain), p. 1439-1445. 3.

[Figueira 2004]
Figueira, L. B. (2004). *Sobre o Modelo Neural RuleNet e suas Características Simbólica e Cooperativa*. Tese (Mestrado), Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de São Carlos, São Carlos. 126.

[Fortes 2003]
Fortes, D. (2003). "O Insaciável Mundo do Armazenamento", Info Exame, **213**: 76-83. 54.

[Frawley et al. 1992]
Frawley, W., Piatetsky-Shapiro, G. e Matheus, C. (1992). "Knowledge Discovery in Databases: An Overview", AI Magazine: 213-228. 2.

[Grzymala-Busse et al. 1997]
Grzymala-Busse, J. W., Sedelow, S. Y. e Sedelow, W. A. (1997). Machine Learning & Knowledge Acquisition, Rough Sets, and the English Semantic Code. Rough Sets and Data Mining. Lin, T. Y. e Cercone, N. (ed.). Norwell, Massachusetts, Kluwer Academic Publishers**:** 91-107. 3.

[Grzymala-Busse 2003]
Grzymala-Busse, J. W. (2003). Rough Set Strategies to Data with Missing Attribute Values. In *Workshop on Foundations and New Directions in Data Mining, associated with the third IEEE International Conference on Data Mining*,(ed.), Melbourne, FL, USA, p. 56-63. 3.

[Grzymala-Busse e Siddhaye 2004]
Grzymala-Busse, J. W. e Siddhaye, S. (2004). Rough Set Approaches to Rule Induction from Incomplete Data. In *the 10th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*,(ed.), Perugia, Italy, p. 923-930. 3.

[Grzymala-Busse 2004]
Grzymala-Busse, J. W. (2004). Three Approaches to Missing Attribute Values—A Rough Set Perspective. In *Workshop on Foundations of Data Mining, associated with the fourth IEEE International Conference on Data Mining*,(ed.), Brighton, UK, p. 47-54. 3.

[Han e Kamber 2001]
Han, J. e Kamber, M. (2001). Data Mining: Concepts and Techniques. San Francisco, CA, Morgan Kaufmann. 2.

[Hand et al. 2001]
Hand, D., Mannila, H. e Smyth, P. (2001). Principles of Data Mining. Cambridge, CA, MIT Press. 2.

[Hu et al. 2004]
Hu, X., Lin, T. Y. e Han, J. (2004). "A New Rough Sets Model Based on Database Systems", Fundamenta Informaticae, **59**(2-3): 135-152. 3.

[Jesus et al. 2004]
Jesus, M. J. d., Hoffmann, F., Navascués, L. J. e Sánchez, L. (2004). "Induction of Fuzzy-Rule-Based Classifiers With Evolutionary Boosting Algorithms", IEEE Transactions on Fuzzy Systems, **12**(3): 296-308. 3.

[Khare e Yao 2002]
Khare, V. e Yao, X. (2002). Artificial Speciation of Neural Network Ensembles. In *UK Workshop on Computational Intelligence (UKCI'02)*, J.A.Bullinaria (ed.), Birmingham, UK, p. 96-103. 131.

[Kohavi e Frasca 1994]
Kohavi, R. e Frasca, B. (1994). Useful Feature Subsets and Rough Set Reducts. In *The Third International Workshop on Rough Sets and Soft Computing. (RSSC'94)*,(ed.), San Jose, California, p. 310-317. 3.

[Komorowski et al. 2002]
Komorowski, J., Pawlak, Z., Polkowski, L. e Skowron, A. (2002). A Rough Set Perspective on Data and Knowledge. Handbook of Data Mining and Knowledge Discovery. Kloesgen, W. e J., Z. (ed.), Oxford University Press**:** 134-149. 3.

[Krishnaswamy et al. 2002]
Krishnaswamy, S., Zaslavsky, A. e Loke, S. W. (2002). Predicting Run Times of Applications Using Rough Sets. In *The 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*,(ed.), Annecy, France, p. 455-462. 3.

[Kusiak 2001]
Kusiak, A. (2001). "Rough Set Theory: A Data Mining Tool for Semiconductor Manufacturing", IEEE Transactions on Electronics Packaging Manufacturing, **24**(1): 44-50. 3.

[Lin e Cercone 1997]
Lin, T. Y. e Cercone, N. (1997). Rough Sets and Data Mining. Norwell, Massachusetts, Kluwer Academic Publishers. 1.

[Lingras 2001]
Lingras, P. (2001). "Unsupervised Rough Set Classification using GAs", Journal Of Intelligent Information Systems, **16**(3): 215-228. 3.

[Liu et al. 2004]
Liu, W. N., Yao, J. e Yao, Y. Y. (2004). Rough Approximations under Level Fuzzy Sets. In *4th International Conference in Rough Sets and Current Trends in Computing*, Tsumoto, S., Slowinski, R., Komorowski, H. J. e Grzymala-Busse, J. W. (ed.), Uppsala, Sweden, Springer, p. 78-83. 4.

[Loney e Koch 2000]
Loney, K. e Koch, G. (2000). Oracle8i: The Complete Reference. Berkeley, CA, Osborne/McGraw-Hill. 126.

[Mangasarian e Wolberg 1990]
Mangasarian, O. L. e Wolberg, W. H. (1990). "Cancer Diagnosis Via Linear Programming", SIAM News, **23**(5): 1-18. 131.

[Markowska-Kaczmar e Trelak 2003]
Markowska-Kaczmar, U. e Trelak, W. (2003). Extraction of Fuzzy Rules from Trained Neural Network Using Evolutionary Algorithm. In *European Symposium on Artificial Neural Networks*,(ed.), Bruges, Belgium, p. 149-154. 3.

[Mitchell 1997]
Mitchell, T. M. (1997). Machine Learning. New York, McGraw-Hill Companies Inc. 120, 124.

[Mitchell 1999]
Mitchell, T. M. (1999). "Machine Learning and Data Mining", Communications of the ACM, **42**(11). 2.

[Mitra et al. 2002]
Mitra, S., Pal, S. K. e Mitra, P. (2002). "Data Mining in Soft Computing Framework: A Survey", IEEE Transactions on Neural Networks, **13**(1): 3-14. 1, 2.

[Nelson 2001]
Nelson, D. E. (2001). *High Range Resolution Radar Target Classification:A Rough Set Approach*. Tese (Doutorado), College of Engineering and Technology, Ohio University, Athens. 3.

[Nicoletti e Uchôa 1997a]
Nicoletti, M. C. e Uchôa, J. Q. (1997a). O Uso de Funções de Pertinência na Caracterização dos Principais Conceitos da Teoria de Conjuntos Aproximados. Relatório Técnico 005, DC-UFSCar, São Carlos. 26. 6, 23.

[Nicoletti e Uchôa 1997b]
Nicoletti, M. C. e Uchôa, J. Q. (1997b). Conjuntos Aproximados Sob a Perspectiva de Função de Pertinência. In *Anais do 3º Simpósio Brasileiro de Automação Inteligente - SBAL*,(ed.), Vitória, p. 307-312. 23.

[Nicoletti e Uchôa 1998]
Nicoletti, M. C. e Uchôa, J. Q. (1998). O Uso da Teoria de Conjuntos Aproximados na Determinação de Redutos de Conjuntos de Atributos. Relatório Técnico Departamento de Computação 001/98, São Carlos, DC-UFSCar. 38.

[Nicoletti e Uchôa 2002]
Nicoletti, M. C. e Uchôa, J. Q. (2002). A Family of Algorithms for Implementing the Main Concepts of the Rough Set Theory. Advances in Soft Computing (Hybrid Information Systems). Abraham, A. e Koppen, M. (ed.). Quito, Equador, Physica-Verlag**:** 583-595. 6.

[Nicoletti e Camargo 2004]
Nicoletti, M. C. e Camargo, H. A. (2004). Fundamentos da Teoria de Conjuntos Fuzzy. São Carlos, EdUFSCar. 144.

[Pawlak 1981]
Pawlak, Z. (1981). "Information Systems: Theoretical Foundations", Information Systems, **6**(3): 205-218. 3, 6.

[Pawlak 1982]
Pawlak, Z. (1982). "Rough Sets", International Journal of Computer and Information Sciences, **11**(5): 341-356. 6.

[Pawlak 1984a]
Pawlak, Z. (1984a). "Rough Classification", International Journal of Man-Machine Studies,(20): 469-483. 6.

[Pawlak 1984b]
Pawlak, Z. (1984b). Rough Sets and Fuzzy Sets. Relatório Técnico ICS 540, Warsaw, ICS. 10. 24.

[Pawlak 1985a]
Pawlak, Z. (1985a). "On Learning - A Rough Set Approach", Lecture Notes in Computer Science. Askpwhon, Springer-Verlag,(28): 197-227. 6.

[Pawlak 1985b]
Pawlak, Z. (1985b). "Rough Sets and Fuzzy Sets", Fuzzy Sets and Systems, **17**: 99-102. 23, 24.

[Pawlak 1991]
Pawlak, Z. (1991). Rough Sets: Theoretical Aspects of Reasoning About Data. Dordrecht, AA, Kluwer Academic Publishers. 6.

[Pawlak 1994a]
Pawlak, Z. (1994a). Hard and Soft Sets. In *Rough Sets, Fuzzy Sets and Knowledge Discovery*, Ziarko, W. P. (ed.), London, Springer-Verlag, p. 130-135. 26.

[Pawlak 1994b]
Pawlak, Z. (1994b). Rough Sets, Rough Relations and Rough Functions. Relatório Técnico 24, Institute of Computer Science, Warsaw, ICS - Warsaw University of Technology. 6. 28.

[Pawlak 1995]
Pawlak, Z. (1995). "Rough Set Approach to Knowledge-based Decision Support", European Journal of Operational Research, **99**(1): 48-57. 3.

[Quinlan 1986]
Quinlan, J. R. (1986). "Induction of Decision Trees", Machine Learning, **1**(1): 81-106. 119.

[Sarafis et al. 2002]
Sarafis, I., Zalzala, A. M. S. e Trinder, P. W. (2002). A Genetic Rule-Based Data Clustering Toolkit. In *Congress on Evolutionary Computation (CEC)*,(ed.), Honolulu, USA, p. 1238-1243. 2.

[Sarkar 2002]
Sarkar, M. (2002). "Rough-Fuzzy Functions in Classification", Fuzzy Sets and Systems, **132**(3): 353-369. 4.

[Shannon 1948]
Shannon, C. E. (1948). "A Mathematical Theory of Communication", The Bell System Technical Journal, **27**: 623-656. 121.

[Tan et al. 2003]
Tan, K. C., Yu, Q., Heng, C. M. e Lee, T. H. (2003). "Evolutionary Computing for Knowledge Discovery in Medical Diagnosis", Artificial Intelligence in Medicine, **27**(2): 129-154. 131.

[Uchôa 1998]
Uchôa, J. Q. (1998). *Representação e Indução de Conhecimento Usando Teoria de Conjuntos Aproximados*. Tese (Mestrado), Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de São Carlos, São Carlos. 6.

[Wei 2003]
Wei, J.-M. (2003). "Rough Set Based Approach to Selection of Node", International Journal of Computational Cognition, **1**(2): 25-40. 2.

[Witten e Frank 2000]
Witten, I. H. e Frank, E. (2000). Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. San Francisco, CA, Morgan Kaufmann. 2.

[Wolberg e Mangasarian 1990]
Wolberg, W. H. e Mangasarian, O. L. (1990). Multisurface Method of Pattern Separation for Medical Diagnosis Applied to Breast Cytology. In *Proceedings of the National Academy of Sciences*,(ed.), U.S.A., p. 9193-9196. 131.

[Wygralak 1989]
Wygralak, M. (1989). "Rough Sets and Fuzzy Sets -Some Remarks on Interrelations", Fuzzy Sets and Systems, **29**(2): 241-243. 25.

[Zadeh 1965]
Zadeh, L. A. (1965). "Fuzzy sets", Information and Control, **8**(3): 338-353. 3, 6, 19, 21.

[Zhou 2003]
Zhou, Z.-H. (2003). "Three Perspectives of Data Mining", Artificial Intelligence Journal, **143**(1): 139-146. 1, 2.

ANEXO A. PRÉ-REQUISITOS MATEMÁTICOS

As definições aqui apresentadas podem ser encontradas em [Nicoletti e Camargo 2004].

Definição A.1: Sejam os conjuntos $A \neq \varnothing$ e $B \neq \varnothing$. Uma relação binária é qualquer subconjunto do produto cartesiano $A \times B$. Assim, se R é uma relação binária de A em B, então $R \subseteq A \times B$. Se o par $<a, b> \in R$, então a e b estão R-relacionados e escreve-se aRb.

Definição A.2: Seja R uma relação e seja A um conjunto. Então:

$$R[A] = \{y \mid \text{para algum } x \text{ em } A, xRy\}$$

é chamado conjunto de *R-relacionados dos elementos de A*.

Definição A.3: Um subconjunto do produto cartesiano $A \times A$ é uma *relação binária no conjunto A*. O conjunto $A \times A$ é a relação universal em A.

Definição A.4: A relação binária R no conjunto $A \neq \varnothing$ pode ser:

(a) *reflexiva:* se xRx para todo $x \in A$

(b) *irreflexiva:* se $\exists x \in A \mid <x, x> \notin R$

(c) *anti-reflexiva:* se $<x, x> \in R$ para nenhum $x \in A$

(d) *identidade:* se for reflexiva e se $<x, y> \in R$ para $x, y \in A \rightarrow x = y$

(e) *simétrica:* se $<x, y> \in R$ para $x, y \in A \rightarrow <y, x> \in R$

(f) *não-simétrica (ou assimétrica):* se $\exists x, y \in A$ tal que $<x, y> \in R$ e $<y, x> \notin R$

(g) *anti-simétrica:* se $<x, y> \in R$ e $<y, x> \in R$ para $x, y \in A \rightarrow x = y$

(h) *transitiva:* se $<x, y>$ e $<y, z> \in R$, para $x, y, z \in A \rightarrow <x, z> \in R$

Definição A.5: Uma relação em um conjunto é uma *relação de equivalência* se for reflexiva, simétrica e transitiva.

Definição A.6: Uma *partição* $\mathcal{P}(A)$ de um conjunto A, $\mathcal{P}(A) = \{A_i \mid i \in I\}$ é uma família de subconjuntos distintos e não vazios de A tal que $\bigcup_{i \in I} A_i = A$ e $A_i \cap A_j = \varnothing$ para todo $i, j \in I$ $(i \neq j)$.

Os conjuntos A_i são chamados de *blocos da partição.*

Definição A.7: Seja R uma relação de equivalência no conjunto A. Considere o elemento a de A. O conjunto dos R-relacionados de a, R[{a}], notado por [a] é chamado *R-classe de equivalência gerada por a.*

Teorema A.1: Seja R uma relação de equivalência no conjunto A e sejam $a, b \in A$. Então:
(a) $a \in [a]$
(b) se aRb então [a]=[b]

Teorema A.2: Seja X o conjunto de relações de equivalência em um conjunto A e seja Y o conjunto de partições de A. Seja ρ qualquer elemento de X. Existe uma função bijetora $f:X \rightarrow Y$ tal que f(ρ) é o conjunto de todas as ρ-classes de equivalência geradas por elementos de A.

O Teorema A.2 garante uma correspondência entre relações de equivalência e partições; essa correspondência permite o estabelecimento dos conceitos básicos da TCA.

ANEXO B. IMPLEMENTAÇÃO DO SISTEMA

O sistema *Rough SQL Query* (RSQ) foi desenvolvido como parte do presente trabalho visando disponibilizar uma interface para permitir a utilização e testes dos Operadores Relacionais Aproximados apresentados no CAPÍTULO 5. O sistema RSQ, cuja interface foi implementada utilizando a linguagem Object Pascal por meio do software de desenvolvimento Borland Delphi 7, se comunica com uma Base de Dados Relacional de modo a emular uma Base de Dados Relacional Aproximada e executa as operações SQL solicitadas, sejam elas aproximadas ou não. A base de dados é gerenciada pelo SGBD Oracle 8i, onde estão também os Operadores Relacionais Aproximados, que foram implementados utilizando a linguagem PL/SQL, do Oracle.

O RSQ, na verdade, possui três funcionalidades: a primeira é a importação de dados contidas em arquivos no formato CSV para tabelas na base de dados utilizada pelo RSQ, no caso dos dados da tabela se encontrarem nesse formato; a segunda é permitir a edição dos dados contidos em tabelas da base; a terceira, e principal funcionalidade, é emular uma Base de Dados Relacional Aproximada, no que diz respeito à utilização de comandos da linguagem PL/SQL e dos Operadores Relacionais Aproximados, sobre uma Base de Dados Relacional. A seção a seguir apresenta detalhes da base de dados utilizada pelo RSQ.

B.1 Sobre a Base de Dados Relacional Aproximada

A Base de Dados Relacional Aproximada utilizada pelo RSQ é, conforme dito anteriormente, uma Base de Dados Relacional gerenciada pelo Oracle cujos componentes são: uma tabela chamada IND representando a relação de indiscernibilidade (ver Definição 4.5); as tabelas de dados (nos testes feitos neste trabalho usou-se a tabela chamada BCANCER, que armazena os dados extraídos da base de dados WBC descrita no CAPÍTULO 7); e os Operadores Relacionais Aproximados (ver CAPÍTULO 5).

Por definição, a relação de indiscernibilidade contém todos os possíveis valores de domínios da base de dados à qual ela pertence, indicando a classe de equivalência de cada um dos valores. Em termos de implementação, porém, estão contidos na tabela IND somente valores que comparecem na base de dados e que pertencem a classes de equivalência com mais de um elemento, ou seja, somente valores que sejam indiscerníveis de algum outro valor do domínio ao

qual ele pertence e que comparece na base. São três as razões que levaram à implementação da IND dessa maneira: a primeira é que são comuns domínios com infinitos valores ou praticamente infinitos e, portanto, é impossível que todos os valores estejam inseridos na tabela IND; a segunda é que a inserção destes valores na IND é feita manualmente e pode depender de um especialista no domínio para indicar a indiscernibilidade entre eles; a terceira e última é evitar um crescimento desnecessário da tabela IND que poderia causar uma demora nas buscas por valores contidos nela, já que em uma Base de Dados Relacional Aproximada a tabela IND é acessada a cada operação executada.

Para garantir que a classe de equivalência de qualquer valor seja encontrada pelo sistema, as buscas por classes de equivalência de valores que não são localizados na IND, aqueles valores que não são indiscerníveis a nenhum outro dentro de um determinado domínio ou que não comparecem em nenhuma tupla[9] da base de dados, retornam uma classe formada por um único elemento, o próprio valor. A Figura B.1 mostra a tabela IND sendo editada por meio da interface do RSQ.

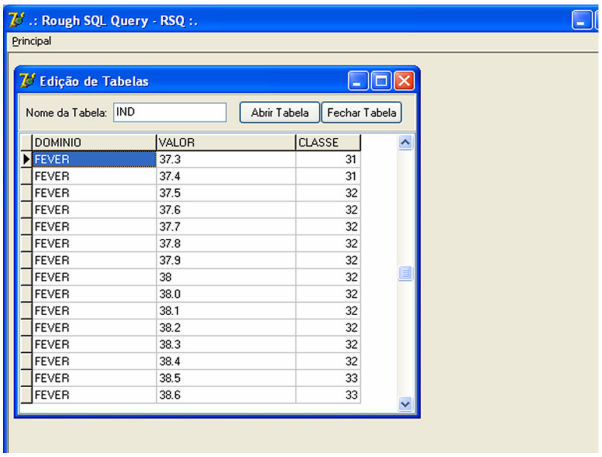


Figura B.1: A tabela IND sendo editada no RSQ.

Numa Base de Dados do Oracle 8i uma das maneiras de utilizar atributos multivalorados é criar um novo tipo de atributo com uma estrutura de vetores, utilizando o tipo VARRAY. O tipo do atributo FEBRE da tabela BCANCER, por exemplo, é um vetor de dez posições de VARCHAR2. A visualização gráfica desta estrutura por meio dos componentes de acesso a

[9] Os termos tupla e instância são sinônimos dentro do contexto de Bases de Dados, porém, neste anexo do trabalho, utiliza-se o termo tupla quando os dados estão armazenados em uma base de dados e o termo instância caso contrário. Essa convenção foi adota para evitar confusão no uso destes termos ao longo deste anexo.

dados disponibilizados pelo Delphi 7 pode ser observada na Figura B.2. O componente representa cada valor do atributo multivalorado como sendo um atributo monovalorado de mesmo nome que o atributo original, acrescentando uma numeração que representa a posição daquele valor dentro do atributo multivalorado.

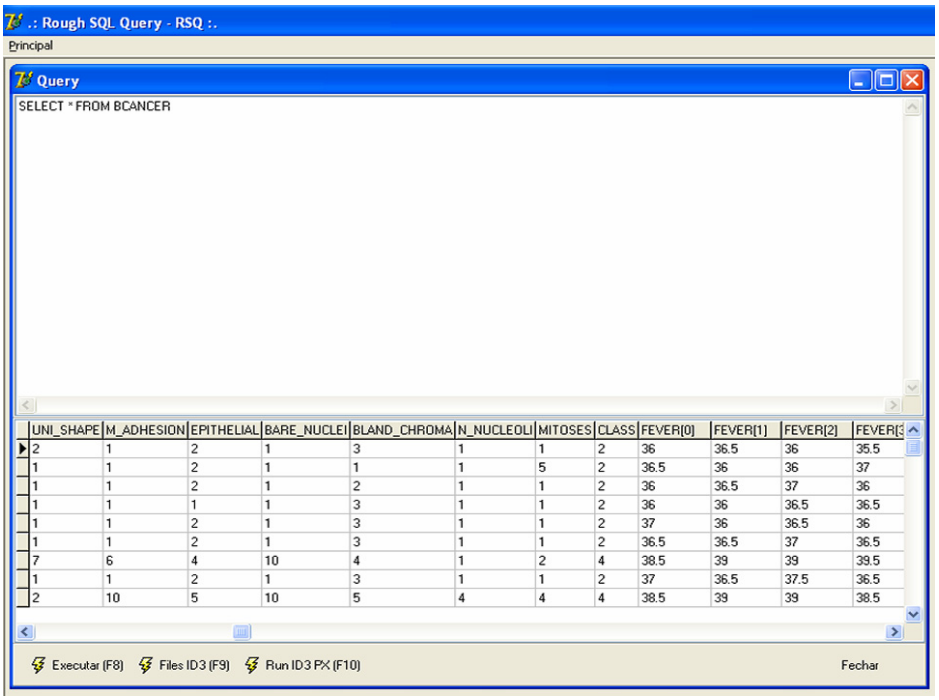


Figura B.2: Visualização de uma consulta sobre a tabela BCANCER.

As próximas seções comentam detalhadamente os módulos do RSQ de modo a explicar a utilização de suas interfaces e suas opções.

B.2 A Importação de Tabelas

A Importação de Tabelas, conforme dito anteriormente, lê arquivos no formato CSV e grava os dados em uma tabela, possibilitando a utilização destes dados na Base de Dados Relacional Aproximada. Os arquivos devem conter os valores dos atributos separados por vírgulas e o esquema da tabela a receber as instâncias deve ser idêntico ao esquema da tabela que deu origem ao arquivo, ou seja, a seqüência dos atributos da tabela deve ser igual à seqüência em que os valores dos atributos comparecem no arquivo e os tipos devem ser compatíveis. Existem duas exceções a essa regra: quando escolhida a utilização do campo DUV, que não é obrigatório, o

mesmo não deve comparecer no arquivo pois o sistema gera automaticamente o valor escolhido na posição indicada; a escolha do campo chave primária no modo automático, pois o sistema gera os valores automaticamente na posição indicada e, portanto, também não deve comparecer no arquivo.

Observa-se que a utilização do campo DUV não é obrigatória para realizar uma importação mas, para que uma tabela seja alvo de um Operador Relacional Aproximado, esta deve conter o atributo DUV em seu esquema.

A importação de dados por meio do RSQ está particularizada a tabelas com apenas um campo chave primária (com tipo compatível com o tipo inteiro) e com um e apenas um campo multivalorado, colocado na última posição do esquema da tabela. Em contrapartida, essa particularização não se aplica à utilização dos Operadores Relacionais Aproximados pois estes aceitam tantos campos multivalorados quantos forem necessários.

A Importação de Tabelas, que pode ser acionada pelo menu Principal ou pela tecla de atalho Ctrl + I, é executada seguindo os passos:

1. Selecionar o arquivo CSV por meio do botão Abrir Arquivo;
2. Indicar o nome da tabela a receber as instâncias na caixa de texto Nome da Tabela;
3. Indicar a quantidade de instâncias na caixa de texto Qtd Instâncias;
4. Indicar a posição do campo chave primária no esquema da tabela e escolher, por meio da opção Automático, se deseja que a importação gere automaticamente os valores da chave primária ou se os valores virão do arquivo;
5. Indicar a posição do campo DUV, se ele for utilizado automaticamente, e qual o valor a ser preenchido (Duvidoso = '*' ou *Null*). Caso este atributo não seja utilizado basta selecionar Não Usar;
6. Selecionar se a tabela a receber os dados deve ser limpa antes da importação ou se os dados a serem importados devem apenas ser inseridos juntos aos dados já existentes. Isso é feito pela opção Limpar Dados Existentes;
7. Iniciar o processo por meio do botão Importar Dados.

O arquivo aberto aparece na paleta Arquivo e é possível edita-lo manualmente antes da importação pois esta utiliza o conteúdo da paleta como fonte. A barra de progresso na parte

inferior da tela indica a porcentagem de instâncias que já foram importadas e a paleta Log exibe um histórico da importação relatando:

1. Quantidade de instâncias iniciais a serem importadas;
2. Quantidade de instâncias ignoradas devido a valores ausentes de atributos;
3. Quantidade de instâncias que falharam na inserção devido a valores de atributos incompatíveis;
4. Quantidade de instâncias inseridas com sucesso;
5. Listagem das instâncias que foram ignoradas ou geraram erro na inserção.

A Figura B.3 mostra a interface do módulo de Importação de Tabelas e as estatísticas retornadas por ela em uma importação.

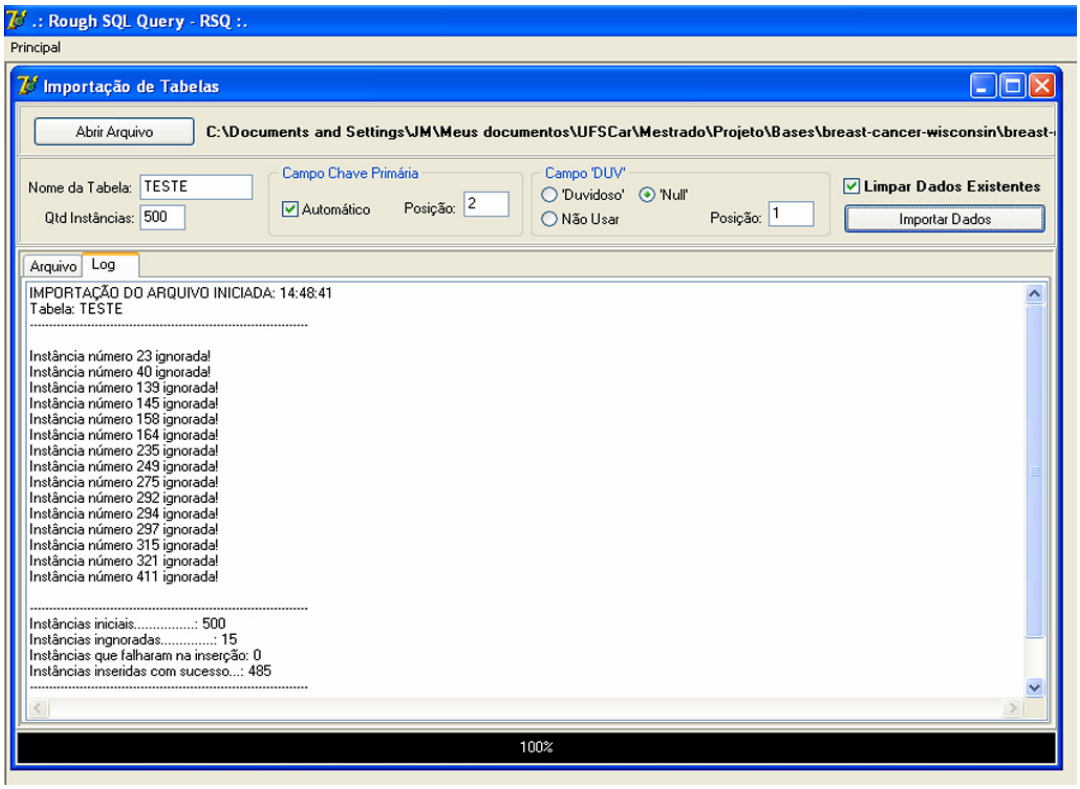
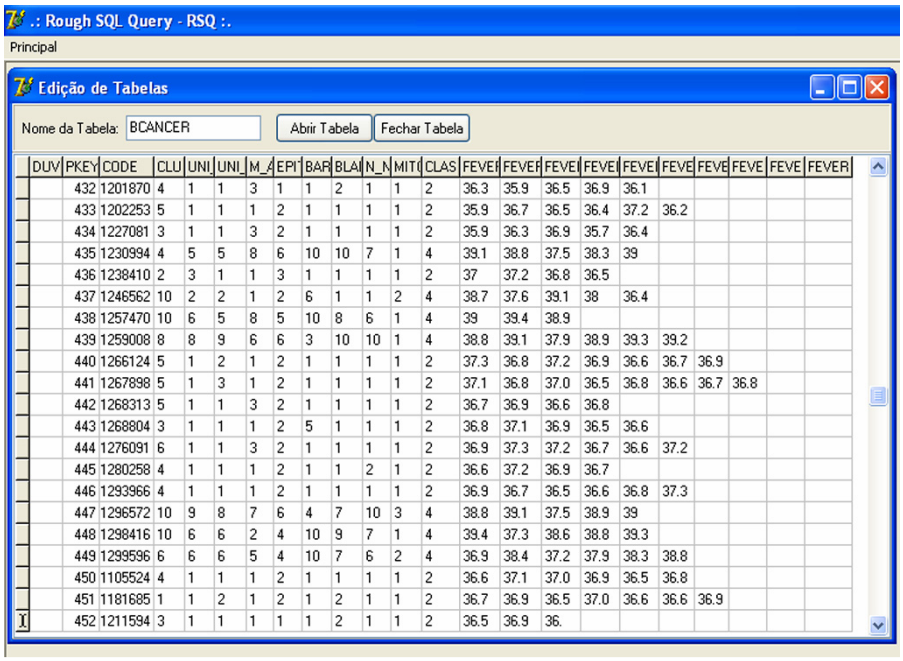


Figura B.3: Interface do módulo de Importação de Tabelas.

B.3 A Edição de Tabelas

A Edição de Tabelas é um módulo muito simples de ser utilizado; fornece uma interface amigável para a edição das tuplas das tabelas que fazem parte da Base de Dados Relacional Aproximada.

O usuário apenas precisa informar o nome da tabela a ser editada e clicar no botão Abrir Tabela. Com a tabela carregada, os valores de atributos das tuplas já existentes podem ser editados selecionando-os e pressionando a tecla F2. A tupla editada é salva quando uma outra tupla é selecionada ou quando a tabela é fechada, clicando no botão Fechar Tabela. Além disso, tuplas também podem ser excluídas ou inseridas, pressionando as teclas Ctrl + Delete e Insert, respectivamente. A Edição de Tabelas pode ser acionada pelo menu Principal ou pela tecla de atalho Ctrl + E e sua interface é mostrada na Figura B.4.



DUV	PKEY	CODE	CLU	UNI	UNI	M_A	EPI	BAR	BLA	N_N	MIT	CLAS	FEVER	FEVER	FEVE	FEVE	FEVE	FEVE	FEVE	FEVE	FEVER
	432	1201870	4	1	1	3	1	1	2	1	1	2	36.3	35.9	36.5	36.9	36.1				
	433	1202253	5	1	1	1	2	1	1	1	1	2	35.9	36.7	36.5	36.4	37.2	36.2			
	434	1227081	3	1	1	3	2	1	1	1	1	2	35.9	36.3	36.9	35.7	36.4				
	435	1230994	4	5	5	8	6	10	10	7	1	4	39.1	38.8	37.5	38.3	39				
	436	1238410	2	3	1	1	3	1	1	1	1	2	37	37.2	36.8	36.5					
	437	1246562	10	2	2	1	2	6	1	1	2	4	38.7	37.6	39.1	38	36.4				
	438	1257470	10	6	5	8	5	10	8	6	1	4	39	39.4	38.9						
	439	1259008	8	8	9	6	6	3	10	10	1	4	38.8	39.1	37.9	38.9	39.3	39.2			
	440	1266124	5	1	2	1	2	1	1	1	1	2	37.3	36.8	37.2	36.9	36.6	36.7	36.9		
	441	1267898	5	1	3	1	2	1	1	1	1	2	37.1	36.8	37.0	36.5	36.8	36.6	36.7	36.8	
	442	1268313	5	1	1	3	2	1	1	1	1	2	36.7	36.9	36.6	36.8					
	443	1268804	3	1	1	1	2	5	1	1	1	2	36.8	37.1	36.9	36.5	36.6				
	444	1276091	6	1	1	3	2	1	1	1	1	2	36.9	37.3	37.2	36.7	36.6	37.2			
	445	1280258	4	1	1	1	2	1	1	2	1	2	36.6	37.2	36.9	36.7					
	446	1293966	4	1	1	1	2	1	1	1	1	2	36.9	36.7	36.5	36.6	36.8	37.3			
	447	1296572	10	9	8	7	6	4	7	10	3	4	38.8	39.1	37.5	38.9	39				
	448	1298416	10	6	6	2	4	10	9	7	1	4	39.4	37.3	38.6	38.8	39.3				
	449	1299596	6	6	6	5	4	10	7	6	2	4	36.9	38.4	37.2	37.9	38.3	38.8			
	450	1105524	4	1	1	1	2	1	1	1	1	2	36.6	37.1	37.0	36.9	36.5	36.8			
	451	1181685	1	1	2	1	2	1	2	1	1	2	36.7	36.9	36.5	37.0	36.6	36.6	36.9		
I	452	1211594	3	1	1	1	1	1	2	1	1	2	36.5	36.9	36.						

Figura B.4: Interface do módulo de Edição de Tabelas.

B.4 SQL Query

O SQL Query é o módulo do RSQ que permite ao usuário acessar a Base de Dados Relacional Aproximada por meio de comandos, sejam eles compostos por Operadores Relacionais ou por Operadores Relacionais Aproximados. Além disso, exporta o resultado de uma consulta

aproximada para arquivos de bases de dados seguindo a sintaxe do ID3 PX, programa que implementa o Sistema Simbólico apresentado no CAPÍTULO 7. Apesar desse sistema poder ser executado por meio do RSQ (acionando o botão Run ID3 PX ou a tecla de atalho F10 e localizando o executável do aplicativo), ele funciona de maneira independente, utilizando como base para a indução de conhecimento os arquivos de bases de dados gerados pelo RSQ.

A execução de comandos é simples, sendo acionada por meio do botão Executar ou da tecla de atalho F8. No entanto, algumas diferenças na sintaxe dos operadores devem ser observadas:

1. As referências a colunas de tabelas (mesmo que a referência seja em relação a todas as colunas de uma tabela por meio do símbolo '*') devem ter, obrigatoriamente, a forma TABELA.COLUNA;
2. Nas condições da cláusula WHERE as comparações são feitas utilizando apenas o símbolo de igualdade (=);
3. Nas condições da cláusula WHERE, quando as comparações são feitas entre um valor de atributo e um valor informado pelo usuário, a condição deve ter, obrigatoriamente, a forma TABELA.COLUNA = ('valor'). Se o valor informado pelo usuário for uma lista de valores, estes devem ser separados por vírgula. Por exemplo: TABELA.COLUNA = ('valor_1', 'valor_2', 'valor_3');
4. Para o uso de Operadores Relacionais não há mudanças e a sintaxe da linguagem PL/SQL do Oracle deve ser seguida.

Quando Operadores Relacionais são utilizados, este módulo do RSQ funciona apenas como interface, enviando o comando diretamente para o Oracle sem qualquer preparação ou verificação de sintaxe e exibindo os resultados na tela. Já os Operadores Relacionais Aproximados são executados por meio de chamadas a funções contidas na base de dados, sendo feitas várias comunicações do módulo com a base até que o resultado final da consulta seja recuperado e exibido na interface. O módulo lê o comando digitado pelo usuário, verifica a sintaxe e retira dele os parâmetros necessários para a execução das funções.

Como pode ser visto na Figura B.5 a área onde os comandos são digitados fica na parte superior da tela, enquanto que na parte inferior são exibidos os resultados das consultas e suas estatísticas, nas paletas Consulta e Log, respectivamente.

As informações relatadas nas estatísticas de uma consulta são:

1. Hora do início da operação;
2. Quantidade de tuplas retornadas na operação;
3. Quantidade de tuplas retornadas que pertencem à aproximação inferior;
4. Quantidade de tuplas retornadas que pertencem à região duvidosa;
5. Hora da conclusão da operação;
6. Duração da operação.

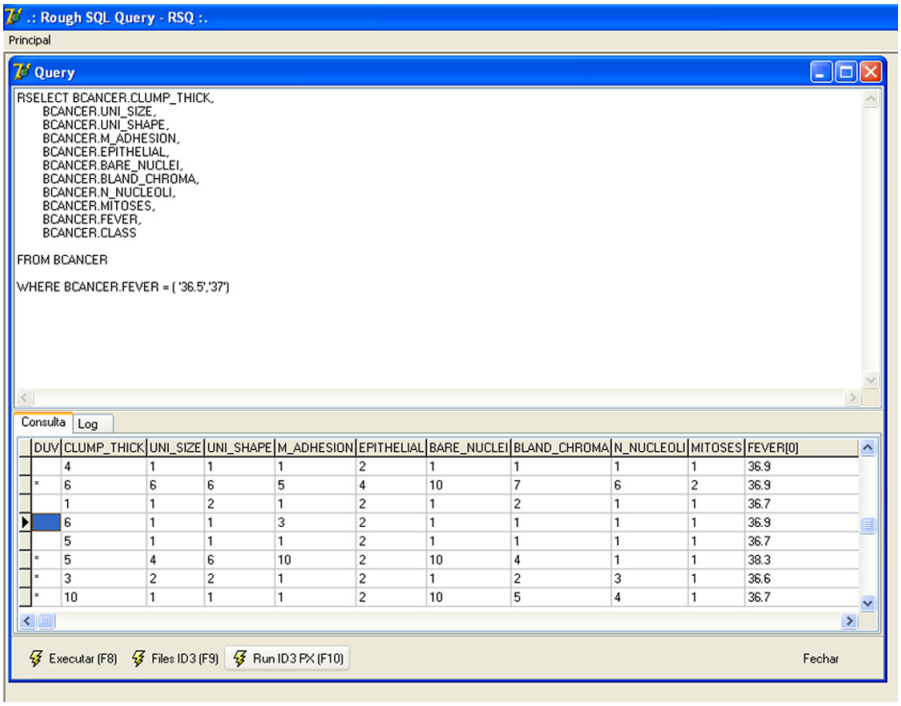


Figura B.5: Interface do módulo SQL Query.

A exportação das tuplas resultantes das consultas em arquivos de bases de dados, que é acionada pelo botão Files ID3 ou pela tecla de atalho F9, gera dois arquivos: um que contém as tuplas pertencentes à aproximação inferior e um que contém as tuplas pertencentes à região duvidosa. Como as consultas do Modelo Relacional não retornam esses dois conjuntos de tuplas a exportação é feita somente com o resultado de uma consulta aproximada. Na paleta Log são exibidas as seguintes estatísticas sobre a geração dos arquivos:

1. Hora do início da exportação;
2. Total de tuplas da relação origem;
3. Quantidade de tuplas da relação origem que pertencem à aproximação inferior;

4. Quantidade de tuplas da relação origem que pertencem à região duvidosa;
5. Total de instâncias geradas com base nas tuplas;
6. Quantidade de instâncias geradas com as tuplas pertencentes à aproximação inferior;
7. Quantidade de instâncias geradas com as tuplas pertencentes à região duvidosa;
8. Hora da conclusão da exportação;
9. Duração da exportação.

Conforme explicado no CAPÍTULO 7 cada tupla resultante da consulta dá origem a *n* instâncias nos arquivos de bases de dados, onde *n* é o número de interpretações (ver Definição 4.4) da tupla. A Figura B.6 mostra a paleta Log com as estatísticas de uma exportação de arquivos para o ID3 PX.

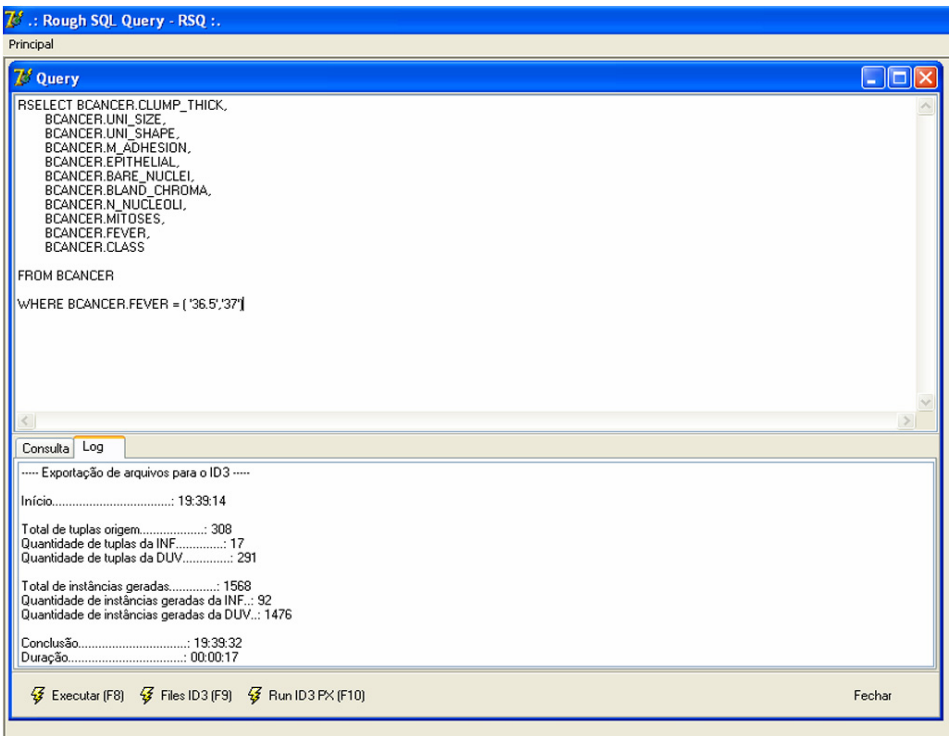


Figura B.6: Estatísticas da exportação de arquivos para o ID3 PX.