

**UNIVERSIDADE FEDERAL DE SÃO CARLOS**  
**DEPARTAMENTO DE COMPUTAÇÃO**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

**Sistema FOQuE para Expansão  
Semântica de Consultas baseada em  
Ontologias Difusas**

**Cristiane Akemi Yaguinuma**

**São Carlos – SP**  
**Junho/2007**

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

***“Sistema FOQuE para Expansão Semântica de Consultas  
baseada em Ontologias Difusas”***

**CRISTIANE AKEMI YAGUINUMA**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

**Membros da Banca:**

  
\_\_\_\_\_  
**Prof. Dr. Mauro Biajiz**  
**(Orientador – DC/UFSCar)**

  
\_\_\_\_\_  
**Profa. Dra. Marilde Teresinha Prado Santos**  
**(Co-Orientadora – DC/UFSCar)**

  
\_\_\_\_\_  
**Prof. Dr. Estevam Rafael Hruschka Junior**  
**(DC/UFSCar)**

  
\_\_\_\_\_  
**Profa. Dra. Maria Luiza Machado Campos**  
**(DCC/UFRJ)**

**São Carlos**  
**Junho/2007**

**Ficha catalográfica elaborada pelo DePT da  
Biblioteca Comunitária da UFSCar**

Y12sf

Yaguinuma, Cristiane Akemi.

Sistema FOQuE para expansão semântica de consultas baseada em ontologias difusas / Cristiane Akemi Yaguinuma. -- São Carlos : UFSCar, 2008.  
111 f.

Dissertação (Mestrado) -- Universidade Federal de São Carlos, 2007.

1. Banco de dados. 2. Expansão de consultas. 3. Informação semântica nebulosa. 4. Ontologia. 5. Fuzzy logic. I. Título.

CDD: 005.74 (20<sup>a</sup>)

*Para meus pais, a quem devo tudo o que sou e tudo o que serei.*

*Para meu namorado Eduardo, uma pessoa essencial para que este trabalho se tornasse  
realidade.*

## **AGRADECIMENTOS**

A Deus, que me permite ter saúde para aproveitar a oportunidade de aprender a cada dia mais.

Aos meus orientadores, Mauro e Marilde, pela amizade, dedicação, paciência e pelo grande aprendizado que me proporcionaram.

Aos amigos de todas as horas, pelo companheirismo nos bons e maus momentos.

Aos colegas do grupo de banco de dados, pela cooperação e amizade.

A Sandra Abib, pela amizade e pelo carinho durante toda a graduação e o mestrado, além do apoio essencial para a realização dos testes.

Ao aluno Victor de Souza Faria, aos professores do Departamento de Computação e à ProGrad, pelo apoio durante a realização dos testes.

À CAPES, pelo auxílio financeiro deste trabalho.

Meus sinceros agradecimentos a todos.

*Sei que meu trabalho é uma gota no oceano,*

*mas sem ele, o oceano seria menor.*

*(Madre Teresa de Calcutá)*

## **RESUMO**

Diante da crescente facilidade de acesso a dados de diversas áreas do conhecimento, cada vez mais são necessárias técnicas eficazes para recuperar a informação desejada, visando reduzir respostas irrelevantes e assegurar que resultados relevantes não sejam desprezados. Dentro deste contexto, este trabalho apresenta o sistema FOQuE, desenvolvido para realizar diversos tipos de expansão de consultas com o intuito de recuperar resultados semanticamente relevantes e abrangentes. Baseado em ontologias difusas, este sistema é capaz de obter resultados aproximados que satisfaçam aos requisitos do usuário, de acordo com parâmetros de expansão especificados por ele. As respostas adicionais recuperadas pelo sistema FOQuE são classificadas segundo o tipo de expansão realizada e a relevância para a consulta, melhorando, assim, a forma como os resultados são apresentados ao usuário.

Palavras-chave: expansão de consultas, ontologias difusas, semântica de dados, lógica difusa.

## **ABSTRACT**

As availability of data from several areas of knowledge grows, it is even more necessary to develop effective techniques to retrieve the desired information, aiming to reduce irrelevant answers and ensure that relevant results are not ignored. Considering this context, we present the FOQuE system, developed to perform query expansion in order to retrieve semantically relevant and broad results. Based on fuzzy ontologies, this system is able to obtain approximate results that satisfy user requirements according to expansion parameters defined by the user. The additional answers retrieved by the FOQuE system are classified according to the semantic expansion performed and the relevance to the query, therefore it is possible to improve results that are presented to the user.

Keywords: query expansion, fuzzy ontologies, data semantics, fuzzy logics.



## LISTA DE FIGURAS

Figura 2.1 – Tipos de representação de ontologias [USCHOLD; GRÜNINGER, 2004].....	15
Figura 2.2 – Exemplo de uma ontologia simples.....	16
Figura 2.3 – Funções de pertinência para os conjuntos difusos <i>jovem, adulto e idoso</i> [KLIR; YUAN, 1995]. .....	20
Figura 2.4 – Representação gráfica dos conceitos <i>jovem, adulto e idoso</i> . ....	21
Figura 2.5 – Exemplo de taxonomia difusa. ....	24
Figura 2.6 – Exemplo de ontologia difusa com classes e relacionamentos difusos.....	25
Figura 2.7 – Representação de classe e relacionamento difusos em <i>Fuzzy OWL</i> [STOILLOS <i>et al.</i> , 2006]....	26
Figura 3.1 – Exemplo de consulta em <i>Corese</i> [CORBY <i>et al.</i> , 2006]. ....	37
Figura 3.2 – Exemplo de consulta por proximidade estrutural [CORBY <i>et al.</i> , 2006]. ....	37
Figura 4.1 – Arquitetura do sistema FOQuE.....	40
Figura 4.2 – Representação abstrata de classe difusa. ....	43
Figura 4.3 – Representação em OWL/XML de uma instância de classe difusa.....	43
Figura 4.4 – Representação abstrata de relacionamento difuso binário.....	44
Figura 4.5 – Representação em OWL/XML de relacionamento difuso binário.....	44
Figura 4.6 – Esquema do banco de dados contendo produtos de supermercado.....	47
Figura 4.7 – Visão geral da ontologia sobre produtos de supermercado.....	47
Figura 4.8 – Consulta contendo o termo homônimo <i>ventilador</i> . ....	51
Figura 4.9 – Diferentes conceitos para o termo <i>ventilador</i> . ....	51
Figura 4.10 – Consulta modificada para recuperar o eletrodoméstico <i>ventilador</i> . ....	52
Figura 4.11 – Consulta modificada para recuperar o componente <i>ventilador</i> . ....	52
Figura 4.12 – Consulta contendo o termo <i>vegetal</i> correspondente a uma classe na ontologia. ....	53
Figura 4.13 – Subclasses e instâncias da classe difusa <i>Vegetal</i> . ....	53
Figura 4.14 – Consulta expandida para recuperar instâncias da classe difusa <i>Vegetal</i> . ....	54
Figura 4.15 - Consulta pelo produto <i>telefone sem fio</i> . ....	56
Figura 4.16 – Relacionamentos de similaridade entre <i>telefone sem fio</i> e outras instâncias da ontologia.....	56
Figura 4.17 – Consulta expandida por similaridade a <i>telefone sem fio</i> .....	56
Figura 4.18 – Partes de <i>desktop</i> contendo graus de pertinência todo-parte. ....	58
Figura 4.19 – Algoritmo para o cálculo de grau de proximidade todo-parte.....	59
Figura 4.20 – Consulta pelo produto <i>desktop</i> , que pode ser expandido por proximidade todo-parte.....	60
Figura 4.21 – “Todos” da ontologia difusa e respectivas partes.....	61
Figura 4.22 – Consulta expandida por proximidade todo-parte. ....	62
Figura 4.23 – Consulta por componentes do produto <i>notebook</i> . ....	63
Figura 4.24 – Fragmento das tabelas <i>Produto</i> e <i>Composto</i> . ....	63
Figura 4.25 – Consulta expandida considerando graus de pertinência em relacionamentos transitivos.....	65
Figura 4.26 – Níveis de transitividade para <i>desktop</i> e <i>notebook</i> com relação a <i>fuz:partOf</i> .....	66
Figura 4.27 – Algoritmo para expandir consultas por transitividade.....	66
Figura 4.28 – Consulta expandida considerando o relacionamento transitivo <i>fuz:hasPart</i> . ....	67

Figura 4.29 – Consulta resultante das expansões de desktop e fuz:hasPart.....	69
Figura 4.30 – Resultados classificados para a consulta expandida. ....	70
Figura 5.1 – Interface gráfica do protótipo do sistema FOQuE.....	74
Figura 5.2 – Visão conceitual do esquema do banco de dados utilizado nos testes.....	76
Figura 5.3 – Ontologia difusa sobre disciplinas da área de computação e informática. ....	78
Figura 5.4 – Consulta por disciplinas da categoria Inteligência artificial. ....	81
Figura 5.5 – Conceitos da ontologia difusa relacionados ao termo homônimo <i>Inteligência artificial</i> . ....	82
Figura 5.6 – Consulta expandida para recuperar disciplinas da classe <i>Inteligência artificial</i> . ....	82
Figura 5.7 – Consulta por disciplinas da categoria <i>Programação</i> . ....	84
Figura 5.8 – Disciplinas pertencentes à classe <i>Programação</i> .....	84
Figura 5.9 – Consulta expandida para recuperar disciplinas da classe <i>Programação</i> .....	85
Figura 5.10 – Consulta por disciplinas que contém o tópico <i>Implementação de Sistemas</i> . ....	87
Figura 5.11 – Consulta por disciplinas próximas a <i>Sistemas distribuídos</i> . ....	90
Figura 5.12 – Consulta por pré-requisitos da disciplina <i>Laboratório de compiladores</i> . ....	91
Figura 5.13 – Níveis de transitividade para a disciplina <i>Laboratório de compiladores</i> e o relacionamento <i>temPreRequisito</i> . ....	92
Figura 5.14 – Consulta expandida por transitividade para a disciplina <i>Laboratório de compiladores</i> e o relacionamento <i>temPreRequisito</i> . ....	93
Figura 5.15 – Pré-requisitos da disciplina <i>Laboratório de compiladores</i> . ....	94
Figura 5.16 – Consulta por notas de um aluno nos pré-requisitos da disciplina <i>Construção de compiladores</i> . ....	96
Figura 5.17 – Notas de um aluno nos pré-requisitos da disciplina <i>Construção de compiladores</i> . ....	97
Figura 5.18 – Consulta por notas de um aluno nas disciplinas da classe <i>Arquitetura de computadores</i> . ....	98
Figura 5.19 – Notas de um aluno nas disciplinas da classe <i>Arquitetura de computadores</i> . ....	98
Figura 5.20 – Consulta por pré-requisitos da disciplina <i>Visualização computacional</i> . ....	99
Figura 5.21 – Pré-requisitos da disciplina <i>Visualização computacional</i> . ....	100
Figura 5.22 – Notas de um aluno em disciplinas próximas a <i>Introdução a ciência da computação 1</i> . ....	100

## LISTA DE TABELAS

Tabela 4.1 – Graus de proximidade todo-parte entre “todos” da ontologia difusa. ....	61
Tabela 5.1 – Revocação e precisão dos resultados obtidos pela consulta original e pela consulta expandida. .....	83
Tabela 5.2 – Similaridade entre tópicos de disciplinas.....	86
Tabela 5.3 – Disciplinas próximas a <i>Sistemas distribuídos</i> (estratégia da média ponderada). ....	88
Tabela 5.4 – Disciplinas próximas a <i>Sistemas distribuídos</i> (estratégia da média aritmética). ....	89

# SUMÁRIO

<b>1. INTRODUÇÃO</b> .....	<b>11</b>
1.1. MOTIVAÇÃO E OBJETIVO.....	11
1.2. ORGANIZAÇÃO DA DISSERTAÇÃO.....	12
<b>2. ONTOLOGIAS DIFUSAS</b> .....	<b>14</b>
2.1. CONSIDERAÇÕES INICIAIS.....	14
2.2. ONTOLOGIAS .....	14
2.2.1. <i>Definição e caracterização</i> .....	14
2.2.2. <i>Representação de ontologias</i> .....	16
2.2.3. <i>Aplicações de ontologias</i> .....	18
2.3. LÓGICA DIFUSA.....	19
2.3.1. <i>Conjuntos difusos</i> .....	19
2.3.2. <i>Relações difusas</i> .....	22
2.4. ONTOLOGIAS DIFUSAS.....	23
2.4.1. <i>Taxonomias difusas</i> .....	23
2.4.2. <i>Ontologias com classes e relacionamentos difusos</i> .....	24
2.4.3. <i>Ontologias com hierarquia difusa de classes e relacionamentos</i> .....	27
2.5. CONSIDERAÇÕES FINAIS .....	28
<b>3. EXPANSÃO SEMÂNTICA DE CONSULTAS BASEADA EM ONTOLOGIAS</b> .....	<b>29</b>
3.1. CONSIDERAÇÕES INICIAIS .....	29
3.2. MOTIVAÇÃO E OBJETIVOS .....	29
3.3. ABORDAGENS DE EXPANSÃO DE CONSULTAS BASEADA EM ONTOLOGIAS TRADICIONAIS .....	30
3.4. ABORDAGENS DE EXPANSÃO DE CONSULTAS BASEADA EM ONTOLOGIAS DIFUSAS.....	34
3.5. CONSIDERAÇÕES FINAIS .....	38
<b>4. SISTEMA FOQUE PARA EXPANSÃO DE CONSULTAS BASEADA EM ONTOLOGIAS DIFUSAS</b> .....	<b>39</b>
4.1. CONSIDERAÇÕES INICIAIS .....	39
4.2. ARQUITETURA DO SISTEMA FOQUE.....	40
4.3. ONTOLOGIA DIFUSA .....	41
4.4. MAPEAMENTO ENTRE ONTOLOGIA DIFUSA E BANCO DE DADOS .....	46
4.5. ANÁLISE DA CONSULTA .....	49
4.6. EXPANSÃO DA CONSULTA .....	50
4.6.1. <i>Homônimos</i> .....	50
4.6.2. <i>Classes</i> .....	53
4.6.3. <i>Similaridade</i> .....	54

4.6.4.	<i>Proximidade todo-parte</i> .....	57
4.6.5.	<i>Transitividade</i> .....	62
4.7.	CLASSIFICAÇÃO DOS RESULTADOS.....	68
4.8.	CONSIDERAÇÕES FINAIS .....	71
<b>5.</b>	<b>IMPLEMENTAÇÃO, TESTES E DISCUSSÃO DOS RESULTADOS</b> .....	<b>73</b>
5.1.	CONSIDERAÇÕES INICIAIS .....	73
5.2.	PROTÓTIPO DESENVOLVIDO .....	73
5.3.	TESTES.....	75
5.3.1.	<i>Banco de dados</i> .....	76
5.3.2.	<i>Ontologia difusa</i> .....	77
5.3.3.	<i>Mapeamento</i> .....	79
5.3.4.	<i>Consultas realizadas</i> .....	81
5.3.4.1	Homônimos.....	81
5.3.4.2	Classes .....	83
5.3.4.3	Similaridade.....	85
5.3.4.4	Proximidade todo-parte.....	87
5.3.4.5	Transitividade .....	90
5.3.4.6	Consultas envolvendo vários tipos de expansão .....	95
5.4.	DISCUSSÃO DOS RESULTADOS.....	101
5.5.	CONSIDERAÇÕES FINAIS .....	102
<b>6.</b>	<b>CONCLUSÕES</b> .....	<b>103</b>
6.1.	RESULTADOS ALCANÇADOS.....	103
6.2.	CONTRIBUIÇÕES.....	104
6.3.	TRABALHOS FUTUROS.....	105
6.3.1.	<i>Testes</i> .....	105
6.3.2.	<i>Expansão de consulta</i> .....	105
6.3.3.	<i>Análise da consulta</i> .....	106
6.3.4.	<i>Ontologia difusa</i> .....	106
<b>7.</b>	<b>REFERÊNCIAS</b> .....	<b>108</b>

# 1. INTRODUÇÃO

## 1.1. MOTIVAÇÃO E OBJETIVO

A crescente disponibilidade de dados tem facilitado o acesso à informação pertencente a diversas áreas do conhecimento. Diante da oferta abundante de dados, cada vez mais são necessárias técnicas eficazes para recuperar a informação desejada, de modo a reduzir a quantidade de respostas irrelevantes (alta precisão) e assegurar que resultados relevantes não sejam desprezados (alta revocação) [KHAN; MCLEOD; HOVY, 2004]. Em muitos casos, o conhecimento do usuário sobre um determinado domínio pode não corresponder à forma como os dados estão representados e, portanto, pode não existir uma resposta exata para suas consultas, sendo necessário obter aproximações que também satisfaçam aos requisitos do usuário. A expansão de consultas visa, então, modificar a consulta do usuário (consulta original), seja pela alteração de suas condições ou pela adição de termos relacionados, com o intuito de tornar as consultas mais eficazes e abrangentes.

Para lidar com problemas relacionados a conflitos semânticos em consultas como, por exemplo, termos sinônimos e homônimos, alguns sistemas de expansão de consultas têm utilizado ontologias como fonte de informação semântica de domínio para expandir consultas e obter resultados mais eficazes. Os mecanismos de inferência sobre ontologias auxiliam o processo de expansão, pois derivam informações que podem ser utilizadas para obter respostas semanticamente relevantes. Entretanto, as abordagens baseadas em ontologias tradicionais (*crisp*) não são capazes de representar nem inferir sobre informações imprecisas como a intensidade com a qual um conceito está relacionado a outro ou o quanto um conceito pertence a uma determinada classe, ignorando informações importantes para a obtenção de respostas aproximadas às consultas. Assim, ontologias que incorporam conceitos da lógica difusa têm sido utilizadas com o objetivo de aumentar a expressividade da representação de domínio e melhorar o processo de expansão.

A maior parte das abordagens de expansão de consultas baseada em ontologias difusas considera principalmente taxonomias difusas e relacionamentos de similaridade entre conceitos. Algumas delas não especificam como é feita a expansão de consultas ambíguas, enquanto outras se dedicam exclusivamente a tratar ambigüidades, fato que pode

comprometer tanto a precisão quanto a revocação das consultas. Deste modo, é interessante que um sistema de expansão de consultas seja capaz de efetuar diversos tipos de expansão semântica, com o intuito de obter resultados aproximados que sejam relevantes e também abrangentes para a consulta do usuário. Além disso, outros tipos de relacionamentos difusos poderiam ser explorados pelo processo de expansão como, por exemplo, relacionamentos transitivos difusos, que permitem inferir conceitos que estejam associados entre si de forma indireta, e relacionamentos todo-parte, que podem indicar conceitos semanticamente próximos entre si em função das partes que os constituem.

Desta forma, o objetivo deste trabalho foi a criação do sistema FOQuE (*Fuzzy Ontology-based Query Expansion*), desenvolvido para realizar diversos tipos de expansão semântica de consultas com base em ontologias difusas, a fim de obter resultados aproximados que sejam semanticamente relacionados aos requisitos do usuário. O sistema FOQuE é capaz de expandir consultas considerando relacionamentos todo-parte (proximidade todo-parte) e relacionamentos transitivos difusos, permitindo assim a recuperação de respostas relevantes que não seriam obtidas pelas consultas originais. Também são realizadas expansões de homônimos, classes e similaridade, com base em mecanismos de inferência sobre conceitos e relacionamentos difusos da ontologia. O sistema FOQuE possui, ainda, o diferencial de classificar as respostas adicionais conforme o tipo de expansão efetuada e a relevância para a consulta. Desta maneira, o usuário pode identificar quais as expansões semânticas realizadas e os dados mais significativos para sua consulta, de acordo com parâmetros de expansão definidos por ele.

## 1.2. ORGANIZAÇÃO DA DISSERTAÇÃO

Esta dissertação está organizada da seguinte maneira:

- no capítulo 2 são apresentados os principais conceitos de ontologias difusas, incluindo conceitos básicos de ontologias, conceitos da lógica difusa que foram incorporados às ontologias tradicionais e algumas abordagens existentes para representação de ontologias difusas;
- no capítulo 3 são descritas as características da expansão semântica de consultas baseada tanto em ontologias tradicionais quanto em ontologias difusas, sendo apresentados alguns trabalhos relevantes existentes na literatura;

- o capítulo 4 apresenta o sistema FOQuE, desenvolvido para realizar diversos tipos de expansão semântica de consultas com base em ontologias difusas;
- o capítulo 5 descreve os testes realizados considerando um protótipo do sistema FOQuE, além de uma discussão sobre os resultados obtidos;
- e finalmente, no capítulo 6, é apresentada a conclusão deste trabalho, descrevendo seus resultados, contribuições e trabalhos futuros.



## 2. ONTOLOGIAS DIFUSAS

### 2.1. CONSIDERAÇÕES INICIAIS

Para compreender a expansão de consultas baseada em ontologias difusas, é necessário, primeiramente, assimilar os conceitos básicos de ontologias, considerando suas formas de representação e principais aplicações. Como as ontologias utilizadas por este trabalho são ontologias difusas, também é importante entender os conceitos da lógica difusa que foram incorporados às ontologias tradicionais para possibilitar a representação de informação vaga ou imprecisa. Assim, o propósito deste capítulo é apresentar as definições e características relacionadas a ontologias difusas, assim como as principais abordagens existentes para representá-las.

### 2.2. ONTOLOGIAS

#### 2.2.1. Definição e caracterização

No sentido filosófico, ontologias referem-se a um sistema particular de categorias que descreve uma determinada visão de mundo [GUARINO, 1998]. Na comunidade de Ciência da Computação, uma das definições mais utilizadas é a de Gruber [GRUBER, 1993]: “Uma ontologia é uma especificação formal e explícita de uma conceitualização compartilhada”. Para melhor compreender tal definição, os autores Uschold & Grüninger [USCHOLD; GRÜNINGER, 2004] analisaram o significado de cada um de seus termos:

- *Conceitualização*: refere-se a um modelo abstrato de como as pessoas pensam sobre as coisas do mundo, geralmente restrito a alguma área particular;
- *Especificação explícita*: significa que definições e nomes explícitos são atribuídos aos conceitos e relacionamentos do modelo abstrato. Neste caso, o nome é um termo e a definição é uma especificação do significado do conceito ou relacionamento;

- *Formal*: significa que a especificação da conceitualização é codificada em uma linguagem cujas propriedades formais são bem compreendidas. A formalização é importante para remover a ambigüidade, que é prevalente na linguagem natural e em outras notações informais. Além disso, possibilita o uso de mecanismos de inferência automática para derivar novas informações a partir das especificações;
- *Compartilhada*: refere-se a um dos principais propósitos de uma ontologia – a utilização e reutilização em diferentes aplicações e comunidades.

Ontologias provêm um vocabulário de termos que se referem aos conceitos de um domínio específico (conceitualização) e alguma especificação do significado destes termos (especificação explícita), geralmente tendo por base alguma teoria formal (formalismo). Segundo Uschold & Grüninger [USCHOLD; GRÜNINGER, 2004], o que distingue as diferentes abordagens de ontologias é a forma de especificar o significado dos termos. A Figura 2.1 apresenta um espectro evolutivo que ilustra os tipos de representações que geralmente são considerados ontologias na literatura. À esquerda da Figura 2.1, existem representações simplificadas chamadas de “ontologias leves”, contendo apenas termos com pouca ou nenhuma especificação de significado. Conforme se segue para o outro lado da figura, o grau de formalização aumenta e a especificação do significado se torna mais precisa (reduzindo-se a ambigüidade). Por fim, no extremo direito do espectro, estão as teorias rigorosamente formalizadas, que compõem as ontologias “autênticas”.

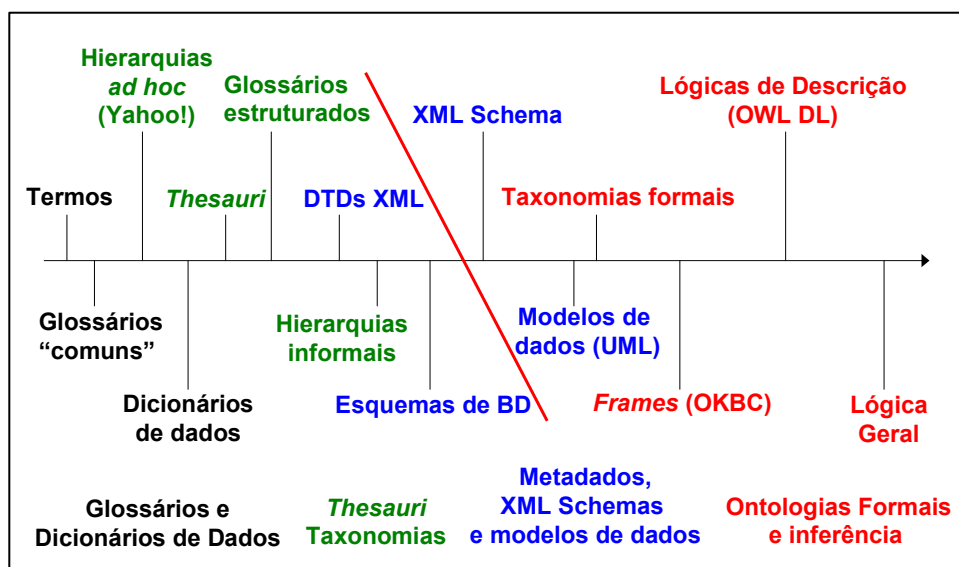


Figura 2.1 – Tipos de representação de ontologias [USCHOLD; GRÜNINGER, 2004].

De forma geral, para modelar adequadamente uma conceitualização, ontologias são constituídas por conceitos, atributos, relacionamentos, axiomas formais e instâncias.

Conceitos ou classes representam entidades que fazem parte de um domínio, tipicamente organizados em uma taxonomia de classes e subclasses. Atributos descrevem características dos conceitos e os relacionamentos expressam como os conceitos estão interligados. Atributos e relacionamentos também podem ser chamados de propriedades dos conceitos. Axiomas são restrições sobre conceitos e propriedades, cujo principal propósito é expressar significado de forma que máquinas sejam capazes de interpretá-lo por meio de mecanismos de raciocínio automático [USCHOLD; GRÜNINGER, 2004]. Por fim, as instâncias dos conceitos e das propriedades representam o conjunto de indivíduos ou objetos do mundo real que estão de acordo com a conceitualização modelada pela ontologia. A Figura 2.2 apresenta um exemplo de ontologia simples, contendo conceitos representados por elipses em branco, propriedades por setas preenchidas e valores de atributos por retângulos. A instanciação é feita por setas tracejadas e as instâncias possuem tom acinzentado. No exemplo, o relacionamento *parte\_de* possui a restrição do axioma de transitividade, permitindo inferir que instâncias de *Cidade* também fazem parte de instâncias de *País*.

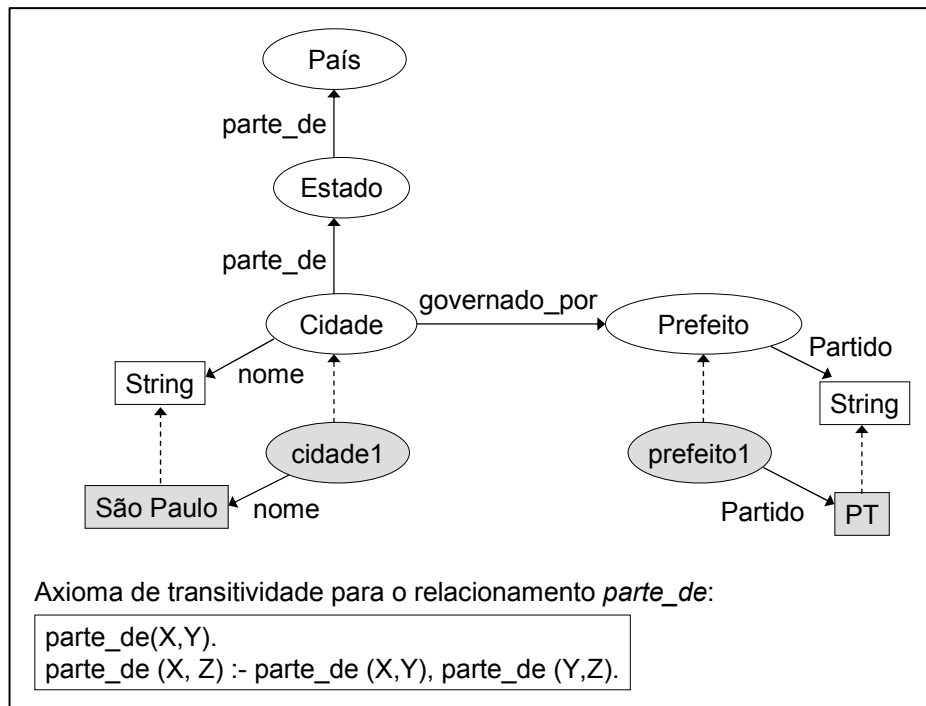


Figura 2.2 – Exemplo de uma ontologia simples.

### 2.2.2. Representação de ontologias

Um aspecto importante do desenvolvimento de ontologias é a escolha de uma forma de representação que seja expressiva o suficiente para capturar a conceitualização. Em

geral, ontologias são fundamentadas em teorias formais a fim de que humanos e máquinas sejam capazes de interpretar o conhecimento de domínio sem ambigüidade. Dentre os formalismos de representação existentes na literatura, as lógicas de descrição (*Description Logics* – DL) [NARDI; BRACHMAN, 2003] são consideradas uma das famílias mais importantes de representação formal do conhecimento, e constituem a base para linguagens de representação de ontologias como, por exemplo, a *Web Ontology Language* (OWL) [SMITH; WELTY; MCGUINESS, 2004].

As lógicas de descrição constituem um subconjunto da lógica de primeira ordem, que propõe a representação do conhecimento através de predicados. Uma base de conhecimento de lógica de descrição é composta por dois componentes – uma *TBox* (*Terminological Box*) e uma *ABox* (*Assertional Box*). A *TBox* contém conhecimento intencional na forma de uma terminologia e é construída através de declarações que descrevem propriedades e relacionamentos gerais dos conceitos. A *ABox* contém conhecimento extensível, ou seja, conhecimento específico sobre os indivíduos do domínio considerado [NARDI; BRACHMAN, 2003].

Cada lógica de descrição é equipada com um conjunto de construções que permitem obter expressões complexas de conceitos a partir de definições mais simples. Alguns exemplos dessas construções são operações de complemento, intersecção e união, restrições de propriedade (restrição existencial ( $\exists$ ), restrição de tipo, quantificação universal ( $\forall$ ), cardinalidade), axiomas (relacionamentos inversos, relacionamentos transitivos, equivalência, disjunção, generalização/especialização), entre outras [CALVANESE; DE GIACOMO, 2005]. As lógicas de descrição distinguem-se pelas construções que provêm. Assim, as diversas linguagens da família de lógicas de descrição *AL* (*Attributive Language*), como *SHIQ*, *SHOIQ*, *SHIN*, *SHIF* e *SHOIN*, diferem-se pela quantidade de construções que permitem modelar, sendo que quanto maior o número de construções, mais expressiva e complexa é a lógica de descrição.

Dentre as diversas linguagens baseadas no formalismo das lógicas de descrição, OWL é uma das linguagens mais difundidas por ser uma recomendação oficial do consórcio *World Wide Web Consortium* (W3C) para a criação de ontologias na *Web Semântica* [BERNERS-LEE; HENDLER; LASSILA, 2001]. OWL divide-se em três sub-linguagens, que diferem entre si quanto à expressividade e complexidade:

- *OWL Lite*: provê os elementos básicos para a representação de conceitos, relacionamentos e restrições simples de propriedades, sendo o dialeto de OWL menos expressivo e complexo. Em função de suas construções mais simples, facilita o desenvolvimento de ferramentas e máquinas de inferência que manipulam as ontologias;
- *OWL DL*: provê maior expressividade que OWL Lite, aliada à garantia de que as inferências sejam computáveis. Corresponde à lógica de descrição *SHOIN* [FIKES; HAYESB; HORROCKS, 2004], sendo uma das linguagens mais utilizadas para a criação de ontologias na *Web Semântica*;
- *OWL Full*: provê expressividade máxima sem garantias computacionais. Desta forma, ainda não foram desenvolvidas máquinas de inferência capazes de tratar todas as construções possíveis deste dialeto de OWL.

De forma geral, as linguagens baseadas em lógicas de descrição são adequadas para representar ontologias de forma expressiva, além de oferecerem suporte ao raciocínio lógico através de máquinas de inferência, possibilitando a verificação de consistência, a classificação correta das instâncias e a obtenção de novas informações a partir das especificações presentes nas ontologias.

### 2.2.3. Aplicações de ontologias

Dentro da Ciência da Computação, ontologias podem ser aplicadas das mais diversas formas. Em geral, são utilizadas em contextos de compartilhamento, reuso e organização de informação semântica, facilitando a compreensão e a comunicação do conhecimento de domínio entre humanos e sistemas computacionais.

Ontologias contribuem para a interoperabilidade entre sistemas [USCHOLD; GRÜNINGER, 2004], sendo utilizadas como um formato neutro para representar os dados compartilhados por diversos sistemas. Outra aplicação é a especificação de sistemas baseada em ontologias, em que ontologias são usadas como base para a especificação e desenvolvimento de software. O editor de ontologias *ODEd* [MIAN; FALBO, 2003] oferece suporte para a especificação baseada em ontologias, possibilitando a criação de uma infraestrutura em linguagem *Java* a partir dos elementos modelados em uma ontologia.

Uma das aplicações mais relevantes de ontologias é a estruturação semântica da informação (documentos, páginas *Web*, bancos de dados, entre outros). Neste contexto, ontologias são utilizadas para descrever a semântica de repositórios de informação e tornar o seu conteúdo explícito [WACHE *et al.*, 2001]. Assim, é possível tratar conflitos semânticos presentes nos dados e propiciar respostas mais abrangentes às consultas. Uma das aplicações de ontologias que segue esta idéia é a *Web Semântica* [BERNERS-LEE; HENDLER; LASSILA, 2001], que propõe o uso de ontologias e metadados para estruturar e dar significado ao conteúdo das páginas *Web*. A partir da organização semântica do conteúdo *Web*, agentes inteligentes podem processar e interpretar o conhecimento embutido nos documentos para realizar consultas e tarefas complexas.

Também seguindo na linha de aplicação de ontologias para a estruturação semântica da informação, este trabalho de mestrado baseia-se em ontologias difusas para expandir consultas realizadas a bancos de dados com o objetivo de recuperar resultados aproximados que sejam semanticamente relevantes aos usuários. Para compreender como esta expansão semântica de consultas é realizada, é necessário conhecer os elementos da lógica difusa que foram incorporados às ontologias tradicionais. Assim, a Seção 2.3 apresenta os conceitos básicos e as principais definições da lógica difusa.

## 2.3. LÓGICA DIFUSA

A Lógica Difusa ou Lógica Nebulosa (*Fuzzy Logic*) baseia-se na teoria de conjuntos difusos (*Fuzzy Sets*) introduzida por Lotfi A. Zadeh [ZADEH, 1987a]. Conjuntos difusos são usados para modelar informação vaga ou imprecisa presente no mundo real, permitindo inferir uma resposta aproximada para uma questão baseada em um conhecimento inexato ou incompleto. Enquanto a lógica booleana clássica define apenas dois valores possíveis — verdadeiro (1) ou falso (0) —, a lógica difusa considera um conjunto de valores possíveis que permite representar informação imprecisa de forma mais adequada.

### 2.3.1. Conjuntos difusos

A função característica de um conjunto clássico (*crisp*) pode assumir apenas os valores 1 ou 0, discriminando, desta forma, os elementos que pertencem ao conjunto daqueles

que não pertencem. Ao definir um conjunto, sua função característica pode ser generalizada de forma a assumir valores em um determinado intervalo, refletindo, assim, o grau com que um elemento pertence ao conjunto (*grau de pertinência*). Esta função chama-se *função de pertinência* e o conjunto definido por ela chama-se *conjunto difuso* [KLIR; YUAN, 1995].

Em geral, a função de pertinência associa elementos de um conjunto clássico  $X$  a números reais do intervalo  $[0,1]$ , que expressam a compatibilidade desses elementos com o conceito representado pelo conjunto difuso. Dado um conjunto difuso  $A$ , uma das notações mais comuns de funções de pertinência é:

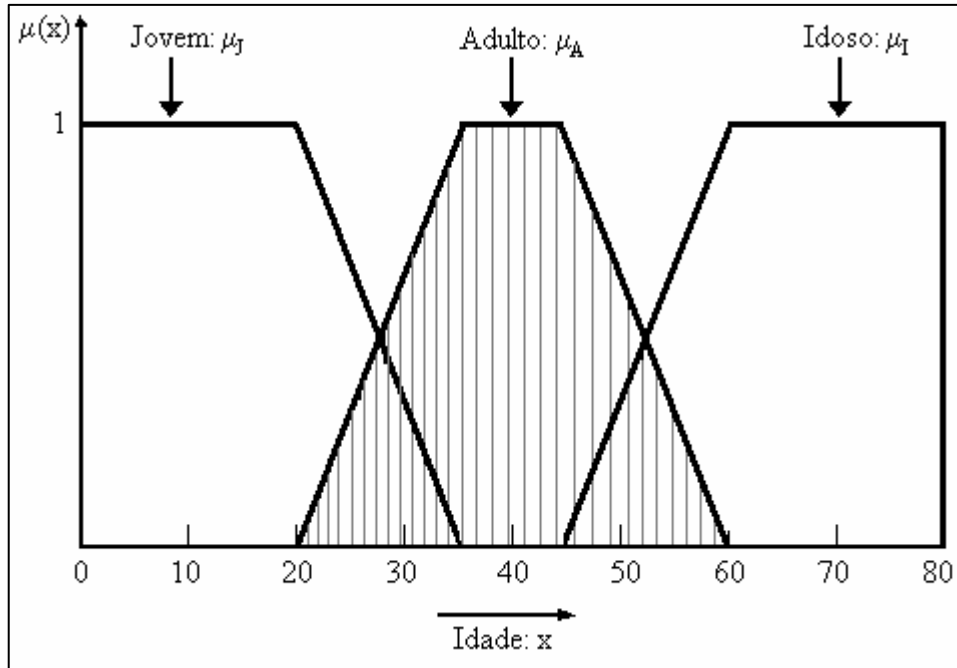
$$\mu_A : X \rightarrow [0,1]$$

Assim, o valor de  $\mu_A(x)$  representa o grau de pertinência do elemento  $x$  no conjunto difuso  $A$ . Quanto mais próximo o valor de  $\mu_A(x)$  for de 1, maior o grau de pertinência de  $x$  no conjunto  $A$ .

Um exemplo comum de conjuntos difusos, extraído de [KLIR; YUAN, 1995], define três conjuntos difusos que representam os conceitos *jovem*, *adulto* e *idoso* em função da idade de um ser humano. A Figura 2.3 exhibe as funções de pertinência definidas no intervalo real  $[0, 80]$  de idades para cada um desses conjuntos (respectivamente  $\mu_J(x)$ ,  $\mu_A(x)$  e  $\mu_I(x)$ ), cujas representações gráficas são ilustradas na Figura 2.4.

$\mu_J(x) = \begin{cases} 1 & \text{Quando } x \leq 20 \\ (35 - x)/15 & \text{Quando } 20 < x < 35 \\ 0 & \text{Quando } x \geq 35 \end{cases}$	
$\mu_A(x) = \begin{cases} 0 & \text{Quando } x \leq 20 \text{ ou } x \geq 60 \\ (x - 20)/15 & \text{Quando } 20 < x < 35 \\ (60 - x)/15 & \text{Quando } 45 < x < 60 \\ 1 & \text{Quando } 35 \leq x \leq 45 \end{cases}$	
$\mu_I(x) = \begin{cases} 0 & \text{Quando } x \leq 45 \\ (x - 45)/15 & \text{Quando } 45 < x < 60 \\ 1 & \text{Quando } x \geq 60 \end{cases}$	

**Figura 2.3 – Funções de pertinência para os conjuntos difusos *jovem*, *adulto* e *idoso* [KLIR; YUAN, 1995].**



**Figura 2.4 – Representação gráfica dos conceitos *jovem*, *adulto* e *idoso*.**

As definições da teoria de conjuntos clássicos também foram generalizadas para os conjuntos difusos [ZADEH, 1987a] [DUBOIS; PRADE, 1980]:

- *Conjunto vazio:*  $\forall x \in X, \mu_{\emptyset}(x) = 0$ ;
- *Igualdade:*  $\forall x \in X, A = B \Leftrightarrow \mu_A(x) = \mu_B(x)$ ;
- *Subconjunto difuso:*  $\forall x \in X, A \subseteq B \Leftrightarrow \mu_A(x) \leq \mu_B(x)$ ;
- *Complemento:*  $\forall x \in X, \mu_{\bar{A}}(x) = 1 - \mu_A(x)$ ;
- *União:*  $\forall x \in X, \mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$ ;
- *Intersecção:*  $\forall x \in X, \mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))$ .

O operador de implicação lógica ( $a \rightarrow b$ ) também foi estendido, resultando em vários operadores de implicação difusa, propostos por diversos pesquisadores. Alguns desses operadores são descritos por Chen [CHEN, 1998], que lista quinze tipos existentes de operadores de implicação difusa e seus correspondentes autores.

Os elementos de dois ou mais conjuntos difusos podem ser relacionados com um determinado grau que representa a intensidade da associação entre eles. Essas relações são chamadas de *relações difusas*, e são apresentadas na Seção 2.3.2.



### 2.3.2. Relações difusas

O conceito de relação difusa generaliza o conceito de relação clássica, pois permite modelar a intensidade com a qual os elementos estão associados entre si. Assim, dada uma relação difusa  $R$ , atribui-se um valor  $\mu_R(x,y)$  no intervalo  $[0,1]$  representando o grau com que os elementos  $x$  e  $y$  estão relacionados por  $R$ . As relações difusas podem envolver diversos elementos, e quando envolvem dois elementos são chamadas *relações difusas binárias*.

Algumas definições associadas às relações clássicas binárias também foram redefinidas para as relações difusas binárias [ZADEH, 1987b]:

- *Composição max-min ( $R \circ S$ ):*

$$R \subset X \times Y, S \subset Y \times Z, \mu_{R \circ S}(x, z) = \max_{y \in Y} \min [ \mu_R(x, y), \mu_S(y, z) ], x \in X, z \in Z$$

- *Propriedade reflexiva:*  $\forall x \in X, \mu_R(x, x) = 1$ ;
- *Propriedade simétrica:*  $\forall (x, y) \in X \times X, \mu_R(x, y) = \mu_R(y, x)$ ;
- *Propriedade transitiva (transitividade max-min):*

$$\forall (x, z) \in X \times X, \mu_R(x, z) \geq \max_{y \in X} \min [ \mu_R(x, y), \mu_R(y, z) ]$$

onde *max* é o operador de máximo (que obtém o valor máximo da relação) e *min* é o operador de mínimo (que obtém o valor mínimo da relação)

A expressão que define a transitividade *max-min* é resultado da composição de duas relações difusas, e no caso essa composição é a *max-min*. Alternativamente, outras definições de transitividade difusa são possíveis e úteis em algumas aplicações [KLIR; YUAN, 1995] [DUBOIS; PRADE, 1980].

Uma relação difusa binária que é reflexiva, simétrica e transitiva é conhecida como *relação difusa de equivalência* ou *relação de similaridade*. Segundo Zadeh [ZADEH, 1987b], “o conceito da relação de similaridade é essencialmente uma generalização do conceito da relação de equivalência”. Outro tipo de relação difusa binária que generaliza a relação de equivalência é a *relação de tolerância* ou *relação de proximidade* [DUBOIS; PRADE, 1980], que possui as propriedades reflexiva e simétrica. A relação de proximidade é considerada uma relação “menos restritiva” que a relação de similaridade, já que ela não atende à propriedade transitiva *max-min*. Ambas relações de similaridade e proximidade possuem respectivos graus de similaridade e de proximidade, que refletem a intensidade da

relação difusa entre os elementos relacionados. No contexto deste trabalho, estas relações desempenham um importante papel para a expansão semântica de consultas.

## 2.4. ONTOLOGIAS DIFUSAS

Conforme apresentado na Seção 2.2, ontologias tradicionais (*crisp*) são capazes de representar conceitos, propriedades, relacionamentos, instâncias e axiomas de um domínio de aplicação. Entretanto, estas ontologias seguem a teoria clássica (booleana) de conjuntos, capturando somente informações precisas ou completas. Existem alguns domínios em que a definição de conceitos, instâncias e relacionamentos é vaga ou imprecisa e, portanto, tais domínios não são representados adequadamente pelas ontologias tradicionais. Por exemplo, é difícil representar em ontologias *crisp* conceitos como “cremoso”, “escuro”, “quente”, “alto” ou “espesso”, para os quais é difícil uma definição clara e precisa [STRACCIA, 2006]. Assim, ontologias tradicionais têm sido estendidas para incorporar conceitos da lógica difusa, resultando em ontologias difusas que possibilitam representar e inferir conhecimento sobre informações imprecisas.

Dentre os trabalhos sobre ontologias difusas existentes na literatura, foram identificadas três abordagens principais, descritas nas seções 2.4.1, 2.4.2 e 2.4.3.

### 2.4.1. Taxonomias difusas

A abordagem mais simples para ontologias difusas considera taxonomias de termos relacionados por generalização / especialização (relacionamento *é-um*) contendo graus de pertinência entre os termos. Desta forma, um termo pode ser mais genérico ou mais específico que outro com um determinado valor, obtido de forma manual ou automática. As ontologias difusas utilizadas pelo sistema *Personalized Abstract Search Service* (PASS) [WIDYANTORO; YEN, 2001] e pela máquina de busca proposta por Parry [PARRY, 2004b] seguem esta abordagem, contendo graus de pertinência obtidos de forma semi-automática por meio da análise da ocorrência de termos em documentos do domínio considerado. Chen & Wei [CHEN; WEI, 2002] também utilizam uma taxonomia difusa durante o processo de mineração de regras de associação difusas, propondo extensões no cálculo do suporte e da confiança de regras que envolvem termos difusos. A Figura 2.5 mostra um exemplo de

taxonomia difusa adaptado de [CHEN; WEI, 2002], em que o termo *tomate* está relacionado a *Legume* e *Fruta*, porém com graus diferentes.

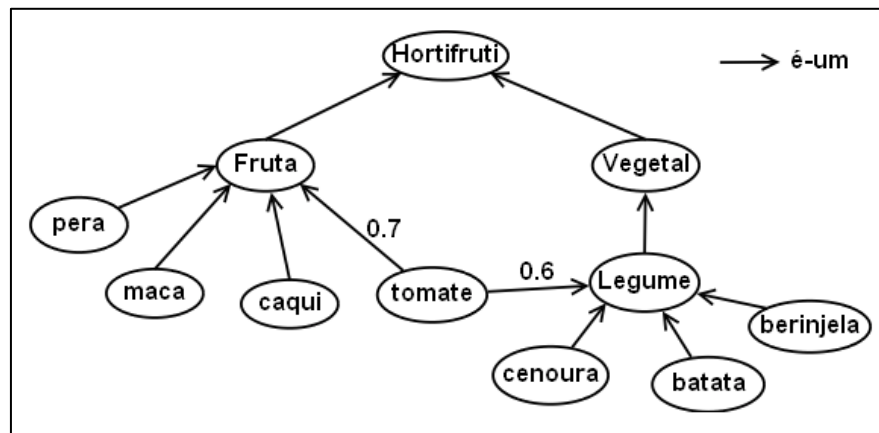


Figura 2.5 – Exemplo de taxonomia difusa.

Na Figura 2.5, termos estão relacionados entre si pelo relacionamento *é-um*, que pode conter um determinado grau no intervalo  $[0,1]$  como, por exemplo, *tomate é-uma Fruta* com grau 0.7 e *tomate é-um Legume* com grau 0.6. No caso dos relacionamentos em que o grau não é explicitado, seu valor é 1. Para obter o grau em relação aos ancestrais de um termo por transitividade, Chen & Wei [CHEN; WEI, 2002] propõem o uso da seguinte fórmula que se assemelha à transitividade *max-min*:

$$\mu(x, y) = \max_{\forall l: x \rightarrow y} [ \min_{\forall e \in l} \mu(l, e) ]$$

onde  $l: x \rightarrow y$  representa um dos caminhos entre  $x$  e  $y$ ,  $e$  em  $l$  representa uma aresta  $e$  do caminho  $l$  e  $\mu(l, e)$  corresponde ao grau da aresta  $e$  em  $l$ . Assim, *tomate é-um Hortifruti* com grau 0.7, pois existem dois caminhos possíveis entre *tomate* e *Hortifruti*: *tomate → Fruta → Hortifruti*, com grau mínimo do caminho = 0.7 e *tomate → Legume → Vegetal → Hortifruti*, com grau mínimo do caminho = 0.6, sendo então o  $\max(0.7, 0.6) = 0.7$ .

#### 2.4.2. Ontologias com classes e relacionamentos difusos

Outra abordagem para ontologias difusas considera ontologias que contêm classes, instâncias, propriedades e axiomas para representar a semântica de domínio. Nesta abordagem, o conceito de classe foi redefinido para classe difusa, correspondente ao conjunto difuso, e o de relacionamento para relacionamento difuso, equivalente à relação difusa. Desta forma, uma instância da ontologia pode pertencer a uma classe com um determinado grau de pertinência no intervalo  $[0,1]$ , assim como instâncias podem estar relacionadas entre si

contendo um valor que expressa a intensidade desse relacionamento. A Figura 2.6 apresenta um exemplo de ontologia difusa que segue esta abordagem, em que instâncias possuem graus de pertinência para as classes às quais elas pertencem, e o relacionamento *similar* representa uma relação de similaridade difusa.

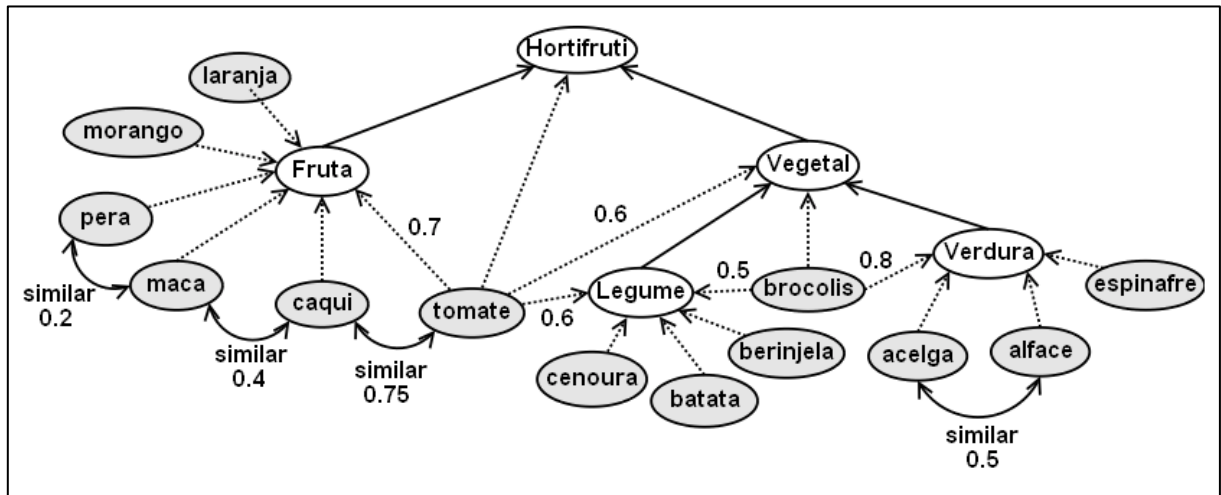


Figura 2.6 – Exemplo de ontologia difusa com classes e relacionamentos difusos.

Na Figura 2.6, as instâncias (elipses acinzentadas) estão relacionadas às classes (elipses em branco) por linhas tracejadas representando a instanciação das classes. O axioma de especialização/generalização entre classes é representado por linhas contínuas. A instanciação pode conter valores que representam o grau de pertinência das instâncias para as classes difusas e, caso o valor não esteja indicado, a pertinência tem valor máximo = 1. Por exemplo, a instância *tomate* pertence à classe *Fruta* com  $\mu_{Fruta}(tomate) = 0.7$ , à classe *Legume* com  $\mu_{Legume}(tomate) = 0.6$ , à classe *Vegetal* com  $\mu_{Vegetal}(tomate) = 0.6$  e à classe *Hortifruti* com  $\mu_{Hortifruti}(tomate) = 1$ . As instâncias também podem estar associadas por meio de relacionamentos difusos, como no caso do relacionamento *similar*, que contém valores que indicam a intensidade com a qual as instâncias são similares entre si. Vale ressaltar que no exemplo da Figura 2.6 o relacionamento *similar* considera a similaridade pela aparência física das frutas, podendo assumir valores diferentes caso outras características sejam analisadas.

Algumas definições da lógica difusa são respeitadas pelas ontologias difusas que seguem esta abordagem, como a intersecção ( $\forall x, \mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))$ ), a união ( $\forall x, \mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$ ), o complemento ( $\forall x, \mu_{\bar{A}}(x) = 1 - \mu_A(x)$ ), a equivalência entre classes difusas ( $\forall x, A = B \Leftrightarrow \mu_A(x) = \mu_B(x)$ ), especialização de classes como subconjuntos difusos ( $\forall x, A \subseteq B \Leftrightarrow \mu_A(x) \leq \mu_B(x)$ ) e relacionamentos transitivos (transitividade *max-min*). A partir destas definições, é possível realizar inferências sobre as

informações contidas na ontologia. Por exemplo, como *Fruta* é subclasse de *Hortifruti*, então  $\mu_{Fruta}(x) \leq \mu_{Hortifruti}(x)$  para todas instâncias  $x$  de ambas as classes. No caso das instâncias *laranja*, *maçã*, *morango*, *pêra* e *caqui*, pode-se inferir que  $\mu_{Hortifruti} = 1$ , já que  $\mu_{Fruta}(x) \leq \mu_{Hortifruti}(x)$  e  $\mu_{Fruta} = 1$  (valor máximo possível) para todas estas instâncias.

Relacionamentos transitivos também permitem realizar inferências através da transitividade *max-min*. Na ontologia difusa da Figura 2.6, o relacionamento *similar* corresponde a uma relação de similaridade da lógica difusa e, portanto, possui as propriedades reflexiva, simétrica e transitiva. Desta forma, pode-se inferir que  $similar(pêra, tomate) \geq 0.2$ , uma vez que é verificada a transitividade *max-min* em  $similar(pêra, maçã) = 0.2$ ,  $similar(maçã, caqui) = 0.4$  e  $similar(caqui, tomate) = 0.75$ .

Existem na literatura algumas propostas de ontologias difusas que contêm classes e relacionamentos difusos. *f-SHIN* [STOILLOS *et al.*, 2005] estende a lógica de descrição *SHIN* e permite realizar algumas inferências sobre classes e relacionamentos difusos através do protótipo de máquina de inferência *FiRE* [STOILLOS *et al.*, 2006]. *FOWL* [GAO; LIU, 2005] e *Fuzzy OWL* [STOILLOS *et al.*, 2006] adicionam conceitos de lógica difusa à linguagem OWL, porém ainda não possuem máquinas de inferências disponíveis para apoiar o raciocínio sobre conceitos e relacionamentos difusos. A Figura 2.7 exemplifica como *Fuzzy OWL* estende OWL para representar classes e relacionamentos difusos. No exemplo, a instância *o1* pertence à classe difusa *Red* com  $\mu_{Red}(o1) \geq 0.8$  e está associada à instância *o2* pelo relacionamento difuso *hasPart* com  $\mu_{hasPart}(o1, o2) \geq 0.75$ .

OWL	Fuzzy OWL
<pre>&lt;rdf:Description rdf:about="o1"&gt;   &lt;rdf:type rdf:resource="Red"/&gt;   &lt;hasPart rdf:resource="o2"/&gt; &lt;/rdf:Description&gt;</pre>	<pre>&lt;rdf:Description rdf:about="o1"&gt;   &lt;rdf:type rdf:resource="Red"     owlx:ineqType="≥"     owlx:degree="0.8"/&gt;   &lt;hasPart rdf:resource="o2"     owlx:ineqType="≥"     owlx:degree="0.75"/&gt; &lt;/rdf:Description&gt;</pre>

Figura 2.7 – Representação de classe e relacionamento difusos em *Fuzzy OWL* [STOILLOS *et al.*, 2006].

### 2.4.3. Ontologias com hierarquia difusa de classes e relacionamentos

Straccia [STRACCIA, 2006] propõe outra abordagem para ontologias difusas que estende a abordagem descrita na Seção 2.4.2. A principal diferença está na forma com que a abordagem de Straccia trata a especialização / generalização, permitindo que haja graus de pertinência em hierarquias de classes e de relacionamentos difusos. Assim, uma classe  $A$  pode ser subclasse de  $B$  com  $\mu_{A \subseteq B}$  e um relacionamento  $R$  pode ser sub-relacionamento de  $S$  com  $\mu_{R \subseteq S}$ .

Straccia propõe o uso de implicação difusa para obter o grau de pertinência em hierarquias de classes e de relacionamentos:

- $A$  subclasse de  $B$  ( $A \subseteq B$ ):  $\forall x, \mu_{A \subseteq B} = \inf_x(\mu_A(x) \rightarrow \mu_B(x))$ ;
- $R$  sub-relacionamento de  $S$  ( $R \subseteq S$ ):  $\forall x, y, \mu_{R \subseteq S} = \inf_{x, y}(\mu_R(x, y) \rightarrow \mu_S(x, y))$ .

sendo que  $\inf$  representa o valor ínfimo (mínimo) da implicação difusa aplicada às instâncias da ontologia.

Considerando a Figura 2.6, é possível obter, por exemplo, o grau de pertinência entre as classes difusas *Legume* e *Vegetal*, com base nas pertinências de suas instâncias. Caso seja adotada a implicação difusa *Kleene-Dienes*:  $\mu_A(x) \rightarrow_{KD} \mu_B(x) = \max(1 - \mu_A(x), \mu_B(x))$ , obtém-se  $\mu_{Legume \subseteq Vegetal} = \inf(\max(0.4, 0.6), \max(0, 1), \max(0, 1), \max(0, 1), \max(0.5, 1), \max(1, 1), \max(1, 1), \max(1, 1)) = 0.6$ . Outros tipos de operadores de implicação difusa podem ser utilizados a fim de obter resultados mais coerentes com a semântica do domínio considerado. Chen [CHEN, 1998] lista quinze tipos de operadores de implicação difusa existentes que podem ser adotados no cálculo do grau de pertinência em hierarquias de classes e relacionamentos difusos.

Para realizar inferências, Straccia propôs o sistema *fuzzy DL* [STRACCIA, 2007], que estende a lógica de descrição *SHIF* com conceitos de lógica difusa. *fuzzy DL* possui uma máquina de inferência capaz de verificar graus de pertinência em hierarquias de classes e relacionamentos, característica que diferencia este sistema das demais máquinas de inferência existentes para raciocínio sobre classes e relacionamentos difusos em ontologias. No entanto, *fuzzy DL* ainda não possui uma versão estável, pois o sistema está em fase de desenvolvimento e testes. Além disso, os mecanismos de inferência de *fuzzy DL* são mais restritos quando comparados à expressividade de *SHOIN* (lógica de descrição correspondente

a OWL DL), visto que *SHIF* é menos expressiva que *SHOIN* [HORROCKS; PATEL-SCHNEIDER, 2003].

## 2.5. CONSIDERAÇÕES FINAIS

Neste capítulo foram apresentados os principais aspectos envolvendo ontologias difusas, incluindo definições e características básicas de ontologias tradicionais, conceitos de lógica difusa que foram incorporados às ontologias e principais abordagens existentes para representação de informação imprecisa utilizando ontologias. Estes conceitos são importantes para compreender o papel das ontologias difusas dentro do contexto da expansão semântica de consultas proposta neste trabalho de mestrado.

No próximo capítulo, serão apresentadas as principais características da expansão semântica de consultas baseada em ontologias, considerando tanto ontologias tradicionais quanto ontologias difusas, além da descrição de alguns trabalhos existentes na literatura.

## 3. EXPANSÃO SEMÂNTICA DE CONSULTAS BASEADA EM ONTOLOGIAS

### 3.1. CONSIDERAÇÕES INICIAIS

Conforme aumenta a facilidade de acesso aos dados, também cresce a necessidade por formas de recuperação que obtenham dados relevantes às consultas. Em muitos casos, não existe uma resposta exata para uma consulta, sendo necessário obter aproximações que também satisfaçam aos requisitos do usuário. A expansão de consultas visa, então, adicionar termos ou modificar a consulta original com o objetivo de tornar as consultas mais eficazes e abrangentes. Para lidar com problemas relacionados a conflitos semânticos como, por exemplo, termos sinônimos e homônimos, alguns sistemas têm adotado ontologias de domínio durante o processo de expansão de consultas. No entanto, como são utilizadas ontologias *crisp*, as expansões não consideram informações imprecisas como, por exemplo, a intensidade com a qual um conceito está relacionado a outro ou o grau de pertinência de uma instância da ontologia para uma determinada classe.

Diante destas limitações, alguns sistemas de expansão de consulta passaram a incorporar conceitos da lógica difusa em ontologias, com o intuito de tornar a semântica de domínio mais expressiva e melhorar o processo de expansão. Assim, o objetivo deste capítulo é descrever as principais características da expansão semântica de consultas baseada tanto em ontologias tradicionais quanto em ontologias difusas, além de apresentar alguns trabalhos relevantes existentes na literatura.

### 3.2. MOTIVAÇÃO E OBJETIVOS

A crescente disponibilidade de dados, principalmente na *Web*, facilita o acesso à informação pertencente a diversas áreas do conhecimento. Diante da oferta abundante de dados, cada vez mais são necessárias técnicas eficazes para busca e recuperação de informação, de modo a reduzir a quantidade de informação irrelevante (alta precisão –



*precision*) e assegurar que informação relevante não seja desprezada (alta revocação – *recall*) [KHAN; MCLEOD; HOVY, 2004].

De forma geral, as técnicas para busca e recuperação de informação consideram o casamento léxico de termos especificados pelo usuário com termos que descrevem os dados a serem recuperados. Entretanto, dados relevantes à consulta podem não conter exatamente os mesmos termos requisitados pelo usuário, sendo necessário, portanto, obter respostas aproximadas que também satisfaçam aos requisitos do usuário. Para atingir tal objetivo, a expansão de consultas visa modificar a consulta original, seja pela alteração de suas condições ou pela adição de termos relacionados aos termos especificados pelo usuário, com o intuito de aumentar tanto a revocação quanto a precisão das respostas.

Uma abordagem clássica para expansão de consultas baseia-se na co-ocorrência estatística de palavras-chave em documentos do domínio considerado, adicionando à consulta termos que freqüentemente ocorrem em conjunto com os termos da consulta original. Segundo Peat & Willett [PEAT; WILLETT, 1999], os sistemas que seguem esta abordagem até são capazes de aumentar a revocação, porém tendem a deteriorar a precisão, pois não consideram a semântica dos termos durante a verificação das ocorrências. Assim, para lidar com os problemas das técnicas baseadas em co-ocorrência de palavras-chave, diversos trabalhos têm utilizado ontologias como fonte de informação semântica de domínio para expandir consultas e obter resultados mais eficazes. A Seção 3.3 descreve as principais características de alguns trabalhos voltados para este tipo de expansão de consultas – a expansão semântica de consultas baseada em ontologias.

### 3.3. ABORDAGENS DE EXPANSÃO DE CONSULTAS BASEADA EM ONTOLOGIAS TRADICIONAIS

Em geral, a expansão semântica de consultas baseada em ontologias analisa conceitos e relacionamentos de um domínio específico para adicionar termos à consulta original ou alterar suas condições. Assim, a consulta é modificada com base nos conceitos da ontologia que estejam semanticamente associados aos termos especificados pelo usuário. Para obter tais conceitos, são aplicados mecanismos de raciocínio automático sobre ontologias, permitindo que informações inferidas sejam consideradas durante o processo de expansão. Segundo Guarino, Masolo & Vetere [GUARINO; MASOLO; VETERE, 1999], o uso de

ontologias e mecanismos de inferência no processo de recuperação implica que os resultados recuperados sejam relevantes para a consulta do usuário. Além disso, Guarino, Masolo & Vetere declaram que a revocação também é melhorada, uma vez que o raciocínio sobre ontologias possibilita a derivação de novas informações relevantes com base no conhecimento de domínio. Desta forma, os resultados obtidos com o uso de ontologias tendem a ser melhores que aqueles recuperados pela abordagem baseada na co-ocorrência estatística de palavras-chave em documentos do domínio. Diante destas razões, muitos sistemas têm utilizado ontologias para realizar a expansão de consultas, com o intuito de obter resultados mais eficazes.

O sistema PICSEL [BIDAULT; FROIDEVAUX; SAFAR, 2000] é um exemplo de sistema de integração que se baseia em ontologias para modificar consultas. PICSEL provê a reparação de consultas, que consiste em reformular consultas do usuário que violam restrições de integridade e, portanto, não retornam resultados. Os conceitos da consulta são generalizados com base na hierarquia de generalização / especialização da ontologia, permitindo que seja construída uma consulta semanticamente próxima à consulta original, mas que não viole as restrições das fontes de dados. Deste modo, PICSEL é capaz de sugerir uma nova consulta (reparo) que pode retornar resultados semanticamente próximos aos desejados.

Embora PICSEL considere somente a generalização de conceitos para reparar consultas, ontologias possibilitam que outros tipos de expansões sejam realizados. Existem alguns trabalhos que propõem um processo de expansão de consulta mais abrangente, que não se limita a analisar taxonomias de conceitos e sinônimos, considerando também outros tipos de relacionamentos semânticos e casos de ambigüidade em consultas. O *framework* proposto por Necib & Freytag [NECIB; FREYTAG, 2004] [NECIB; FREYTAG, 2005a] [NECIB; FREYTAG, 2005b] aplica uma série de regras semânticas para expandir consultas com base em uma ontologia de domínio:

- *Regra de Vocabulário (Vocabulary Rule)*: reformula a condição de uma consulta ao utilizar termos semanticamente equivalentes derivados de uma ontologia (uso de sinônimos e especialização de conceitos). Por exemplo, a consulta em que o usuário deseja saber o preço do produto *computador* pode ser expandida utilizando-se os termos *micro-computador* (sinônimo), *pc* e *notebook* (especializações do conceito);

- *Regra Todo-parte (Part-whole Rule)*: baseia-se em relacionamentos todo-parte para reescrever a condição da consulta considerando elementos que são constituídos pelo mesmo conjunto de partes. Por exemplo, dada uma ontologia que descreva produtos constituídos por componentes (relacionamento todo-parte), a consulta em que o usuário deseja saber o preço do produto *pc* pode ser expandida para também obter todos os produtos que são constituídos pelo mesmo conjunto de componentes que *pc*;
- *Regra de Suporte (Support Rule)*: usa o axioma de transitividade (especificamente em relacionamentos todo-parte) para inferir informações adicionais que podem ser relevantes para o usuário. Por exemplo, considerando o domínio de biologia molecular, caso o usuário consulte por elementos constituídos por *carbono*, podem ser recuperados elementos constituídos por outros elementos que contenham *carbono*, embora o banco de dados não represente explicitamente esta informação;
- *Regra de Característica (Feature Rule)*: explora relacionamentos específicos de domínio para reescrever as consultas e obter informações adicionais. Mecanismos de raciocínio analisam as regras específicas de domínio para inferir novas informações e utilizá-las na expansão da consulta. Por exemplo, dada uma ontologia que descreva frutas, uma regra de domínio pode estabelecer que maçãs vermelhas são maçãs de cor vermelha, assim, caso o usuário deseje buscar as instâncias de *maçã vermelha*, a consulta pode ser expandida para recuperar instâncias de *maçã* associadas ao atributo cor *vermelha*;
- *Regra de Sensitividade (Sensitivity Rule)*: baseia-se na ontologia para identificar consultas ambíguas (consultas contendo termos homônimos) e apresentar ao usuário os diferentes contextos envolvidos de forma que ele seja capaz de escolher qual o contexto mais apropriado. Por exemplo, dada uma consulta pelo termo *banco*, dois contextos podem ser identificados a partir da ontologia: *banco* referente ao conceito de agência bancária ou *banco* relativo ao conceito de assento coletivo. Ambos os contextos são apresentados ao usuário, que é responsável por decidir qual deles é o mais adequado para sua consulta.

Outro sistema que explora diversos tipos de relacionamentos semânticos ao realizar as consultas é o sistema ROSA<sup>+</sup> [MATTOS; MOURA; CAVALCANTI, 2006].

ROSA<sup>+</sup> é capaz de obter resultados abrangentes para as consultas do usuário, pois se baseia em mecanismos de inferência sobre axiomas da ontologia (transitividade, simetria, inversão e igualdade) e relacionamentos específicos de domínio representados por regras especificadas em *Semantic Web Rule Language* (SWRL). Além disso, o modelo de dados de ROSA<sup>+</sup> permite representar os dados em diferentes níveis de abstração, sendo possível consultar objetos de aprendizagem (*Learning Objects* – LO), relacionamentos e regras em nível de Mapa Conceitual (instâncias), nível de Mapa de Domínio (esquema do mapa conceitual e regras) e em nível de Modelo ROSA<sup>+</sup> (definição dos formalismos do sistema, tais como LO, Relacionamento e Regra). Assim, o objetivo de ROSA<sup>+</sup> é disponibilizar ao usuário mecanismos de consulta que lhe permitam obter tanto conhecimento explícito quanto implícito, sendo este obtido através de mecanismos de raciocínio lógico sobre propriedades de relacionamentos da ontologia (representados na linguagem OWL DL) e sobre regras de domínio (representadas em SWRL).

Alguns trabalhos de expansão de consultas possuem um enfoque voltado para o tratamento de consultas ambíguas, como as propostas de Khan, McLeod & Hovy [KHAN; MCLEOD; HOVY, 2004] e Stojanovic [STOJANOVIC, 2005]. A abordagem de Khan, McLeod & Hovy [KHAN; MCLEOD; HOVY, 2004] expande consultas com base em sinônimos e especializações de conceitos da ontologia, além de identificar automaticamente qual o contexto desejado pelo usuário em consultas ambíguas. Para tratar a ambigüidade, Khan, McLeod & Hovy propõem o cálculo de proximidade semântica entre termos homônimos e os demais termos especificados pelo usuário na consulta. A proximidade semântica é obtida a partir da comparação de palavras-chave da requisição do usuário com os sinônimos dos conceitos, e da análise do caminho entre os conceitos da ontologia em termos da quantidade de arestas (relacionamentos) entre eles. Quanto maior a semelhança entre as palavras-chave e mais curto o caminho entre os conceitos da ontologia, maior a proximidade semântica entre os conceitos. Desta forma, o contexto mais apropriado é obtido com base no termo homônimo que possui a maior proximidade semântica com os demais termos da consulta. Stojanovic [STOJANOVIC, 2005] também trata ambigüidade em consultas do usuário, através de um agente de software que calcula métricas de ambigüidade baseadas na generalização de conceitos e nos relacionamentos de ontologias. Assim, dada uma consulta do usuário, o agente mede a sua ambigüidade e, caso seja alta, sugere ao usuário de forma interativa qual a consulta reformulada mais adequada.

De forma geral, os trabalhos que expandem consultas com base em ontologias são capazes de obter respostas mais eficazes que as abordagens baseadas em palavras-chave. No entanto, como são utilizadas ontologias tradicionais (*crisp*), as expansões não consideram informações imprecisas como, por exemplo, a intensidade com a qual um conceito está relacionado a outro ou o grau de pertinência de uma instância da ontologia para uma determinada classe. Além disso, os sistemas mencionados nesta seção recuperam as respostas adicionais sem justificar o motivo pelo qual foram retornadas e não classificam os resultados conforme a sua relevância para a consulta do usuário. Diante destas limitações, alguns sistemas de expansão de consulta passaram a incorporar conceitos da lógica difusa em ontologias, com o intuito de tornar a semântica de domínio mais expressiva e permitir a classificação dos resultados de acordo com a relevância das respostas para as consultas. A Seção 3.4 apresenta alguns sistemas existentes na literatura que aplicam a expansão semântica de consultas baseada em ontologias difusas.

### 3.4. ABORDAGENS DE EXPANSÃO DE CONSULTAS BASEADA EM ONTOLOGIAS DIFUSAS

Como a expansão de consultas baseada em ontologias tradicionais não é capaz de lidar com informações imprecisas como, por exemplo, a intensidade com a qual conceitos estão relacionados entre si, alguns sistemas têm utilizado ontologias que incluem conceitos de lógica difusa, a fim de aumentar a expressividade da representação de domínio. A partir do uso de ontologias difusas, o processo de expansão pode considerar graus de relacionamentos e conceitos difusos para classificar ou ordenar os resultados das consultas expandidas, melhorando assim a forma como as respostas são apresentadas ao usuário.

Existem alguns sistemas que utilizam taxonomias difusas para expandir as consultas, a exemplo do sistema PASS (*Personalized Abstract Search Services*) [WIDYANTORO; YEN, 2001] e do sistema proposto por Parry [PARRY, 2004a] [PARRY, 2004b]. O sistema PASS tem como objetivo refinar consultas a resumos de artigos na *Web* ao sugerir uma lista de termos mais generalizados ou mais específicos que os termos fornecidos pelo usuário. A taxonomia difusa utilizada por PASS é obtida de forma semi-automática pela análise de co-ocorrência estatística de palavras-chave em documentos (publicações acadêmicas), juntamente com o *feedback* do usuário. Assim, embora PASS use a técnica

baseada em palavras-chave, a taxonomia difusa pode evoluir de acordo com a opinião do usuário sobre as respostas retornadas.

Parry [PARRY, 2004a] [PARRY, 2004b] propõe o uso de taxonomias difusas para tratar consultas ambíguas, respeitando as preferências dos usuários. Para tanto, cada termo homônimo possui graus de pertinência que variam de acordo com o perfil do usuário ou do grupo de usuários, sendo recuperados apenas os dados do contexto com maior grau de pertinência, dependendo de cada perfil de usuário. A obtenção dos graus de pertinência também é feita por técnicas baseadas em palavras-chave e *feedback* do usuário, sendo que apenas conceitos homônimos possuem graus de pertinência para os perfis de usuários. Desta forma, o sistema proposto por Parry somente expande consultas com ambigüidade, não levando em consideração outros tipos de expansões.

Além de considerar hierarquias de generalização/especialização de conceitos, algumas abordagens utilizam outros tipos de relacionamentos difusos como, por exemplo, relacionamentos de similaridade, para expandir as consultas e obter melhores resultados. Um exemplo é o sistema MIEL [BUCHE *et al.*, 2005], que combina taxonomias difusas e similaridade entre conceitos para expandir consultas conforme as preferências do usuário. A taxonomia difusa contém conceitos que representam o domínio dos atributos do banco de dados, além de graus de pertinência que expressam quanto um conceito é mais generalizado que outro. Com base nos elementos desta taxonomia difusa, usuários definem conjuntos difusos de preferências, cujos graus de pertinência representam a relevância dos elementos para o usuário. Assim, MIEL expande as consultas ao generalizar o conjunto de preferências, inferindo os graus de pertinência dos conceitos resultantes da generalização a partir da similaridade entre os conceitos e dos graus de pertinência dos elementos originalmente pertencentes ao conjunto de preferências. O cálculo de similaridade baseia-se nos graus de generalização entre os conceitos e nos caminhos (em termos de quantidade de relacionamentos) entre eles, de modo que conceitos muito genéricos não sejam considerados na expansão. No final do processo, o conjunto difuso de preferências contém, além dos elementos especificados pelo usuário, os conceitos generalizados e respectivos graus de pertinência inferidos por MIEL. Deste modo, a consulta é expandida pelo conjunto de preferências estendido com base na taxonomia difusa. Em comparação com o sistema proposto por Parry, MIEL realiza generalizações sobre quaisquer conceitos da taxonomia difusa, não se restringindo apenas a conceitos homônimos. Por outro lado, MIEL não

apresenta formas de tratar consultas ambíguas e, portanto, pode eventualmente obter resultados com baixa precisão.

*Corese* [CORBY *et al.*, 2006] é outro sistema que se baseia em taxonomias para determinar a similaridade entre conceitos. *Corese* utiliza uma extensão de ontologias em OWL Lite para expandir consultas especificadas em linguagem de consulta *Corese*, que contém comandos especiais para indicar cada tipo de expansão desejada. As expansões são realizadas através da especialização de conceitos, similaridade e proximidade estrutural:

- *Especialização de conceitos*: *Corese* expande as subclasses de um conceito através do operador  $\leq$ : aplicado nas condições das consultas. Por exemplo, a Figura 3.1 apresenta a condição  $?person \leq c:Person$ , que permite recuperar todas as instâncias das subclasses de *Person*;
- *Similaridade entre conceitos*: Assim como MIEL, a similaridade entre conceitos baseia-se em uma hierarquia difusa, a partir do cálculo da distância ontológica (*ontological distance*) com base nos níveis de profundidade da taxonomia. Conceitos mais especializados (profundidade maior) são considerados mais similares entre si que conceitos mais generalizados (profundidade menor). Para cada relacionamento da hierarquia de generalização/especialização, é atribuído um grau  $(1/2)^d$ , onde  $d$  é a profundidade do ancestral na hierarquia. Em casos de herança múltipla,  $d$  é igual à profundidade do ancestral mais profundo (mais especializado). O comprimento do caminho entre um conceito e um ancestral é então definido pela soma dos graus de cada relacionamento de generalização/especialização entre eles. Assim, dados dois conceitos  $c1$ ,  $c2$  e um ancestral em comum  $t$ , a distância ontológica entre  $c1$  e  $c2$  é definida como o mínimo das somas dos caminhos de  $c1$  a  $t$  e de  $c2$  a  $t$ . Relacionamentos *rdf:seeAlso* reduzem a distância entre dois conceitos à distância entre irmãos, permitindo representar conceitos similares que estejam em locais distantes na hierarquia. A expansão por similaridade é realizada ao adicionar a palavra-chave *more* na cláusula *select* da consulta, sendo possível obter os conceitos cuja distância ontológica não exceda um parâmetro máximo especificado pelo usuário. O exemplo da Figura 3.1 mostra uma consulta contendo *more* em *select*, permitindo consultar por conceitos similares a *KnowledgeEngineering* e a *Project*.

```

select more where
?person c:interestedIn ?k
?person <=: c:Person
?k rdf:type c:KnowledgeEngineering
?person c:member ?project
?project rdf:type c:Project

```

Figura 3.1 – Exemplo de consulta em *Corese* [CORBY *et al.*, 2006].

- *Proximidade estrutural*: A expansão por proximidade estrutural considera a quantidade de ligações (relacionamentos) entre conceitos, não importando quais os tipos destas ligações. Assim, *Corese* considera caminhos de relacionamentos entre conceitos da ontologia, sendo que o usuário fornece qual o tamanho máximo do caminho desejado. A Figura 3.2 apresenta um exemplo de consulta contendo a condição `?org all::c:relation{2} ?topic`, que possibilita recuperar conceitos de *Organization* que estejam ligados a conceitos de *HumanScience* por até dois relacionamentos quaisquer entre eles.

```

?org all::c:relation{2} ?topic
?org rdf:type c:Organization
?topic rdf:type c:HumanScience

```

Figura 3.2 – Exemplo de consulta por proximidade estrutural [CORBY *et al.*, 2006].

Algumas abordagens possuem um foco voltado para a obtenção automática das ontologias difusas, ao estenderem ontologias tradicionais com graus que indicam a intensidade dos relacionamentos. Por exemplo, as abordagens propostas por Abulaish & Dey [ABULAISH; DEY, 2005] e por Lee, Jian & Huang [LEE; JIAN; HUANG, 2005] concentram-se na extração de relacionamentos difusos com base em técnicas de mineração de documentos específicos do domínio (*text mining*). Assim, os graus de relacionamento obtidos automaticamente são utilizados na expansão das consultas e na classificação dos resultados nos domínios de biologia molecular e de notícias meteorológicas, respectivamente. Em ambos trabalhos, as inferências sobre relacionamentos transitivos são feitas apenas sobre as ontologias tradicionais, para descobrir quais conceitos estão relacionados por transitividade e então obter os respectivos graus de relacionamento através de técnicas de mineração de texto. Desta forma, não é garantido que a transitividade difusa *max-min* seja respeitada. Além disso, ambos trabalhos não mencionam como é feito o tratamento de consultas contendo termos homônimos, fato que pode afetar a precisão dos resultados no caso de consultas ambíguas.



### 3.5. CONSIDERAÇÕES FINAIS

Ao longo do capítulo 3 foram apresentados os principais conceitos relacionados à expansão semântica de consultas baseada em ontologias. Inicialmente, foram descritas as motivações para expandir consultas e o papel que ontologias desempenham no processo de expansão. Posteriormente, foram citadas algumas abordagens que utilizam ontologias tradicionais e suas limitações devido ao uso de ontologias *crisp* para representar o conhecimento de domínio. Tais limitações conduziram à incorporação de conceitos da lógica difusa nas ontologias utilizadas para a expansão de consultas, resultando em sistemas de expansão de consultas baseada em ontologias difusas. Por fim, alguns destes sistemas foram descritos, detalhando-se o modo como representam e manipulam informações imprecisas para melhorar o processo de expansão de consultas.

No próximo capítulo será apresentado o sistema FOQuE, desenvolvido no escopo deste trabalho para realizar diversos tipos de expansão de consultas com base em ontologias difusas. Serão descritas as principais características deste sistema, juntamente com exemplos que ilustram como as consultas são expandidas e quais os resultados obtidos.

## 4. SISTEMA FOQuE PARA EXPANSÃO DE CONSULTAS BASEADA EM ONTOLOGIAS DIFUSAS

### 4.1. CONSIDERAÇÕES INICIAIS

Conforme apresentado ao longo do capítulo 3, a expansão de consultas baseada em ontologias possui um papel importante para a recuperação de respostas semanticamente relacionadas à consulta do usuário. Mecanismos de inferência sobre ontologias auxiliam o processo de expansão, pois derivam informações que podem ser utilizadas para obter respostas relevantes. Entretanto, as abordagens baseadas em ontologias tradicionais não são capazes de representar nem inferir sobre informações imprecisas como, por exemplo, graus de pertinência ou de similaridade entre conceitos, que contribuem para tornar a representação de domínio mais expressiva. Assim, o uso de ontologias difusas possibilita que o processo de expansão considere informações imprecisas e classifique os resultados de acordo com a relevância para o usuário.

A maior parte das abordagens citadas na Seção 3.4 considera taxonomias difusas (especialização/generalização difusa de conceitos) e relacionamentos de similaridade para expandir consultas com base em ontologias difusas. Alguns destes sistemas não especificam como é feita a expansão de consultas ambíguas, fato que pode comprometer a precisão de consultas, enquanto outros se dedicam exclusivamente a tratar ambigüidades, diminuindo a revocação pela omissão de outras expansões relevantes. Deste modo, é interessante que um sistema de expansão de consultas seja capaz de efetuar diversos tipos de expansão semântica com o intuito de obter tanto resultados relevantes quanto abrangentes. Além disso, outros tipos de relacionamentos difusos poderiam ser explorados pelo processo de expansão como, por exemplo, relacionamentos transitivos difusos, que permitem inferir conceitos que estejam associados entre si de forma indireta, e relacionamentos todo-parte, que podem indicar conceitos semanticamente próximos entre si em função das partes que os constituem.

Diante deste contexto, este capítulo apresenta o sistema FOQuE (*Fuzzy Ontology-based Query Expansion*), que utiliza ontologias difusas para expandir consultas não somente por especialização de conceitos e similaridade, mas também por proximidade todo-

parte e transitividade difusa, além de tratar consultas que contenham termos homônimos., As próximas seções (4.2 a 4.7) descrevem as características do sistema FOQuE, incluindo exemplos que ilustram como as expansões semânticas são realizadas.

## 4.2. ARQUITETURA DO SISTEMA FOQuE

O objetivo do sistema FOQuE é obter resultados aproximados que sejam semanticamente relacionados aos termos especificados pelo usuário, através da expansão de consultas com base na análise de conceitos e relacionamentos difusos em ontologias. FOQuE classifica as respostas de acordo com cada tipo de expansão, permitindo que resultados sejam exibidos em ordem decrescente de relevância. A Figura 4.1 apresenta a arquitetura geral do sistema, ilustrando seus principais módulos, a ontologia difusa e o mapeamento entre a ontologia e o banco de dados.

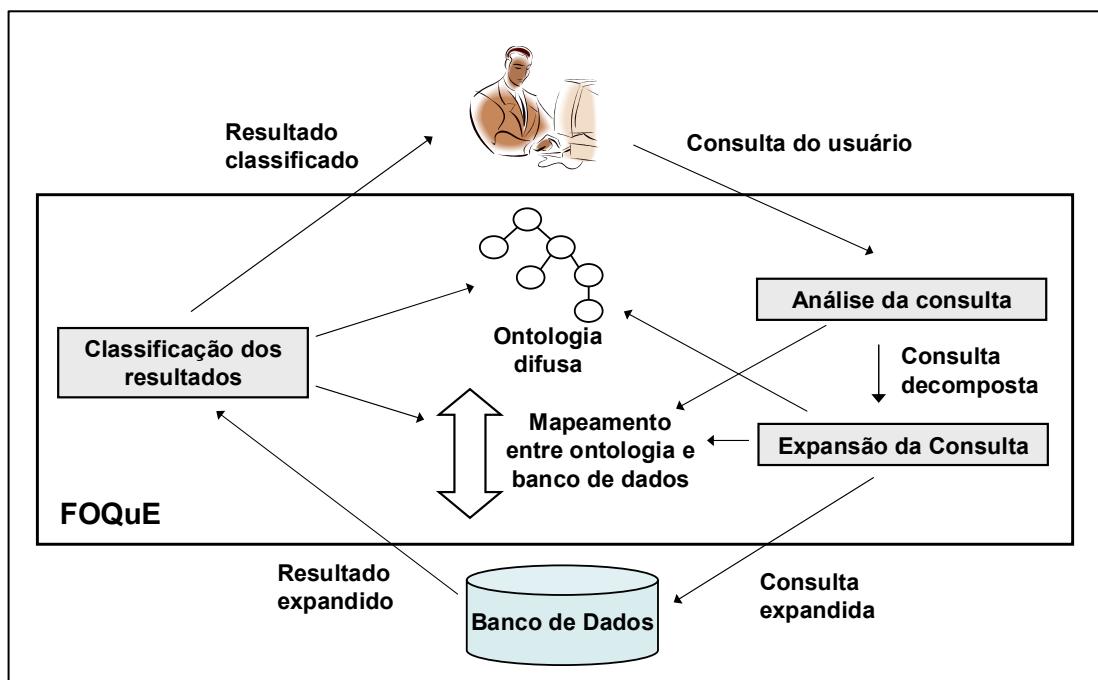


Figura 4.1 – Arquitetura do sistema FOQuE.

Conforme ilustrado na Figura 4.1, o sistema FOQuE se localiza entre a interface com o usuário e o banco de dados, sendo responsável por analisar a consulta (módulo *Análise da consulta*), expandi-la (módulo *Expansão da consulta*) e classificar os resultados retornados pelo banco de dados (módulo *Classificação dos resultados*). A consulta do usuário é especificada na linguagem *Structured Query Language* (SQL) que, por ser a linguagem padrão para consultas em bancos de dados relacionais, dispensa que usuários e

desenvolvedores de aplicações tenham que aprender uma nova linguagem para expandir as consultas. O usuário fornece ainda alguns parâmetros para controlar as expansões: similaridade mínima (*minSimilarity*), pertinência mínima (*minMembership*) e proximidade mínima (*minCloseness*).

O módulo *Análise da consulta* decompõe a consulta para que o módulo *Expansão da consulta* seja capaz de identificar quais expansões podem ser aplicadas com base na ontologia difusa e no mapeamento entre a ontologia e o banco de dados. A ontologia difusa contém conceitos e relacionamentos difusos que descrevem a semântica dos dados armazenados, permitindo que sejam realizadas inferências lógicas para o processo de expansão de consulta. O mapeamento entre a ontologia e o banco de dados contém informações essenciais para escrever a consulta expandida e analisar os resultados retornados, pois descreve como os conceitos e relacionamentos da ontologia são representados pelo banco de dados. A consulta expandida também é especificada em SQL e, portanto, pode ser executada por qualquer Sistema Gerenciador de Banco de Dados Relacional (SGBDR). As respostas retornadas são, por fim, classificadas pelo módulo *Classificação dos resultados* de acordo com as expansões realizadas, etapa que também se baseia na ontologia difusa e no mapeamento.

As próximas seções (4.3 a 4.7) descrevem com mais detalhes cada um dos elementos da arquitetura do sistema FOQuE.

### 4.3. ONTOLOGIA DIFUSA

A ontologia difusa utilizada pelo sistema FOQuE representa a semântica dos dados armazenados no banco, ou seja, especifica quais conceitos, relacionamentos e axiomas do domínio estão associados às tabelas, atributos e dados pertencentes ao banco. A ontologia difusa também representa a intensidade de relacionamentos e axiomas por meio de graus difusos, aumentando assim a expressividade do conhecimento do domínio. Deste modo, é possível inferir novas informações que, embora não estejam explícitas no banco de dados, são relevantes para a consulta do usuário.

As máquinas de inferência para ontologias difusas citadas nas Seções 2.4.2 e 2.4.3 apresentam algumas restrições com relação aos requisitos necessários para a expansão de consultas de FOQuE. Por exemplo, *FuzzyJ Toolkit* [ORCHARD, 2001] realiza inferências

somente sobre conjuntos difusos quantitativos, ou seja, conjuntos que representam atributos numéricos como idade, velocidade, temperatura, entre outros. *FiRE* [STOILLOS *et al.*, 2006] permite realizar inferências sobre conceitos e relacionamentos difusos, porém possui restrições para representar atributos de conceitos, além de não disponibilizar uma interface para aplicações (*Application Programming Interface* – API). *fuzzy DL* [STRACCIA, 2007] ainda não possui uma versão estável, pois ainda está em fase de desenvolvimento e testes. *FOWL* [GAO; LIU, 2005] e *Fuzzy OWL* [STOILLOS *et al.*, 2006] ainda não possuem máquinas de inferência disponíveis para apoiar o raciocínio sobre conceitos e relacionamentos difusos.

Diante destas limitações, decidiu-se desenvolver uma estratégia própria para representar ontologias difusas, com base na abordagem citada na Seção 2.4.2 (ontologias com classes e relacionamentos difusos). Esta abordagem é adequada para o processo de expansão de FOQuE, pois permite realizar inferências sobre classes difusas, relacionamentos difusos com propriedades simétrica e transitiva, além de considerar relacionamentos especiais como similaridade e proximidade. A linguagem escolhida para a codificação das ontologias foi OWL, por viabilizar o uso de mecanismos de inferência na expansão das consultas, além de ser uma linguagem recomendada pela W3C, amplamente utilizada como padrão para ontologias. Assim, criou-se uma ontologia de alto nível (meta-ontologia) que modela classes e relacionamentos difusos para serem herdados ou instanciados pelas ontologias específicas de domínio, de modo que estas sejam capazes de representar informações imprecisas. As Figuras 4.2 a 4.5 ilustram, respectivamente, como a meta-ontologia permite modelar classes e relacionamentos difusos, tanto de forma abstrata (Figuras 4.2 e 4.4) quanto em formato OWL/XML (Figuras 4.3 e 4.5).

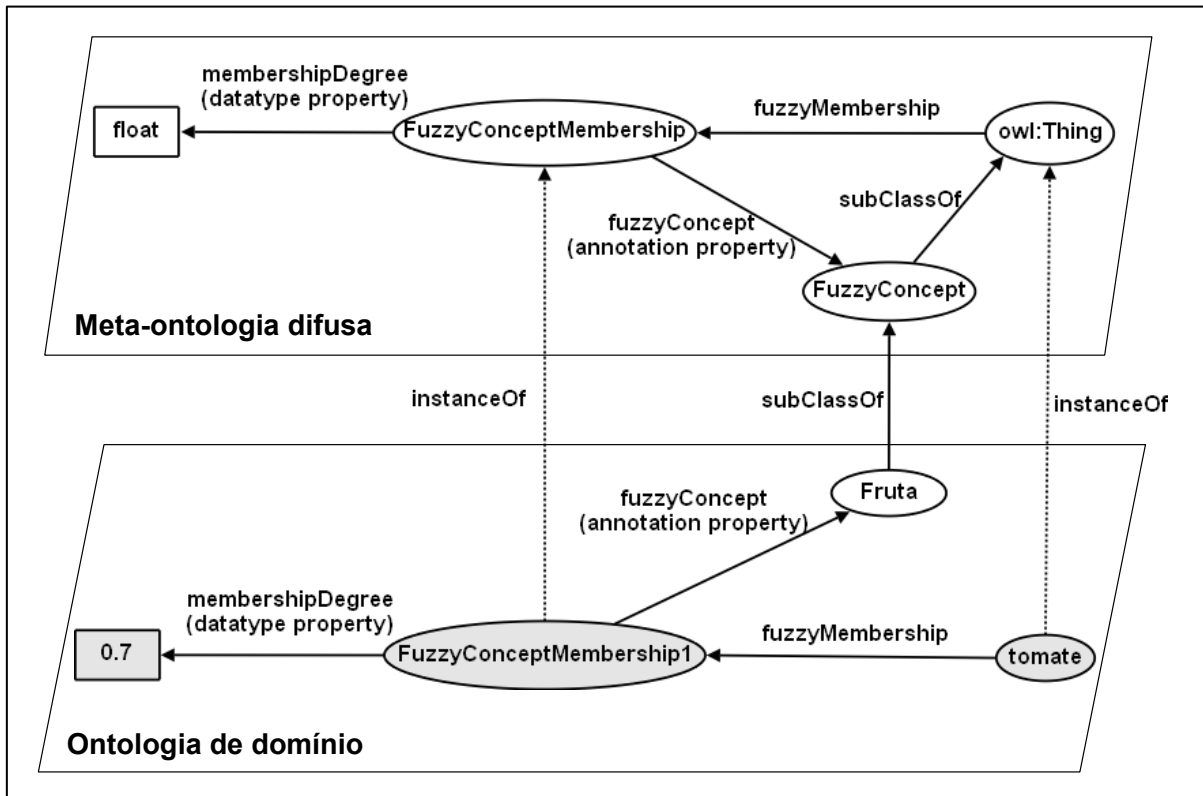


Figura 4.2 – Representação abstrata de classe difusa.

```

<Fruta rdf:ID="tomate">
  <rdfs:label>tomate</rdfs:label>
  <fuz:fuzzyMembership>
    <fuz:FuzzyConceptMembership>
      <fuz:fuzzyConcept rdf:resource="#Fruta"/>
      <fuz:membershipDegree rdf:datatype="xsd:float">0.7</fuz:membershipDegree>
    </fuz:FuzzyConceptMembership>
  </fuz:fuzzyMembership>
</Fruta>

```

Figura 4.3 – Representação em OWL/XML de uma instância de classe difusa.

Na Figura 4.2, a classe *FuzzyConceptMembership* contém informações para representar classes difusas: o relacionamento *fuzzyMembership*, que liga uma instância da ontologia a uma classe difusa, a classe difusa propriamente dita (*fuzzyConcept*) e o grau de pertinência correspondente (*membershipDegree*). O relacionamento *fuzzyConcept* é uma propriedade de anotação (*annotationProperty*) de OWL, pois ela permite relacionar uma instância da ontologia a uma classe, enquanto os outros tipos de propriedades de OWL somente representam associações entre instâncias. Na ontologia de domínio, as classes difusas são modeladas como subclasses de *FuzzyConcept* e relacionam-se às instâncias que possuem grau de pertinência  $< 1$  por *FuzzyConceptMembership*. No exemplo, a instância *tomate* possui grau de pertinência 0.7 para a classe difusa *Fruta*. A Figura 4.3 mostra a representação deste

exemplo em OWL/XML (elementos contendo o prefixo *fuz:* representam os elementos da meta-ontologia).

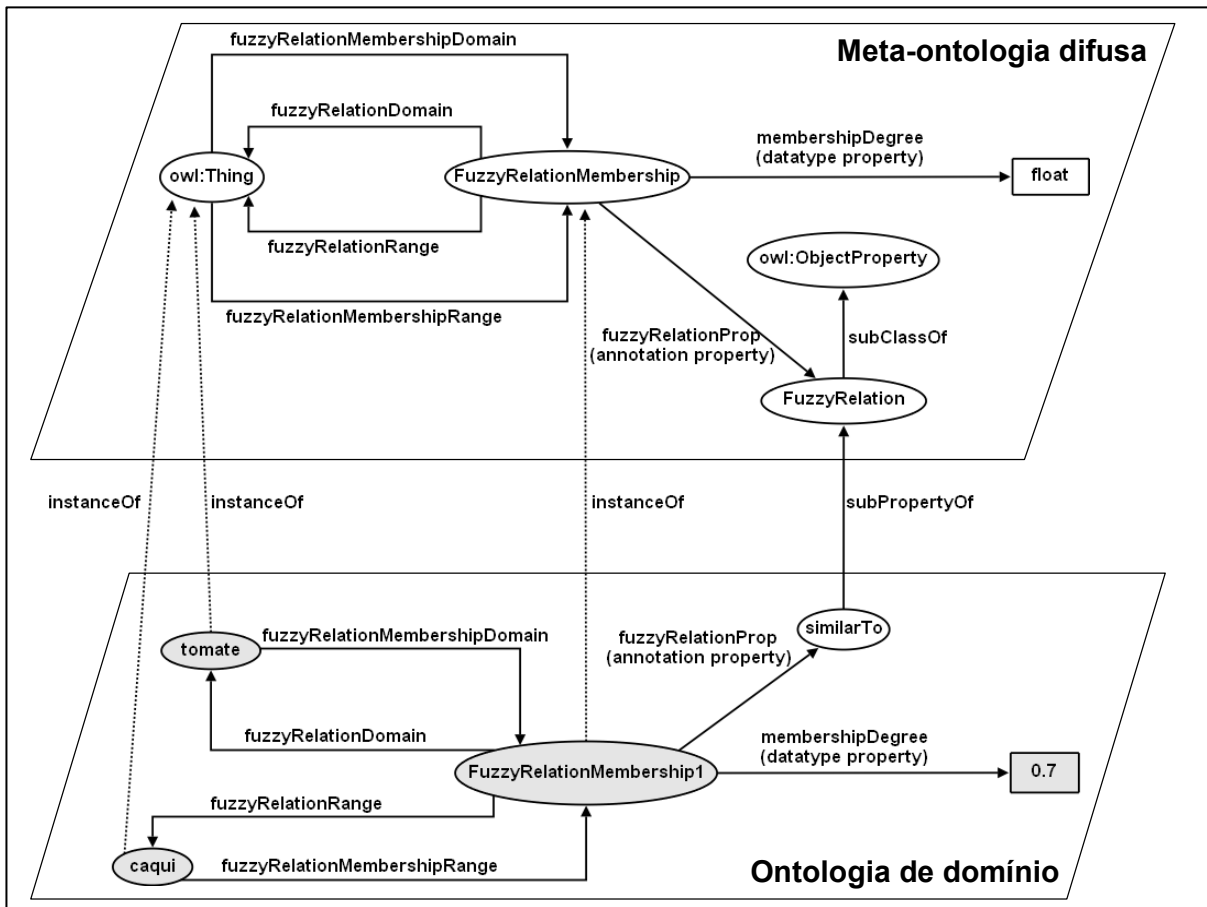


Figura 4.4 – Representação abstrata de relacionamento difuso binário.

```
<fuz:FuzzyRelationMembership>
  <fuz:fuzzyRelationProp rdf:resource="#fuz:similarTo" />
  <fuz:fuzzyRelationDomain rdf:resource="#tomate" />
  <fuz:fuzzyRelationRange rdf:resource="#caqui" />
  <fuz:membershipDegree rdf:datatype="#xsd:float">0.7</fuz:membershipDegree>
</fuz:FuzzyRelationMembership>
```

Figura 4.5 – Representação em OWL/XML de relacionamento difuso binário.

As Figuras 4.4 e 4.5 mostram como a meta-ontologia representa relacionamentos difusos binários através da classe *FuzzyRelationMembership*, responsável por atribuir um grau de pertinência (*membershipDegree*) ao relacionamento difuso (*fuzzyRelationProp*) entre duas instâncias (*fuzzyRelationDomain* e *fuzzyRelationRange*). Os relacionamentos *fuzzyRelationMembershipDomain* e *fuzzyRelationMembershipRange* são inversos de *fuzzyRelationDomain* e *fuzzyRelationRange*. O relacionamento *fuzzyRelationProp* também é uma propriedade de anotação assim como *fuzzyConcept*, pois associa instâncias de *FuzzyRelationMembership* a relacionamentos de OWL. Considerando a ontologia de domínio,

os relacionamentos difusos são modelados como sub-propriedades de *FuzzyRelation* e instâncias de *FuzzyRelationMembership* relacionam as instâncias da ontologia de domínio ao grau do relacionamento difuso correspondente. No exemplo, é apresentado o relacionamento difuso *similarTo* entre *tomate* e *caqui* com grau de similaridade = 0.7.

Esta estratégia para representar ontologias difusas não estende a linguagem OWL como *FOWL* e *Fuzzy OWL*, pois a sintaxe da linguagem não é modificada. Para representar classes e relacionamentos difusos, basta que as ontologias de domínio herdem ou instanciem os elementos da meta-ontologia difusa. Deste modo, a incorporação de conceitos de lógica difusa não impede o uso de máquinas de inferência existentes para OWL, sendo necessário adicionar alguns mecanismos de raciocínio sobre conceitos e relacionamentos difusos, para serem utilizados pelo processo de expansão de consultas de FOQuE:

- *Generalização/especialização de conceitos difusos*:  $A \subseteq B \Leftrightarrow \mu_A(x) \leq \mu_B(x)$ ;
- *Propriedade reflexiva de relacionamentos difusos*:  $\mu_R(x, x) = 1$ ;
- *Propriedade simétrica de relacionamentos difusos*:  $\mu_R(x, y) = \mu_R(y, x)$ ;
- *Composição max-min de relacionamentos transitivos difusos*:

$$\mu_R(x, z) = \max_{y \in X} \min [ \mu_R(x, y), \mu_R(y, z) ]$$

No caso da composição de relacionamentos transitivos difusos, é importante notar que ela respeita as definições de composição e transitividade *max-min*, apresentadas na Seção 2.3.2. A partir das definições listadas, o sistema FOQuE é capaz de realizar inferências para determinar, por exemplo, graus de pertinência e de similaridade entre conceitos da ontologia difusa.

Vale ressaltar que a ontologia difusa é fundamental para o sistema FOQuE, pois as expansões semânticas baseiam-se nos conceitos e relacionamentos difusos para realizar inferências e determinar quais as respostas aproximadas para as consultas. Portanto, a qualidade das ontologias difusas utilizadas é importante para garantir que o processo de expansão de consultas seja realizado corretamente. No entanto, o foco do sistema FOQuE está no uso e não na criação e avaliação de ontologias difusas, desta forma assume-se que as ontologias difusas utilizadas são adequadas para o processo de expansão semântica de consultas.



#### 4.4. MAPEAMENTO ENTRE ONTOLOGIA DIFUSA E BANCO DE DADOS

Uma vez definida a ontologia difusa que especifica a semântica dos dados armazenados, é necessário descrever associações entre seus elementos e o banco de dados para possibilitar a reformulação de consultas contendo os conceitos expandidos. O mapeamento é, portanto, responsável por fazer a correspondência de tabelas, atributos e dados com os respectivos conceitos e relacionamentos da ontologia. A partir destas correspondências, FOQuE decide como representar as expansões semânticas em termos dos elementos do banco de dados.

A especificação do mapeamento segue a abordagem de Enriquecimento de Estrutura [WACHE *et al.*, 2001], em que a ontologia se assemelha à estrutura do banco e contém definições que estendem a semântica dos dados. Deste modo, a ontologia adiciona aspectos semânticos aos dados, ao incluir definições de classes e relacionamentos difusos, generalização/especialização, relacionamentos transitivos, similaridade, entre outros. Como as estruturas do banco de dados e da ontologia são semelhantes, o mapeamento geralmente possui correspondências um-para-um, ou seja, a maioria das associações é direta. No caso das instâncias da ontologia, uma instância pode estar mapeada para um conjunto de dados correspondente ao conceito semanticamente representado por ela. Por exemplo, a instância *celular* da ontologia pode representar o conjunto de todos os celulares cadastrados no banco de dados, independentemente do tipo, marca, cor e outras características.

No sistema FOQuE, o mapeamento é composto por correspondências em forma de regras *antecedente*  $\rightarrow$  *conseqüente*, onde o antecedente contém elementos da ontologia e o conseqüente os elementos do banco de dados. Vale ressaltar que nem todos os elementos do banco de dados são associados aos elementos da ontologia, pois as correspondências são feitas somente entre elementos que contribuam para a expansão semântica das consultas. Por exemplo, atributos como código de produto ou quantidade de produtos não são considerados pelo mapeamento por não serem relevantes ao processo de expansão semântica.

Para exemplificar a definição de mapeamentos, a Figura 4.6 mostra o esquema de um banco de dados relacional simples sobre produtos de supermercados de grandes redes (hipermercados). O banco de dados contém informações usuais sobre produtos, como nome, marca, descrição, cor, preço, unidade de compra (quilo, unitário, etc) e tipo (categoria do

produto). Alguns produtos podem ser compostos por outros produtos, fato representado pela tabela de relacionamento *Composto*. A

Figura 4.7 apresenta uma visão geral da ontologia difusa correspondente, contendo as principais classes de produtos (generalização/especialização por *subClassOf*) e os relacionamentos difusos *similarTo* (relacionamento de similaridade), *hasPart* e *partOf* (relacionamentos todo-parte).

Tabela Produto							
id	nome	marca	descricao	cor	unidade	preco	tipo
0415	desktop	Compaq	-	-	unitário	2900	eletro-eletronico
0118	hard disk	Seagate	-	-	unitário	220	componente

Tabela Composto	
idtodo	idparte
0415	0118

Figura 4.6 – Esquema do banco de dados contendo produtos de supermercado.

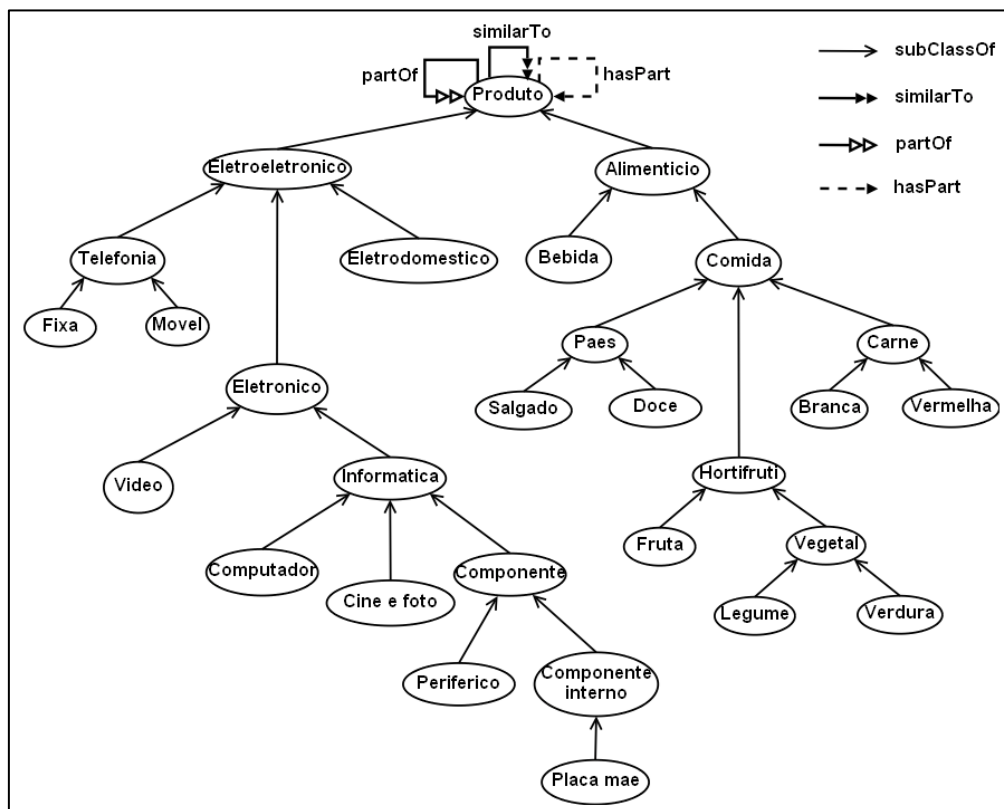


Figura 4.7 – Visão geral da ontologia sobre produtos de supermercado.

Considerando o esquema da Figura 4.6 e a ontologia da

Figura 4.7, são obtidas as seguintes correspondências:

- A tabela *Produto* corresponde à classe *Produto* da ontologia, ou seja, os dados contidos na tabela estão semanticamente relacionados às subclasses e instâncias da classe *Produto* da ontologia. Esta correspondência é representada pelo mapeamento entre o atributo *nome* da tabela *Produto* e a propriedade *rdfs:label* que rotula os nomes das subclasses e instâncias da classe *Produto*. Assim, são definidas as regras:

$$rdfs:subClassOf(x, Produto), rdfs:label(x,y) \rightarrow Produto(x), nome(x,y)$$

$$Produto(x), rdfs:label(x,y) \rightarrow Produto(x), nome(x,y)$$

- A tabela *Composto* corresponde aos relacionamentos todo-parte *fuz:hasPart* e *fuz:partOf* através de seus atributos que constituem chaves estrangeiras para a tabela *Produto*. Para associar um atributo de tabela com um relacionamento da ontologia, decidiu-se por convenção considerar o sujeito (*subject*) do relacionamento para construir o mapeamento. Deste modo, *fuz:hasPart* corresponde ao atributo *idtodo* da tabela *Composto* pela regra:

$$fuz:hasPart(x,y) \rightarrow Composto(z), idtodo(z,x)$$

e *fuz:partOf* corresponde ao atributo *idparte* pela regra:

$$fuz:partOf(x,y) \rightarrow Composto(z), idparte(z,x)$$

Desta forma, toda vez que o atributo *idtodo* for referenciado, FOQuE pode interpretar que a consulta envolve o sujeito do relacionamento *fuz:hasPart* da ontologia, ou seja, referencia o conceito que representa o “todo”;

- No caso de homônimos como o produto *ventilador*, que pode ser interpretado como eletrodoméstico ou componente de informática (*cooler*), o mapeamento trata a ambigüidade através de correspondências adequadas entre os dados do banco de dados e os respectivos conceitos da ontologia. Para tanto, utiliza-se algum atributo do banco de dados que remova a ambigüidade dos dados como, no exemplo, o atributo *tipo* da tabela *Produto*. Assim, o produto *ventilador* que representa o conceito de ventilador como eletrodoméstico é especificado pela regra de mapeamento:

$ventilador\_ambiente \rightarrow Produto(x), nome(x, 'ventilador'), tipo(x, 'eletrodomestico')$

e o produto *ventilador* que representa o conceito de ventilador como componente de informática:

$ventilador\_componente \rightarrow Produto(x), nome(x, 'ventilador'), tipo(x, 'componente')$

sendo que *ventilador\_ambiente* e *ventilador\_componente* são as instâncias da ontologia que representam os diferentes conceitos relacionados ao termo homônimo *ventilador*.

Deste modo, o mapeamento contém as correspondências entre os elementos da ontologia e os elementos do banco de dados, que são informações essenciais ao sistema FOQuE porque permitem que as expansões semânticas sejam reescritas como consultas expandidas ao banco de dados. Portanto, é importante conhecer bem a organização da ontologia difusa e do banco de dados para construir corretamente os mapeamentos, ou seja, esta é uma tarefa para especialistas de domínio, no caso de uma construção manual ou semi-automática.

#### 4.5. ANÁLISE DA CONSULTA

O módulo *Análise da consulta* é responsável por decompor a consulta original do usuário de modo que o sistema FOQuE seja capaz de identificar quais as expansões semânticas possíveis. A consulta é decomposta segundo os elementos da linguagem SQL: SELECT, que contém quais atributos o usuário deseja consultar; FROM, que contém as tabelas necessárias para a consulta; e WHERE, que especifica as condições que devem ser satisfeitas. Até o momento, o sistema FOQuE interpreta consultas simples contendo atributos em SELECT, tabelas em FROM e condições em WHERE do tipo *atributo = atributo* ou *atributo = literal* ligadas por operadores AND ou OR. Trabalhos futuros podem incluir também funções de agregação, consultas aninhadas e outros recursos.

Identificados os atributos, suas respectivas tabelas e as condições envolvidas na consulta, o módulo verifica por condições do tipo *atributo = literal*, pois elas contêm os conceitos especificados pelo usuário. Posteriormente, verifica-se se *atributo* está especificado no mapeamento: caso positivo, a consulta decomposta é enviada para o módulo *Expansão da consulta* para aplicar as expansões possíveis; caso contrário, a consulta não é expandida

porque *atributo* não contribui para a expansão semântica (caso de atributos como código do produto, quantidade de produtos, entre outros). Considerando o exemplo dos produtos de supermercado (Figuras 4.6 e 4.7), a consulta *SELECT Produto.preco FROM Produto WHERE Produto.nome = 'desktop'* pode ser expandida, uma vez que o atributo *nome* da tabela *Produto* está presente no mapeamento (Seção 4.4). Por outro lado, a consulta *SELECT Produto.preco FROM Produto WHERE Produto.id = '0415'* não é expandida, pois o atributo *id* não é especificado nas regras de mapeamento.

Desta forma, o módulo *Análise da consulta* decompõe os elementos da consulta e também acessa o mapeamento entre a ontologia e o banco de dados para indicar se a consulta do usuário deve prosseguir ou não para o processo de expansão.

## 4.6. EXPANSÃO DA CONSULTA

O principal módulo da arquitetura de FOQuE é o módulo *Expansão da consulta*, que modifica a consulta original para obter respostas aproximadas que sejam semanticamente relevantes aos requisitos do usuário. Neste módulo, a consulta decomposta é analisada para determinar quais as expansões possíveis de acordo com a ontologia difusa e o mapeamento. O processo de expansão de FOQuE estende algumas expansões propostas por Necib & Freytag [NECIB; FREYTAG, 2004] [NECIB; FREYTAG, 2005a] [NECIB; FREYTAG, 2005b], ao incorporar conceitos de lógica difusa para representar informações imprecisas. São realizados cinco tipos de expansões (*Homônimos*, *Classes*, *Similaridade*, *Proximidade todo-parte* e *Transitividade*) segundo três parâmetros fornecidos pelo usuário: pertinência mínima (*minMembership*), para determinar um valor mínimo de pertinência em relacionamentos difusos; similaridade mínima (*minSimilarity*), para relacionamentos de similaridade; e proximidade mínima (*minCloseness*), para relacionamentos de proximidade todo-parte. As seções 4.6.1 a 4.6.5 detalham cada tipo de expansão, juntamente com exemplos ilustrativos para facilitar a compreensão da abordagem desenvolvida.

### 4.6.1. Homônimos

O primeiro tipo de expansão realizado por FOQuE é a expansão de *Homônimos*, que identifica e apresenta ao usuário os diferentes contextos presentes em

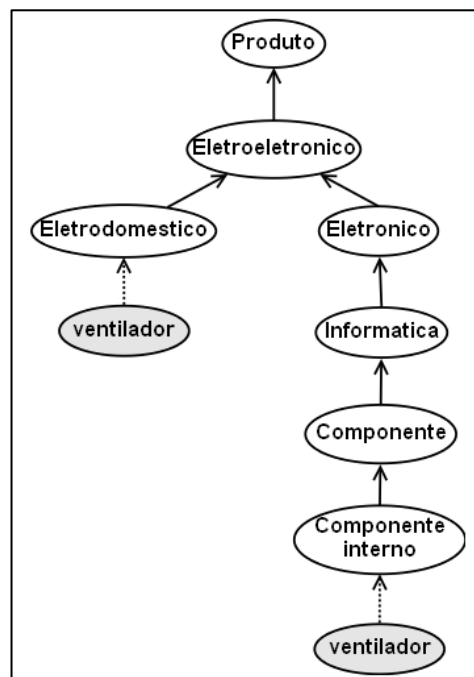
consultas ambíguas. O termo fornecido pelo usuário é analisado na ontologia e, caso seja representado por mais de um conceito, é considerado um termo homônimo. Para cada conceito identificado, são mostrados os respectivos caminhos na hierarquia de classes da ontologia, permitindo, assim, que o usuário escolha qual deles é o mais apropriado para seus requisitos. O mapeamento é então utilizado para reescrever a consulta correspondente ao contexto escolhido, evitando-se que resultados irrelevantes sejam recuperados.

Para ilustrar como é feita a expansão de homônimos, considere a seguinte consulta pelo produto *ventilador* (Figura 4.8):

```
SELECT Produto.marca, Produto.preco
FROM Produto
WHERE Produto.nome = 'ventilador'
```

**Figura 4.8 – Consulta contendo o termo homônimo *ventilador*.**

Como o atributo *Produto.nome* pode ser semanticamente expandido (está presente no mapeamento), o termo *ventilador* é verificado na ontologia difusa para obter quais os conceitos correspondentes na semântica de domínio. A Figura 4.9 mostra os dois conceitos da ontologia referentes ao termo *ventilador*: ventilador como eletrodoméstico (instância *ventilador\_ambiente*) e ventilador como componente de informática (instância *ventilador\_componente*).



**Figura 4.9 – Diferentes conceitos para o termo *ventilador*.**

Os conceitos identificados são exibidos ao usuário, que decide qual deles é o mais adequado para sua consulta. Suponha, por exemplo, que o usuário deseje consultar por ventilador como eletrodoméstico. A condição da consulta original deve ser modificada em função da regra de mapeamento correspondente ao conceito *ventilador\_ambiente*, ilustrada na Seção 4.4. A Figura 4.10 mostra a consulta resultante desta alteração, que recupera somente dados de ventiladores que sejam eletrodomésticos.

```
SELECT Produto.marca, Produto.preco
FROM Produto
WHERE ((Produto.nome = 'ventilador')
AND (Produto.tipo = 'eletrodomestico'))
```

**Figura 4.10 – Consulta modificada para recuperar o eletrodoméstico *ventilador*.**

Na Figura 4.10, adicionou-se a condição *Produto.tipo = 'eletrodomestico'*, de acordo com a regra de mapeamento para o conceito *ventilador\_ambiente*. O operador AND liga as duas condições porque é necessário que ambas sejam satisfeitas para remover a ambigüidade. Além disso, a condição reformulada contém parênteses e pertence ao mesmo local da condição original que foi expandida, para que a expansão realizada não altere a precedência das demais condições da consulta, caso existam.

Suponha agora que o usuário deseje consultar pelo produto ventilador como componente de informática. A condição da consulta original deve ser alterada em função da regra de mapeamento referente ao conceito *ventilador\_componente*, também apresentada na Seção 4.4. Assim, a Figura 4.11 mostra a consulta modificada para obter apenas os ventiladores que sejam componentes internos de computadores.

```
SELECT Produto.marca, Produto.preco
FROM Produto
WHERE ((Produto.nome = 'ventilador')
AND (Produto.tipo = 'componente'))
```

**Figura 4.11 – Consulta modificada para recuperar o componente *ventilador*.**

Depois da expansão de homônimos, FOQuE verifica as demais expansões possíveis de acordo com o contexto apropriado. Deste modo, apenas o conceito *ventilador\_ambiente* será considerado para as demais expansões da consulta da Figura 4.10, assim como o conceito *ventilador\_componente* para a consulta da Figura 4.11.

#### 4.6.2. Classes

A expansão de *Classes* é realizada quando a consulta do usuário contém termos que correspondem a classes na ontologia difusa de domínio. A consulta original é expandida ao adicionar as instâncias das classes e das respectivas subclasses, desde que possuam grau de pertinência maior ou igual ao parâmetro *minMembership*. Desta forma, o sistema FOQuE permite recuperar respostas aproximadas pela especialização de conceitos, ordenando os resultados em função da pertinência às classes especificadas pelo usuário.

Para exemplificar a expansão de classes, a Figura 4.12 mostra uma consulta em que o usuário especifica o termo *vegetal*, que corresponde a uma classe na ontologia difusa de domínio.

```
SELECT Produto.preco, Produto.unidade
FROM Produto
WHERE Produto.nome = 'vegetal'
```

Figura 4.12 – Consulta contendo o termo *vegetal* correspondente a uma classe na ontologia.

Supondo *minMembership* = 0.7, a ontologia difusa é analisada para obter as instâncias da classe *Vegetal* cujos graus de pertinência sejam maiores ou iguais a 0.7. A Figura 4.13 apresenta o trecho da ontologia que contém a classe *Vegetal*, suas subclasses, instâncias e respectivos graus de pertinência.

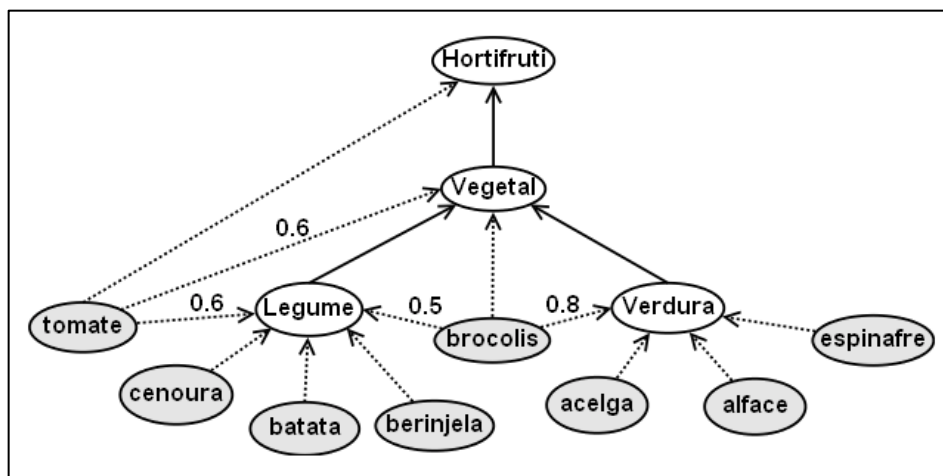


Figura 4.13 – Subclasses e instâncias da classe difusa *Vegetal*.

Ao analisar a ontologia difusa, são realizadas inferências para determinar o grau de pertinência de algumas instâncias para a classe *Vegetal*. Por exemplo, como *cenoura*



possui grau de pertinência = 1 (grau máximo) para a subclasse *Legume*, pode-se inferir que *cenoura* também possui grau de pertinência = 1 para *Vegetal*, de acordo com a definição de generalização/especialização de conceitos difusos ( $A \subseteq B \Leftrightarrow \mu_A(x) \leq \mu_B(x)$ ). Já no caso de *tomate*, pode-se inferir que  $\mu_{Vegetal}(tomate) \geq 0.6$ , sendo necessário definir explicitamente  $\mu_{Vegetal}(tomate) = 0.6$  na ontologia para obter seu valor exato. Depois da análise dos graus de pertinência, a consulta é expandida pela adição de condições OR para cada instância de *Vegetal* (e de suas subclasses) que tiver grau de pertinência acima de *minMembership*. A Figura 4.14 mostra a consulta modificada por FOQuE, que permite recuperar as instâncias da classe *Vegetal* com grau de pertinência  $\geq 0.7$ .

```
SELECT Produto.preco, Produto.unidade
FROM Produto
WHERE (((((((Produto.nome = 'vegetal')
OR Produto.nome = 'cenoura')
OR Produto.nome = 'batata')
OR Produto.nome = 'berinjela')
OR Produto.nome = 'brocolis')
OR Produto.nome = 'acelga')
OR Produto.nome = 'alface')
OR Produto.nome = 'espinafre')
```

**Figura 4.14 – Consulta expandida para recuperar instâncias da classe difusa *Vegetal*.**

Deste modo, a consulta original (Figura 4.12), que não retorna resultados porque não existe um produto específico chamado *vegetal*, é expandida para obter respostas semanticamente relacionadas pela especialização do conceito *Vegetal*. Além disso, é possível determinar um parâmetro mínimo para esta expansão de modo que sejam recuperados somente os conceitos relevantes para os requisitos do usuário. Assim, considerando a consulta da Figura 4.14, *tomate* não será recuperado, pois seu grau de pertinência para *Vegetal* ( $\mu_{Vegetal}(tomate) = 0.6$ ) é inferior que *minMembership*.

### 4.6.3. Similaridade

A expansão por *Similaridade* possibilita recuperar conceitos semanticamente relevantes considerando graus de similaridade entre conceitos. Para tanto, FOQuE modifica a consulta ao adicionar conceitos similares ao conceito especificado pelo usuário, desde que

satisfaçam ao parâmetro de similaridade mínima (*minSimilarity*). Os graus de similaridade também são usados pelo módulo *Classificação dos resultados* para ordenar as respostas de forma que resultados mais similares sejam exibidos em ordem decrescente.

Na ontologia difusa utilizada por FOQuE, os graus de similaridade são definidos manualmente por especialistas do domínio. Por enquanto, não são utilizados métodos automáticos porque o foco deste trabalho está voltado para o uso dos graus de similaridade no processo de expansão das consultas, independentemente da forma como foram definidos. Eventualmente, trabalhos futuros podem considerar formas automatizadas para determinar os graus de similaridade, com base em abordagens existentes como as citadas na Seção 3.4.

Embora a similaridade seja definida manualmente, não é necessário que o especialista especifique todos os graus de similaridade entre os conceitos da ontologia, pois FOQuE realiza inferências segundo a composição *max-min*. No entanto, as inferências por composição *max-min* de relações de similaridade no sistema FOQuE são restritas às instâncias de uma mesma classe da ontologia, ou seja, instâncias “irmãs” considerando a classe mais especializada para a qual elas pertencem (“pai” imediato). Duas razões justificam esta restrição. Em primeiro lugar, como os dados correspondem semanticamente às instâncias da ontologia, faz sentido considerar as relações de similaridade em nível de instância. Além disso, ao restringir a similaridade entre instâncias de uma mesma classe, garante-se que todos os conceitos similares entre si possuam de fato características compatíveis. Evita-se, portanto, que sejam feitas interpretações inadequadas sobre a semântica do domínio. Por exemplo, caso dois conceitos pertençam a classes diferentes e tenham similaridade elevada, pode ser que exista algum problema na representação do domínio, uma vez que tais conceitos deveriam ter a mesma classificação na ontologia por serem bastante similares entre si.

Para demonstrar como é feita a expansão por similaridade, a Figura 4.15 ilustra uma consulta em que o usuário especifica o produto *telefone sem fio*, correspondente à instância *telefone sem fio* que possui relacionamentos de similaridade (*similarTo*) para outras instâncias da ontologia difusa (Figura 4.16).

```

SELECT Produto.marca, Produto.preco
FROM Produto
WHERE Produto.nome = 'telefone sem fio'

```

Figura 4.15 - Consulta pelo produto *telefone sem fio*.

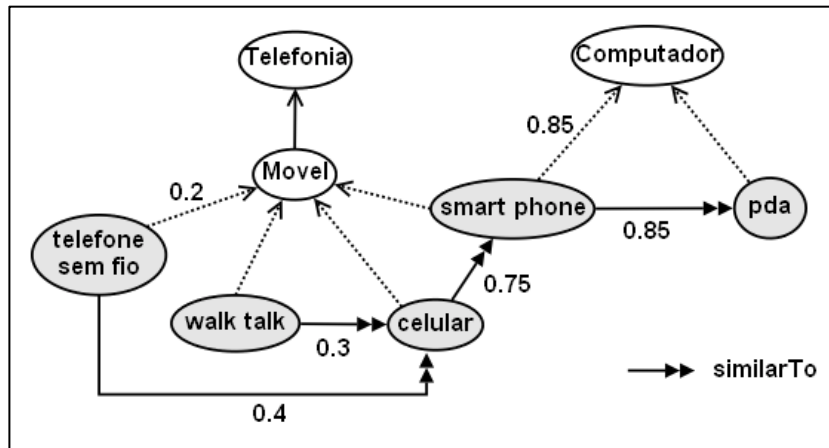


Figura 4.16 – Relacionamentos de similaridade entre *telefone sem fio* e outras instâncias da ontologia.

Conforme a Figura 4.16, *telefone sem fio* é diretamente similar a *celular*, porém outras instâncias similares podem ser obtidas através de inferências sobre a ontologia difusa. Por exemplo, *telefone sem fio* é similar a *walk talk* com grau de similaridade = 0.3, pois são verificadas a simetria  $\mu_{similarTo}(celular, walk\ talk) = \mu_{similarTo}(walk\ talk, celular)$  e a composição *max-min*  $\mu_{similarTo}(telefone\ sem\ fio, walk\ talk) = \min(0.4, 0.3)$ . Além disso, pode-se inferir que *telefone sem fio* é similar a *smart phone* com grau de similaridade = 0.4. No entanto, não é possível inferir a similaridade entre *telefone sem fio* e *pda*, pois ambos não possuem um “pai” imediato em comum (*Movel* é “pai” imediato de *telefone sem fio* enquanto *Computador* é “pai” imediato de *pda*). Assim, a Figura 4.17 mostra a consulta modificada por FOQuE ao adicionar condições OR para cada instância que possui grau de similaridade para *telefone sem fio* maior que *minSimilarity* = 0.4.

```

SELECT Produto.marca, Produto.preco
FROM Produto
WHERE (((Produto.nome = 'telefone sem fio')
OR (Produto.nome = 'celular'))
OR (Produto.nome = 'smart phone'))

```

Figura 4.17 – Consulta expandida por similaridade a *telefone sem fio*.

Na consulta da Figura 4.17, não seria adequado considerar *telefone sem fio* similar a *pda*, pois ambos não compartilham muitas características em comum, já que

pertencem a classes diferentes na ontologia. Por outro lado, *smart phone* pode ser considerado similar a *telefone sem fio* e a *pda*, pois a instância *smart phone* contém características de ambas as classes *Movel* e *Computador*.

#### 4.6.4. Proximidade todo-parte

Outro tipo de expansão de consultas realizado por FOQuE é a *Proximidade todo-parte*, que parte do princípio que conceitos constituídos por um conjunto aproximado de partes ou componentes em comum podem ser semanticamente próximos. O sistema FOQuE analisa relacionamentos todo-parte da ontologia difusa para obter conceitos próximos aos requisitos do usuário. Assim, da mesma forma que a expansão de classes e por similaridade, adicionam-se condições OR para cada conceito cujo grau de proximidade todo-parte seja maior ou igual ao parâmetro de proximidade mínima *minCloseness*.

Antes de descrever como são obtidos os graus de proximidade todo-parte, é necessário compreender como os relacionamentos todo-parte são representados na ontologia difusa utilizada por FOQuE. A meta-ontologia difusa (Seção 4.3) possui os relacionamentos *fuz:hasPart* (relação semântica de “todo” para “parte”) e *fuz:partOf* (relação semântica de “parte” para “todo”), que podem ser instanciados ou herdados pelas ontologias específicas de domínio. Estes relacionamentos são difusos, ou seja, é possível definir um grau de pertinência todo-parte no intervalo [0,1] representando a intensidade com a qual uma parte é relevante ou significativa para o todo. Por exemplo, o produto *impressora* pode ser parte do produto *desktop* com  $\mu_{partOf}(impressora, desktop) = 0.5$ , representando que a parte *impressora* possui uma relevância mediana para *desktop* porque nem todos tipos de *desktop* contêm *impressora* ou então porque esta parte não é fundamental para o funcionamento do “todo”. A Figura 4.18 mostra o trecho da ontologia difusa sobre produtos de supermercado que descreve as partes de *desktop* e os respectivos graus de pertinência todo-parte ( $\mu_{partOf} = 1$  quando o valor não é explícito na ontologia).

Além de considerar os graus de pertinência todo-parte definidos pelo usuário, FOQuE é capaz de inferir aqueles que não foram explicitados na ontologia. As inferências são realizadas com base na composição *max-min* dos relacionamentos todo-parte. Por exemplo, considerando ainda a Figura 4.18, pode-se inferir que *placa de fax modem* é parte de *desktop*

com  $\mu_{partOf}(placa\ de\ fax\ modem, desktop) = 0.5$ , pois  $\mu_{partOf}(placa\ de\ fax\ modem, placa\ mae\ onboard) = 0.5$  e  $\mu_{partOf}(placa\ mae\ onboard, desktop) = 1$ .

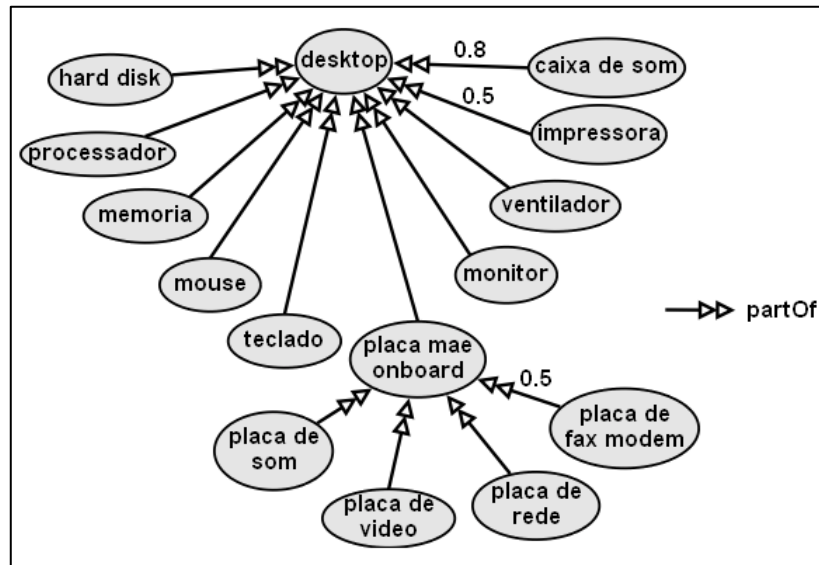


Figura 4.18 – Partes de *desktop* contendo graus de pertinência todo-parte.

A partir dos relacionamentos todo-parte difusos, é possível determinar a proximidade entre o conceito especificado pelo usuário e os demais conceitos da ontologia difusa. Em primeiro lugar, deve-se identificar se o termo fornecido na consulta pode ser semanticamente expandido, ou seja, se o atributo referente ao termo está presente no mapeamento. Em seguida, FOQuE verifica na ontologia se o termo especificado corresponde a um conceito constituído por partes, ao analisar se este possui propriedades *fuz:hasPart* (ou sub-propriedades). Caso positivo, calcula-se o grau de proximidade para os demais “todos” da ontologia, de acordo com a estratégia descrita pelo algoritmo apresentado na Figura 4.19.

```

1  c1 = instância correspondente ao termo especificado pelo usuário
2  prop1 = propriedade (ou sub-propriedade de) fuz:hasPart de c1
3  partes_c1 = conjunto de partes de c1 com graus de pertinência >=
4  minMembership considerando prop1
5  total_c1 = soma dos graus de pertinência das partes de partes_c1
6  para cada instância i1 que possui propriedades prop1
7    total_i1 = 0
8    comum_i1 = 0
9    comum_c1 = 0
10   para cada parte p1 de i1 considerando prop1
11     se grau de pertinência de p1 para i1 >= minMembership
12       total_i1 = total_i1 + grau de pertinência de p1 para i1
13       se p1 pertence a partes_c1
14         comum_i1 = comum_i1 + grau de pertinência de p1 para i1
15         comum_c1 = comum_c1 + grau de pertinência de p1 para c1
16       senão se p1 é similar a alguma parte p2 de partes_c1 com
17       grau de similaridade >= minSimilarity
         comum_i1 = comum_i1 + grau de pertinência de p1 para i1

```

18	$comum\_c1 = comum\_c1 + \text{grau de pertinência de } p2 \text{ para } c1$
19	fim se
20	fim se
21	fim para
22	$grau\_proximidade = (comum\_i1 + comum\_c1) / (total\_i1 + total\_c1)$
23	fim para

**Figura 4.19 – Algoritmo para o cálculo de grau de proximidade todo-parte.**

O algoritmo da Figura 4.19 analisa as partes cujos graus de pertinência todo-parte são maiores ou iguais a *minMembership*. No caso de partes transitivas, os graus de pertinência são inferidos através da composição *max-min* de relacionamentos todo-parte. Assim, para cada instância da ontologia que representa um “todo” (*i1*) considerando o mesmo tipo de relacionamento todo-parte (*prop1*) do conceito fornecido pelo usuário (*c1*), obtém-se a soma total dos graus de pertinência de suas partes (*total\_i1*) e a soma total dos graus de pertinência das partes em comum com *c1* (*comum\_i1*). Também se calcula a soma total dos graus de pertinência das partes de *c1* em comum com *i1* (*comum\_c1*), uma vez que as partes em comum podem ter graus de pertinência diferentes para *i1* e *c1*. Além disso, consideram-se graus de similaridade entre as partes de *i1* e *c1* para determinar *comum\_i1* e *comum\_c1*.

Uma vez definidos *total\_i1*, *total\_c1*, *comum\_i1* e *comum\_c1*, é necessário combinar esses valores para obter os graus de proximidade todo-parte. Decidiu-se por usar a média ponderada da divisão das partes em comum pelo total de partes, tanto para *i1* quanto para *c1*, sendo que a quantidade total de partes (*total\_i1* e *total\_c1*) é usada como peso:

$$\text{prox}(i1, c1) = \frac{\left( \frac{comum\_i1}{total\_i1} \times total\_i1 \right) + \left( \frac{comum\_c1}{total\_c1} \times total\_c1 \right)}{total\_i1 + total\_c1}$$

que é simplificada para:

$$\text{prox}(i1, c1) = \frac{comum\_i1 + comum\_c1}{total\_i1 + total\_c1}$$

Esta abordagem para calcular a proximidade considera a divisão da quantidade de partes em comum pelo total de partes porque ela indica se as partes em comum são significativas com relação ao total de partes de cada “todo”. Porém, esta divisão pode resultar em valores próximos a 1 mesmo que a quantidade total de partes de *i1* e *c1* seja bastante diferente, quando a quantidade de partes em comum é próxima do total de partes de somente um dos “todos”. Tal fato pode distorcer o resultado dos cálculos, levando a graus de proximidade inadequados. Portanto, o cálculo deve considerar também a quantidade total de

partes de cada “todo”, razão pela qual escolhida a média ponderada com o total de partes como peso. No entanto, esta não é a única abordagem possível para calcular a proximidade todo-parte, outras fórmulas podem ser consideradas dependendo do domínio de aplicação, como, por exemplo, a média aritmética, o mínimo das divisões, entre outras. Portanto, podem ser definidas outras formas de calcular a proximidade todo-parte com base em *total\_i1*, *total\_c1*, *comum\_i1* e *comum\_c1*, com o intuito de obter graus de proximidade apropriados para a semântica do domínio considerado.

Para exemplificar a expansão por proximidade todo-parte, considere a consulta por preços do produto *desktop* (Figura 4.20), cujas partes foram apresentadas na Figura 4.18.

```
SELECT Produto.preco
FROM Produto
WHERE Produto.nome = 'desktop'
```

**Figura 4.20 – Consulta pelo produto *desktop*, que pode ser expandido por proximidade todo-parte.**

Como *desktop* é constituído por partes na ontologia difusa, pode ser expandido por proximidade todo-parte. Assim, o sistema FOQuE efetua o cálculo do grau de proximidade de *desktop* para os demais “todos” da ontologia, com base no algoritmo da Figura 4.19 e nos parâmetros *minMembership* e *minSimilarity*. A Figura 4.21 mostra os outros “todos” da ontologia sobre produtos de supermercado, sendo que algumas partes estão repetidas na figura para visualizar com mais clareza os relacionamentos difusos *fuz:partOf*. Em seguida, a Tabela 4.1 apresenta os graus de proximidade calculados para os “todos” desta ontologia, considerando *minMembership* = 0.5 e *minSimilarity* = 0.7.

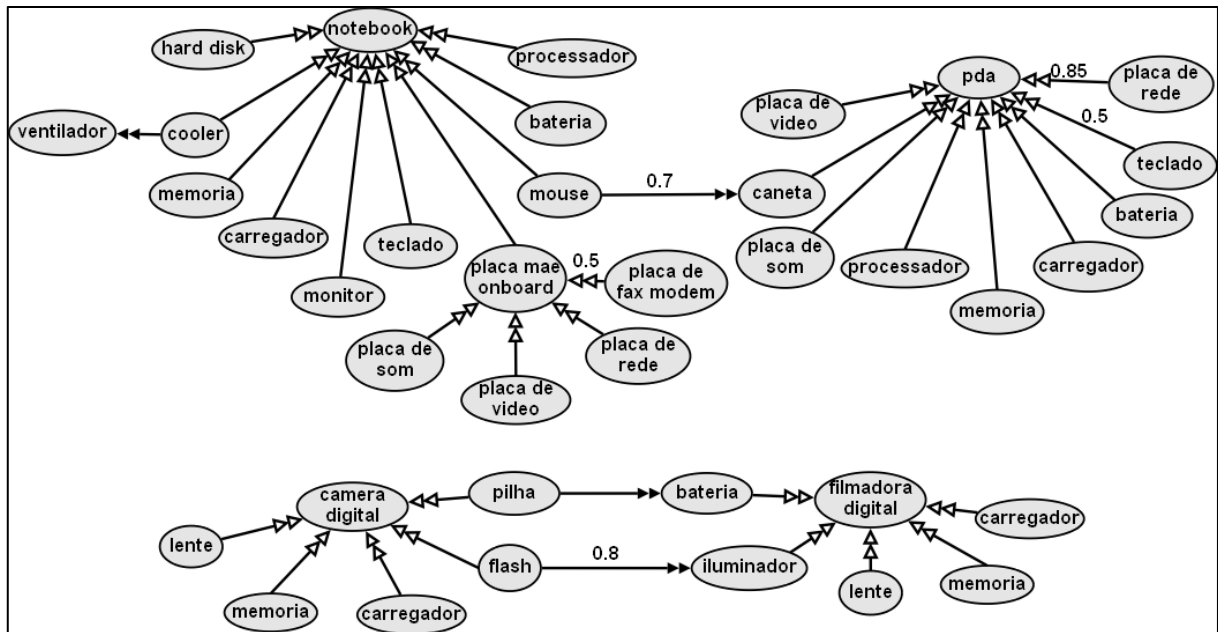


Figura 4.21 – “Todos” da ontologia difusa e respectivas partes.

Tabela 4.1 – Graus de proximidade todo-parte entre “todos” da ontologia difusa.

<i>i1</i>	<i>c1</i>	<i>comum i1</i>	<i>comum c1</i>	<i>total i1</i>	<i>total c1</i>	<i>prox(i1, c1)</i>
<i>notebook</i>	<i>desktop</i>	11.5	11.5	13.5	12.8	0.8745247
<i>pda</i>	<i>desktop</i>	6.35	7	8.35	12.8	0.6312056
<i>camera digital</i>	<i>desktop</i>	1	1	5	12.8	0.1123596
<i>filmadora digital</i>	<i>desktop</i>	1	1	5	12.8	0.1123596
<i>pda</i>	<i>notebook</i>	8.35	9	8.35	13.5	0.7940503
<i>camera digital</i>	<i>notebook</i>	3	3	5	13.5	0.3243243
<i>filmadora digital</i>	<i>notebook</i>	3	3	5	13.5	0.3243243
<i>camera digital</i>	<i>pda</i>	3	3	5	8.35	0.4494382
<i>filmadora digital</i>	<i>pda</i>	3	3	5	8.35	0.4494382
<i>camera digital</i>	<i>filmadora digital</i>	5	5	5	5	1

A Tabela 4.1 mostra um exemplo que justifica porque esta expansão considera proximidade entre “todos” e não similaridade. Conforme visto na Seção 2.3.2, a relação de similaridade possui as propriedades reflexiva, simétrica e transitiva. Entretanto, segundo a Tabela 4.1,  $prox(pda, desktop) < \min (prox(pda, notebook), prox(notebook, desktop))$ .



Portanto, a transitividade *max-min* não é necessariamente respeitada, levando a concluir que a relação obtida é de proximidade, que possui as propriedades reflexiva e simétrica.

Os valores de proximidade todo-parte podem ser diferentes conforme os parâmetros *minMembership* e *minSimilarity* fornecidos pelo usuário. Por exemplo, caso *minSimilarity* = 0.8, a parte *caneta* de *pda* não seria considerada similar a *mouse* de *desktop*, resultando em  $prox(pda, desktop) = 0.536643026$ , o que representa uma diferença de aproximadamente 0.1 em relação ao resultado obtido quando estas partes são consideradas similares (Tabela 4.1). Deste modo, o sistema FOQuE calcula a proximidade para cada consulta do usuário, mesmo que seja repetida, pois os resultados dependem dos parâmetros especificados.

Uma vez calculados os graus de proximidade, FOQuE expande a consulta do usuário ao adicionar condições OR para os “todos” que satisfazem ao parâmetro de proximidade mínima *minCloseness*. Assim, a Figura 4.22 mostra a consulta resultante da expansão de *desktop* por proximidade todo-parte, considerando *minCloseness* = 0.6.

```
SELECT Produto.preco
FROM Produto
WHERE (((Produto.nome = 'desktop')
OR (Produto.nome = 'notebook'))
OR (Produto.nome = 'pda'))
```

**Figura 4.22 – Consulta expandida por proximidade todo-parte.**

Depois de expandir por proximidade todo-parte, FOQuE aplica a expansão por similaridade para cada conceito expandido. Neste caso, os conceitos similares ao conceito próximo também são considerados próximos com o mesmo grau que o conceito expandido por proximidade. Por exemplo, como o conceito *laptop* é similar a *notebook* com grau de similaridade maior que *minSimilarity*, *laptop* também é considerado próximo a *desktop* com  $prox(laptop, desktop) = 0.874524715$ .

#### 4.6.5. Transitividade

O último tipo de expansão realizado pelo sistema FOQuE é a expansão por *Transitividade*, que analisa relacionamentos transitivos da ontologia difusa que não são explícitos no banco de dados. Nesta expansão, o conceito especificado na consulta é

verificado na ontologia para identificar quantos níveis de transitividade ele possui e se os conceitos relacionados a ele por transitividade satisfazem ao parâmetro *minMembership*. A consulta original é modificada pela adição de novas condições que possibilitam recuperar dados por transitividade que sejam relevantes aos requisitos do usuário.

Para mostrar quais as alterações realizadas pela expansão por transitividade, considere a consulta por componentes do produto *notebook* (Figura 4.23) e o fragmento das tabelas *Produto* e *Composto* apresentado na Figura 4.24.

```
SELECT Comp.nome, Comp.preco
FROM Produto Comp, Produto Todo, Composto C
WHERE Comp.id = C.idparte
AND Todo.id = C.idtodo
AND Todo.nome = 'notebook'
```

**Figura 4.23 – Consulta por componentes do produto *notebook*.**

<b>Tabela Produto</b>							
<u>id</u>	nome	marca	descricao	cor	unidade	preco	tipo
0122	notebook	HP	-	-	unitário	3564,00	eletro-eletronico
0182	placa mae onboard	Gigabyte	LGA800	-	unitário	299,00	componente
0078	placa de video	Nvidia	MX4000 64mb	-	unitário	110,00	componente

<b>Tabela Composto</b>	
<u>idtodo</u>	<u>idparte</u>
0122	0182
0182	0078

**Figura 4.24 – Fragmento das tabelas *Produto* e *Composto*.**

Ao executar a consulta da Figura 4.23 sobre as tabelas da Figura 4.24, somente o componente *placa mae onboard* é recuperado, em função das junções de chave estrangeira da tabela *Composto*. Porém, como o relacionamento *produto é composto por produto* (*fuz:hasPart*) é transitivo na semântica do domínio, o produto *placa de video* também deveria ser recuperado, pois é um componente de *placa mae onboard*. Para recuperar os dados envolvidos em relacionamentos transitivos, ou o banco de dados explicita todos esses relacionamentos (inserir em *Composto* a tupla *idtodo* = 0122 e *idparte* = 0078) ou então o usuário deve conhecer previamente quantos níveis de transitividade são possíveis para o

produto requisitado, a fim de formular uma consulta que alcance todos esses níveis. Como estas opções podem ser inviáveis em algumas situações, o objetivo da expansão por transitividade realizada pelo sistema FOQuE é reformular a consulta de forma transparente, sem que seja necessário alterar os dados do banco nem exigir que o usuário especifique explicitamente a transitividade na consulta.

O primeiro passo da expansão por transitividade de FOQuE é identificar se a consulta do usuário contém elementos que correspondem a relacionamentos transitivos da ontologia. Para tanto, o módulo *Análise da consulta* verifica as condições de WHERE do tipo  $atributo1 = atributo2$ , que indicam possíveis junções de chave estrangeira relativas a tabelas de relacionamento como, no exemplo, a tabela *Composto*. Em seguida, deve-se identificar se  $atributo1$  ou  $atributo2$  estão associados a algum relacionamento transitivo da ontologia difusa através do mapeamento. No exemplo da Figura 4.23, o atributo  $C.idparte$  está mapeado para o relacionamento transitivo  $fuz:partOf$  da ontologia, segundo as regras de mapeamento apresentadas na Seção 4.4 ( $fuz:partOf(x,y) \rightarrow Composto(z), idparte(z,x)$ ). Depois de identificar um relacionamento transitivo (no caso  $fuz:partOf$ ), é necessário verificar se o seu relacionamento inverso também está presente na consulta, para determinar qual deles deverá ser expandido. Assim, voltando ao exemplo, o atributo  $C.idtodo$  da condição  $Todo.id = C.idtodo$  está mapeado para  $fuz:hasPart$ , que é inverso de  $fuz:partOf$  na ontologia. Por fim, resta identificar qual dos relacionamentos será expandido, dependendo do termo fornecido pelo usuário. No exemplo, como o termo *notebook* pode ser semanticamente expandido (atributo *nome* está no mapeamento) e foi especificado para a tabela *Todo* que referencia  $idtodo$  pela condição  $Todo.id = C.idtodo$ , a expansão deve ser realizada considerando o relacionamento  $fuz:hasPart$ . Caso o termo do usuário fosse especificado para a tabela *Comp*, a expansão seria feita para  $fuz:partOf$ , pois o atributo  $idparte$  faz parte da junção que referencia a tabela *Comp*. Ainda, caso fossem especificados termos associados a ambos relacionamentos transitivos (direto e inverso), a expansão seria feita para um dos dois sentidos. Desta forma, o sistema FOQuE é capaz de determinar como expandir a consulta por transitividade dependendo das condições de WHERE e do(s) termo(s) especificado(s) pelo usuário.

Como os conceitos resultantes de outros tipos de expansão (homônimo, classe, similaridade e proximidade) também podem ter relacionamentos transitivos na ontologia, a expansão deve considerar não somente o conceito especificado pelo usuário, mas também todos os conceitos expandidos. Por esta razão, a expansão por transitividade é realizada depois das outras expansões, de modo que resultados expandidos também sejam considerados.

Uma vez definidos o relacionamento transitivo e os conceitos a serem expandidos, a próxima etapa é reformular a consulta considerando os níveis de transitividade que devem ser alcançados. Para tanto, FOQuE analisa a ontologia difusa para obter os conceitos relacionados a cada conceito a ser expandido por transitividade, considerando o relacionamento transitivo identificado (no exemplo *fuz:hasPart*). Porém, os conceitos relacionados que possuem grau de pertinência menor que *minMembership* não devem ser recuperados pela consulta. Assim, são adicionadas condições *atributo*  $\diamond$  *conceito\_n*, para cada *conceito\_n* com grau de pertinência abaixo de *minMembership*, associadas à mesma condição que contém o conceito analisado. Por exemplo, considerando *minMembership* = 0.6 para a consulta por componentes de *notebook*, foram adicionadas à consulta expandida (Figura 4.25) as condições *Comp.nome*  $\diamond$  *placa de fax modem* para o conceito *notebook*, *Comp.nome*  $\diamond$  *placa de fax modem* e *Comp.nome*  $\diamond$  *impressora* para o conceito *desktop* expandido por proximidade, pois  $\mu_{partOf}(placa\ de\ fax\ modem, notebook) = 0.5$ ,  $\mu_{partOf}(placa\ de\ fax\ modem, desktop) = 0.5$  e  $\mu_{partOf}(impressora, desktop) = 0.5$ , segundo os trechos da ontologia difusa apresentados nas Figuras 4.18 e 4.21 e as inferências com base na composição *max-min*.

```
SELECT Comp.nome, Comp.preco
FROM Produto Comp, Produto Todo, Composto C
WHERE (((Comp.id = C.idparte)
AND (Todo.id = C.idtodo))
AND (((Todo.nome = 'notebook') AND (Comp.nome <> 'placa de fax modem'))
OR (((Todo.nome = 'desktop') AND (Comp.nome <> 'placa de fax modem'))
AND (Comp.nome <> 'impressora'))))
```

**Figura 4.25 – Consulta expandida considerando graus de pertinência em relacionamentos transitivos.**

Embora a consulta da Figura 4.25 considere graus de pertinência em relacionamentos transitivos, ainda não é possível obter os conceitos por transitividade porque a consulta alcança somente os conceitos que estejam diretamente relacionados por *fuz:hasPart* a *notebook* e *desktop*. Para resolver este problema, é necessário determinar qual o maior nível de transitividade considerando cada conceito presente na consulta expandida. No exemplo, como *notebook* e *desktop* são os conceitos a serem expandidos por transitividade, verifica-se na ontologia qual o caminho de transitividade mais longo para eles considerando o relacionamento *fuz:hasPart*. A Figura 4.26 mostra os caminhos transitivos mais longos de *fuz:partOf*, inverso de *fuz:hasPart*, referentes a *desktop* e *notebook*.

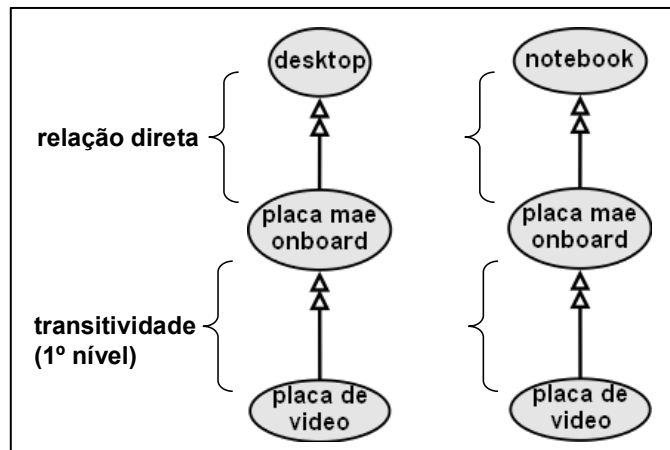


Figura 4.26 – Níveis de transitividade para *desktop* e *notebook* com relação a *fuz:partOf*.

Segundo a Figura 4.26, o 1º nível de transitividade é o maior nível possível com relação a *fuz:partOf* (inverso de *fuz:hasPart*), pois a relação direta já é contemplada pela consulta da Figura 4.25. É necessário, portanto, alterar a consulta de modo que ela recupere os dados envolvidos nesse nível de transitividade como, no exemplo, o componente *placa de video*. Assim, o algoritmo da Figura 4.27 mostra a estratégia adotada pelo sistema FOQuE para modificar a consulta considerando os níveis de transitividade.

```

1  rel = relacionamento transitivo a ser expandido
2  n = maior nível de transitividade possível considerando rel e os
   conceitos da consulta
3  tr = tabela de relacionamento que contém chaves estrangeiras
4  ch1 = chave estrangeira de tr referente a rel
5  ch2 = chave estrangeira de tr referente ao relacionamento inverso de
   rel
6  c = consulta previamente expandida pelos outros tipos de expansão
7  atrb = atributo de uma tabela associado a uma chave estrangeira de
   outra tabela
8
9  c_final = c
10 t = tr
11 para i = 1 até n (inclusive)
12   ci = c
13   adicione a ci uma nova tabela tr com nome tr_i, onde tr é o nome
   da tabela tr e i é o contador do loop
14   substitua em ci a condição t.ch2 = atrb por t.ch2 = tr_i.ch1
15   adicione a ci a condição tr_i.ch2 = atrb
16   c_final = c_final UNION ci
17   c = ci
18   t = tr_i
19 fim para
20
21 retorne c_final

```

Figura 4.27 – Algoritmo para expandir consultas por transitividade.

Conforme o algoritmo apresentado na Figura 4.27, a consulta é modificada pela adição de tabelas de relacionamento e alteração das condições envolvendo as chaves

estrangeiras, para que os diversos níveis de transitividade sejam devidamente alcançados. No entanto, como cada consulta alterada atinge somente os conceitos de um determinado nível, é necessário uni-las para obter todos os conceitos pertencentes aos diferentes níveis de transitividade. No final do processo, haverá uma consulta correspondente a cada nível de transitividade, além da própria consulta inicial (*c*) resultante das outras expansões (homônimo, classe, similaridade e proximidade).

Considerando a consulta por componentes de *notebook* e o algoritmo apresentado, *c* corresponde à consulta previamente expandida pelos outros tipos de expansão (Figura 4.25), *rel* ao relacionamento *fuz:hasPart*, *tr* à tabela *C* (*Composto*), *ch1* ao atributo *idtodo* e *ch2* ao atributo *idparte*. Além disso, *n* = 1, pois existe no máximo um nível de transitividade de acordo com a Figura 4.26. Segundo a estratégia adotada, a consulta *ci* é modificada pela adição da tabela *C\_1* e das condições *C.idparte* = *C\_1.idtodo* e *C\_1.idparte* = *Comp.id*, para que seja possível recuperar os conceitos referentes ao 1º nível de transitividade. Por fim, as duas consultas *c* e *ci* devem ser unidas para recuperar todos os conceitos relacionados por transitividade a *notebook* e *desktop*. A consulta resultante é apresentada na Figura 4.28.

```

SELECT Comp.nome, Comp.preco
FROM Produto Comp, Produto Todo, Composto C, Composto C_1
WHERE (((Comp.id = C_1.idparte) AND (C_1.idtodo = C.idparte))
AND (Todo.id = C.idtodo))
AND (((Todo.nome = 'notebook') AND (Comp.nome <> 'placa de fax modem'))
OR (((Todo.nome = 'desktop') AND (Comp.nome <> 'placa de fax modem'))
AND (Comp.nome <> 'impressora'))))
UNION
( SELECT Comp.nome, Comp.preco
FROM Produto Comp, Produto Todo, Composto C
WHERE (((Comp.id = C.idparte)
AND (Todo.id = C.idtodo))
AND (((Todo.nome = 'notebook') AND (Comp.nome <> 'placa de fax modem'))
OR (((Todo.nome = 'desktop') AND (Comp.nome <> 'placa de fax modem'))
AND (Comp.nome <> 'impressora'))))
)

```

**Figura 4.28 – Consulta expandida considerando o relacionamento transitivo *fuz:hasPart*.**

No caso de consultas que podem ser expandidas por dois relacionamentos transitivos inversos entre si como, por exemplo, *fuz:hasPart* (*Todo.nome* = '*notebook*') e

*fuz:partOf* (*Comp.nome* = 'placa de video'), deve-se analisar qual o caminho mais longo de transitividade para cada relacionamento para então obter o maior dentre eles. Desta forma, expande-se o relacionamento transitivo que tiver o caminho de transitividade mais longo na ontologia difusa, pois assim garante-se que todos os níveis possíveis serão alcançados. Caso ambos tenham o mesmo tamanho, pode-se expandir qualquer um dos relacionamentos.

Vale ressaltar que a abordagem de FOQuE para expansão por transitividade leva em consideração relacionamentos transitivos da ontologia que não formam ciclos. Além disso, somente são analisados os relacionamentos que correspondem a auto-relacionamentos no banco de dados. Eventualmente, trabalhos futuros podem estender a análise de transitividade para outros tipos de relacionamento mais complexos.

#### 4.7. CLASSIFICAÇÃO DOS RESULTADOS

Depois que a consulta original foi expandida por FOQuE com base nos diversos tipos de expansão citados anteriormente, ela é executada sobre o banco de dados para retornar os resultados expandidos. É neste ponto que atua o módulo *Classificação dos resultados*, responsável por classificar as respostas obtidas de acordo com o tipo de expansão que as originaram. Assim, FOQuE informa ao usuário quais expansões foram realizadas e quais os resultados correspondentes em ordem decrescente de relevância com relação ao parâmetro considerado (grau de pertinência, similaridade ou proximidade).

A cada expansão realizada para a consulta do usuário, FOQuE mantém quais os conceitos expandidos e os respectivos graus analisados. Por exemplo, no caso de uma expansão por similaridade, os conceitos similares são armazenados em uma estrutura de dados juntamente com seus respectivos graus de similaridade. Depois de executar a consulta expandida, os dados retornados são analisados e, caso correspondam a esses conceitos, é feita a classificação de acordo com o grau de similaridade. A ontologia difusa e o mapeamento são importantes neste processo, pois são usados para identificar se os dados obtidos estão associados aos conceitos expandidos. Por fim, os dados que não tiverem correspondência serão considerados resultados da consulta original. Desta forma, é possível distinguir as respostas expandidas das originais.

Para ilustrar como os resultados são exibidos pelo módulo *Classificação dos resultados*, considere a consulta da Figura 4.29, resultante da expansão da consulta por

componentes do produto *desktop*. Os parâmetros utilizados foram  $minMembership = 0.0$ ,  $minSimilarity = 0.5$  e  $minCloseness = 0.6$ .

```

SELECT Todo.id, Todo.nome, Comp.id, Comp.nome, Comp.preco
FROM Produto Comp, Produto Todo, Composto C, Composto C_1
WHERE (((C.idparte = C_1.idtodo) AND (C_1.idparte = Comp.id))
AND (Todo.id = C.idtodo))
AND (((Todo.nome = 'desktop') OR (Todo.nome = 'pc'))
OR (((Todo.nome = 'notebook') OR (Todo.nome = 'laptop'))
OR (Todo.nome = 'pda')) OR (Todo.nome = 'smart phone'))))
UNION
( SELECT Todo.id, Todo.nome, Comp.id, Comp.nome, Comp.preco
FROM Produto Comp, Produto Todo, Composto C
WHERE (((Comp.id = C.idparte) AND (Todo.id = C.idtodo))
AND (((Todo.nome = 'desktop') OR (Todo.nome = 'pc'))
OR (((Todo.nome = 'notebook') OR (Todo.nome = 'laptop'))
OR (Todo.nome = 'pda')) OR (Todo.nome = 'smart phone'))))
)

```

**Figura 4.29 – Consulta resultante das expansões de desktop e fuz:hasPart.**

Considerando a consulta da Figura 4.29, foram realizadas as seguintes expansões semânticas:

- *Similaridade*: *pc* é similar a *desktop*, *laptop* é similar a *notebook* e *smart phone* é similar a *pda*;
- *Proximidade*: *notebook* e *pda* são próximos a *desktop*;
- *Transitividade*: *desktop* e *notebook* possuem no máximo um nível de transitividade para o relacionamento *fuz:hasPart* (mapeado para o atributo *idtodo*).

Para cada uma dessas expansões, FOQuE mantém um registro dos conceitos expandidos e seus respectivos graus difusos, pois estas informações, juntamente com a ontologia difusa e o mapeamento, são utilizadas pelo módulo *Classificação da consulta* para classificar os resultados obtidos. Assim, a Figura 4.30 mostra as respostas da consulta da Figura 4.29 classificadas de acordo com as expansões realizadas e os graus difusos correspondentes.



Original query results:					
Todo.id	Todo.nome	Comp.id	Comp.nome	Comp.preco	
166	desktop	41	monitor	370.00	1.0
166	desktop	43	memoria	93.84	1.0
166	desktop	46	hard disk	160.00	1.0
166	desktop	48	placa mae onboard	236.90	1.0
166	desktop	49	mouse	15.60	1.0
166	desktop	52	teclado	15.00	1.0
166	desktop	54	processador	154.00	1.0
166	desktop	105	cooler	21.92	1.0
166	desktop	173	caixa de som	13.24	0.8
Result augmented by transitivity to desktop					
Transitive relationship: hasPart					
Todo.id	Todo.nome	Comp.id	Comp.nome	Comp.preco	Membership degree
166	desktop	77	placa de rede	13.80	1.0
166	desktop	78	placa de video	102.12	1.0
166	desktop	103	placa de som	42.19	1.0
166	desktop	181	placa de fax modem	40.00	0.5
Result augmented by closeness to desktop					
Todo.id	Todo.nome	Comp.id	Comp.nome	Comp.preco	Closeness degree
124	notebook	54	processador	154.00	0.8745247
124	notebook	77	placa de rede	13.80	0.8745247
124	notebook	78	placa de video	102.12	0.8745247
124	notebook	103	placa de som	42.19	0.8745247
124	notebook	105	cooler	21.92	0.8745247
124	notebook	111	bateria	444.17	0.8745247
124	notebook	118	hard disk	353.13	0.8745247
124	notebook	120	memoria	176.56	0.8745247
124	notebook	150	monitor	499.00	0.8745247
124	notebook	152	mouse	30.00	0.8745247
124	notebook	153	teclado	50.00	0.8745247
124	notebook	154	placa mae onboard	359.00	0.8745247
124	notebook	161	carregador	399.00	0.8745247
124	notebook	181	placa de fax modem	40.00	0.8745247
155	laptop	181	placa de fax modem	40.00	0.8745247
155	laptop	161	carregador	399.00	0.8745247
155	laptop	154	placa mae onboard	359.00	0.8745247
155	laptop	153	teclado	50.00	0.8745247
155	laptop	152	mouse	30.00	0.8745247
155	laptop	150	monitor	499.00	0.8745247
155	laptop	123	processador	206.91	0.8745247
155	laptop	120	memoria	176.56	0.8745247
155	laptop	118	hard disk	353.13	0.8745247
155	laptop	113	ventilador	68.75	0.8745247
155	laptop	112	bateria	474.32	0.8745247
155	laptop	103	placa de som	42.19	0.8745247
155	laptop	78	placa de video	102.12	0.8745247
155	laptop	77	placa de rede	13.80	0.8745247
126	pda	190	caneta stylus	34.90	0.6312056
126	pda	189	placa de rede	199.00	0.6312056
126	pda	188	placa de som	159.00	0.6312056
126	pda	187	placa de video	170.00	0.6312056
126	pda	157	carregador	149.90	0.6312056
126	pda	156	bateria	199.00	0.6312056
126	pda	151	processador	170.00	0.6312056
126	pda	121	teclado	299.00	0.6312056
126	pda	59	memoria	69.90	0.6312056
125	smart phone	157	carregador	149.90	0.6312056
125	smart phone	156	bateria	199.00	0.6312056
125	smart phone	151	processador	170.00	0.6312056
125	smart phone	60	memoria	109.90	0.6312056
Result augmented by similarity to desktop					
Todo.id	Todo.nome	Comp.id	Comp.nome	Comp.preco	Similarity degree
149	pc	40	monitor	345.00	1.0
149	pc	44	memoria	33.12	1.0
149	pc	46	hard disk	160.00	1.0
149	pc	47	placa mae onboard	287.50	1.0
149	pc	49	mouse	15.60	1.0
149	pc	52	teclado	15.00	1.0
149	pc	62	impressora	199.00	1.0
149	pc	77	placa de rede	13.80	1.0
149	pc	78	placa de video	102.12	1.0
149	pc	103	placa de som	42.19	1.0
149	pc	104	ventilador	19.32	1.0
149	pc	123	processador	206.91	1.0
149	pc	170	caixa de som	13.58	1.0
Result augmented by transitivity to notebook					
Transitive relationship: hasPart					
Todo.id	Todo.nome	Comp.id	Comp.nome	Comp.preco	Membership degree
124	notebook	77	placa de rede	13.80	1.0
124	notebook	78	placa de video	102.12	1.0
124	notebook	103	placa de som	42.19	1.0
124	notebook	181	placa de fax modem	40.00	0.5

Figura 4.30 – Resultados classificados para a consulta expandida.

Conforme apresentado na Figura 4.30, o primeiro resultado a ser exibido é o da consulta original, cujos dados não resultam das expansões. No exemplo, este resultado contém os componentes de *desktop* sem considerar a transitividade, sendo que os valores mostrados na última coluna correspondem ao grau de pertinência destes componentes para *desktop*. Em seqüência, são ilustradas as expansões efetuadas, cujas respostas são ordenadas de forma decrescente segundo o parâmetro utilizado na expansão. Algumas tuplas estão repetidas nos resultados expandidos porque elas resultam de mais de um tipo de expansão, porém FOQuE pode ser configurado para não repetir as tuplas expandidas. Considerando a expansão por transitividade a *desktop*, foram recuperados os componentes que pertencem ao 1º nível de transitividade do relacionamento *fuz:hasPart*, ordenados pelo grau de pertinência (*Membership degree*). No caso da proximidade todo-parte, foram obtidos componentes de *notebook*, *laptop*, *pda* e *smart phone*, sendo que a coluna *Closeness degree* mostra os respectivos graus de proximidade todo-parte para *desktop* em ordem decrescente. Os produtos *laptop* e *smart phone* foram considerados próximos a *desktop* com o mesmo grau de proximidade que *notebook* e *pda*, por serem respectivamente similares a estes conceitos. Já a expansão por similaridade de *desktop* recuperou os componentes do produto *pc*, pois *pc* é similar a *desktop* com  $\mu_{similarTo}(pc, desktop) = 1$ , conforme a coluna *Similarity degree*. Por fim, como a expansão por transitividade também é realizada para os conceitos expandidos, foram obtidos os componentes de *notebook* por transitividade, ordenados por *Membership degree*.

Portanto, o módulo *Classificação dos resultados* permite que os resultados sejam classificados em ordem decrescente de relevância para cada tipo de expansão realizada. Deste modo, o usuário pode compreender melhor quais as expansões feitas, além de comparar as respostas expandidas com as respostas da consulta original.

## 4.8. CONSIDERAÇÕES FINAIS

Este capítulo apresentou as principais características do sistema FOQuE para expansão semântica de consultas com base em ontologias difusas. Cada elemento de sua arquitetura foi detalhado, juntamente com algoritmos e exemplos ilustrativos para facilitar a compreensão da abordagem desenvolvida.

No próximo capítulo, serão apresentados o protótipo implementado e os testes realizados considerando um domínio de aplicação, além de uma discussão sobre as contribuições e limitações do sistema FOQuE com base nos resultados obtidos.

## 5. IMPLEMENTAÇÃO, TESTES E DISCUSSÃO DOS RESULTADOS

### 5.1. CONSIDERAÇÕES INICIAIS

Conforme apresentado no capítulo 4, o sistema FOQuE é capaz de efetuar diversos tipos de expansão de consulta baseada em ontologias difusas, com o intuito de recuperar respostas aproximadas que sejam semanticamente relevantes para o usuário. Para avaliar a aplicabilidade de FOQuE, suas contribuições e limitações, implementou-se um protótipo do sistema, em que o usuário pode especificar suas consultas em SQL e obter os resultados expandidos, classificados de acordo com cada tipo de expansão efetuada. A partir deste protótipo, foram realizados testes considerando dados referentes a alunos de disciplinas da área de computação, com o objetivo de analisar as respostas resultantes da expansão de diversas consultas. Assim, este capítulo apresenta o protótipo desenvolvido e os testes realizados, além de uma discussão sobre as contribuições e limitações do sistema com base nos resultados obtidos.

### 5.2. PROTÓTIPO DESENVOLVIDO

Construiu-se um protótipo do sistema FOQuE com o objetivo de expandir semanticamente as consultas do usuário e classificar os resultados obtidos de acordo com os tipos de expansão aplicados. Para a implementação do protótipo, utilizou-se tecnologia Java e o *framework Jena* [CARROLL *et al.*, 2004], voltado para desenvolvimento de aplicações baseadas em ontologias. As inferências sobre a ontologia e as regras de mapeamento são realizadas pela máquina de inferência (*reasoner*) do *framework Jena*, que oferece suporte a ontologias em OWL DL e também ao raciocínio sobre regras (*Jena rules*). O *framework Jena* também serviu de base para a implementação dos mecanismos de inferência sobre conceitos e relacionamentos difusos, que analisam tanto a meta-ontologia difusa quanto a ontologia difusa de domínio para derivar informações imprecisas. Com relação ao Sistema Gerenciador de Banco de Dados (SGBD), utilizou-se o SGBD PostgreSQL, porém qualquer outro tipo de SGBD relacional pode ser considerado, basta que exista um *driver* apropriado para aplicações

Java. A Figura 5.1 mostra a interface gráfica do protótipo, que permite especificar consultas em SQL e visualizar os resultados obtidos, classificados de acordo com as expansões realizadas.

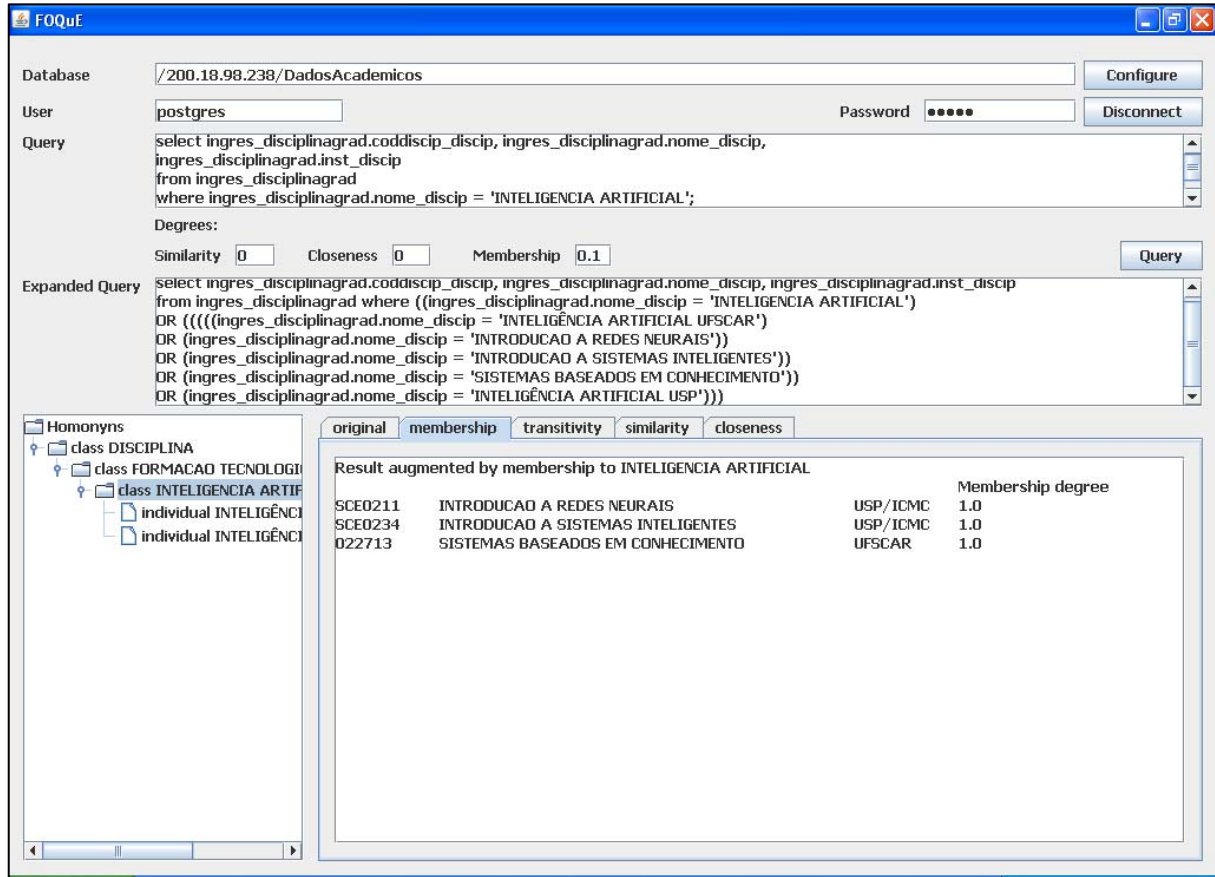


Figura 5.1 – Interface gráfica do protótipo do sistema FOQuE.

O protótipo do sistema FOQuE possibilita, de acordo com a Figura 5.1, que o usuário configure qual o banco de dados a ser consultado, além de outras informações importantes como a ontologia difusa e o mapeamento utilizados (botão *Configure*). Efetuadas estas configurações, o usuário pode especificar consultas SQL e os parâmetros mínimos para as expansões (*Similarity*, *Closeness*, *Membership*). O protótipo, então, gera a consulta modificada (*Expanded query*) e exibe seus resultados em diversas abas, tanto para as respostas obtidas sem expansão (aba *original*) quanto para as respostas expandidas (abas *membership*, *transitivity*, *similarity* e *closeness*). Quando a consulta possui termos homônimos, os diversos conceitos envolvidos são exibidos em uma árvore que contém os caminhos de cada um destes conceitos até a superclasse mais genérica da ontologia difusa. Assim, ao escolher um dos conceitos homônimos, os resultados das demais abas são alterados em função do conceito selecionado.

Vale ressaltar que a interface gráfica do protótipo ilustrada na Figura 5.1 acessa os módulos do sistema FOQuE por meio de uma interface para aplicações (API *FOQuE.jar*), desenvolvida para ser independente de aplicação. Deste modo, outras aplicações Java podem acessar as funcionalidades do sistema FOQuE através do uso desta API.

### 5.3. TESTES

Considerando o protótipo desenvolvido, realizaram-se testes para avaliar a aplicabilidade do sistema FOQuE e identificar suas principais contribuições e limitações. O domínio de aplicação escolhido descreve disciplinas da área de computação, suas categorias, requisitos e ementas. São consideradas também informações sobre alunos matriculados em cursos de computação que cursaram estas disciplinas. O banco de dados utilizado contém dados sobre disciplinas e alunos de computação da Universidade Federal de São Carlos (UFSCar), obtidas através da Pró-Reitoria de Graduação (ProGrad). Além disso, foram coletados dados referentes às disciplinas de computação oferecidas pela Universidade de São Paulo (USP), campus de São Carlos, através do *site* desta instituição ([www.sc.usp.br](http://www.sc.usp.br)). Como a UFSCar permite que seus alunos cursem disciplinas eletivas na USP, desde que tenham completado os requisitos necessários, os dados sobre disciplinas e alunos de computação da UFSCar foram integrados aos dados das disciplinas da computação da USP. Neste contexto, o sistema FOQuE pode ser utilizado para recomendar, por exemplo, as disciplinas de computação da USP que um determinado aluno de computação da UFSCar poderia cursar, através da análise de proximidade todo-parte entre disciplinas com base nos tópicos que constituem as ementas. Além disso, o domínio escolhido permite que outros tipos de expansão semântica também sejam aplicados, como a expansão de homônimos, classes, similaridade e transitividade.

As próximas seções descrevem mais detalhes sobre os testes, como o esquema do banco de dados utilizado, a ontologia difusa do domínio, o mapeamento entre eles e as consultas que ilustram as expansões semânticas efetuadas.

### 5.3.1. Banco de dados

Conforme já mencionado, o banco de dados utilizado contém informações sobre as disciplinas e alunos de computação da UFSCar, além de dados das disciplinas de computação da USP São Carlos. As disciplinas de computação da UFSCar que foram analisadas pertencem aos currículos dos cursos de Bacharelado em Ciência da Computação (currículo do ano de 1994) e de Engenharia da Computação (currículo do ano de 1999), e as disciplinas de computação da USP pertencem ao currículo de 2007 do curso de Bacharelado em Ciência da Computação. Considerou-se uma visão sobre o banco de dados original dos alunos da UFSCar fornecido pela ProGrad, para obter dados dos alunos matriculados nos cursos de Bacharelado em Ciência da Computação e Engenharia da Computação da UFSCar, suas notas e faltas nas disciplinas de computação. Foram realizadas algumas alterações nesta visão para que fosse possível cadastrar as disciplinas de computação da USP, os tópicos de ementa de cada disciplina da computação e seus respectivos pré-requisitos. A Figura 5.2 mostra o esquema referente à visão modificada, que corresponde ao banco de dados utilizado nos testes.

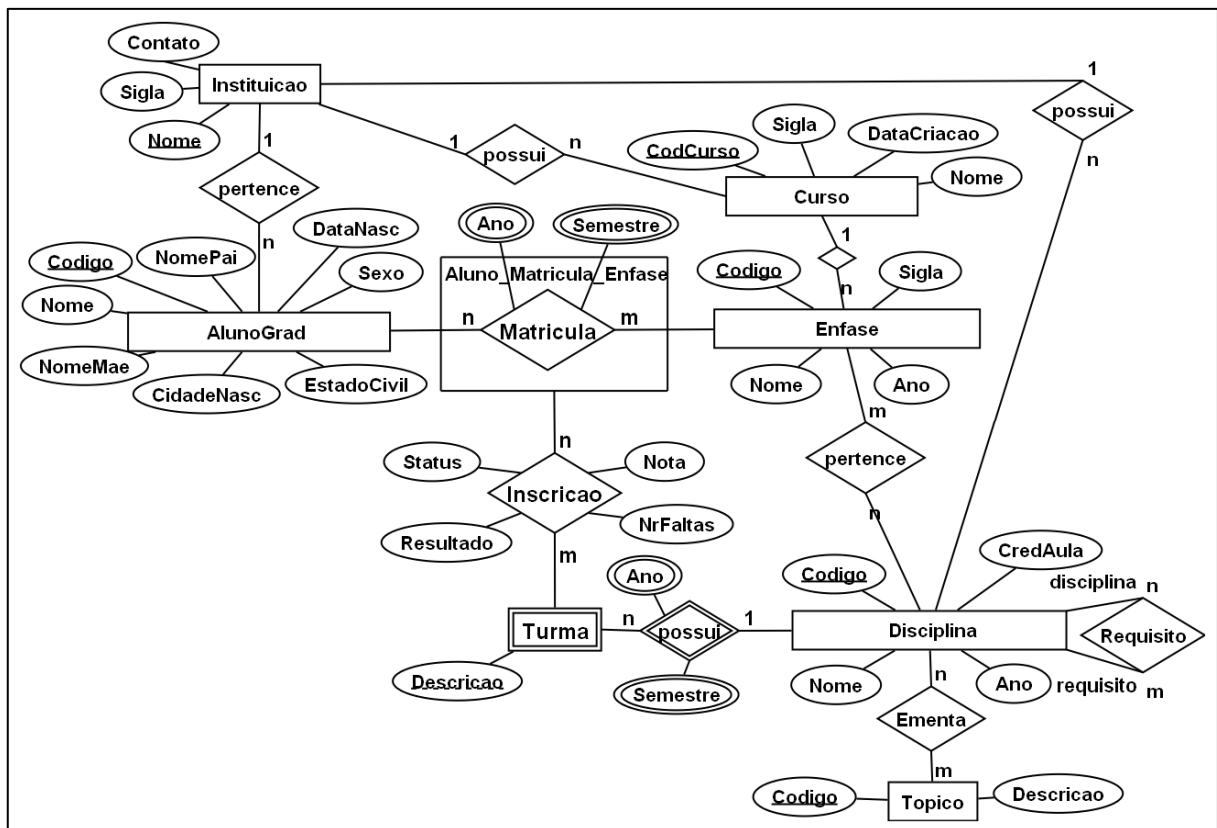


Figura 5.2 – Visão conceitual do esquema do banco de dados utilizado nos testes.

No esquema ilustrado, *AlunoGrad* contém os dados pessoais dos alunos que estão matriculados em uma *Ênfase* ou currículo de um determinado *Curso*. Os alunos matriculados podem fazer *Inscrições* em *Turmas* de uma determinada *Disciplina*, sendo que cada inscrição contém informações sobre seu *Status* (trancada, indeferida, etc) e seu *Resultado* (aprovado, reprovado, etc). As disciplinas podem ter *Requisitos* (pré-requisitos) e possuem uma ementa composta por *Tópicos*. Por fim, alunos, cursos e disciplinas pertencem a uma determinada *Instituição* que, no caso, pode ser UFSCar ou USP.

Uma vez definido qual o banco de dados a ser consultado, o próximo passo é especificar a ontologia difusa que representa a semântica de domínio correspondente aos dados armazenados. Assim, a Seção 5.3.2 apresenta a ontologia difusa referente aos dados de disciplinas de computação.

### 5.3.2. Ontologia difusa

De forma geral, a ontologia difusa utilizada nos testes descreve semanticamente as categorias de disciplinas da área de computação, seus requisitos e tópicos de ementa. Para construir a ontologia, utilizou-se como base as diretrizes curriculares de cursos da área de computação e informática [MEC, 1999], elaboradas por membros do Ministério da Educação (MEC) e da Sociedade Brasileira de Computação (SBC). Além disso, foram analisados o relatório de avaliação do MEC para o curso de Bacharelado em Ciência da Computação da UFSCar [UFSCAR, 2000] e ementas das disciplinas de computação oferecidas pela UFSCar e pela USP. Assim, a ontologia utilizada nos testes foi elaborada manualmente segundo a metodologia de criação de ontologias *Ontology Development 101* [NOY; MCGUINNESS, 2001], com base em documentos oficiais elaborados por especialistas do domínio considerado. A Figura 5.3 mostra uma visão geral da ontologia difusa construída, contendo classes que representam as categorias de disciplinas da computação e alguns relacionamentos do domínio.

De acordo com a Figura 5.3, as subclasses da classe *Disciplina* representam as categorias de disciplinas de computação e informática definidas pelas diretrizes curriculares do MEC [MEC, 1999]. Estas categorias são consideradas classes difusas (subclassess da classe *FuzzyConcept* da meta-ontologia difusa), pois as disciplinas podem ter um grau de pertinência definido com base nas horas-aula para cada uma das categorias. Por exemplo, segundo o



relatório de avaliação do MEC para o curso de Bacharelado em Ciência da Computação da UFSCar [UFSCAR, 2000], a disciplina *Estruturas de Dados* possui 40 horas-aula de *Programação* e 20 horas-aula de *Computação e Algoritmos*. Assim, *Estruturas de Dados* pode pertencer a cada uma destas categorias com um determinado grau de pertinência que reflete a distribuição de horas-aula.

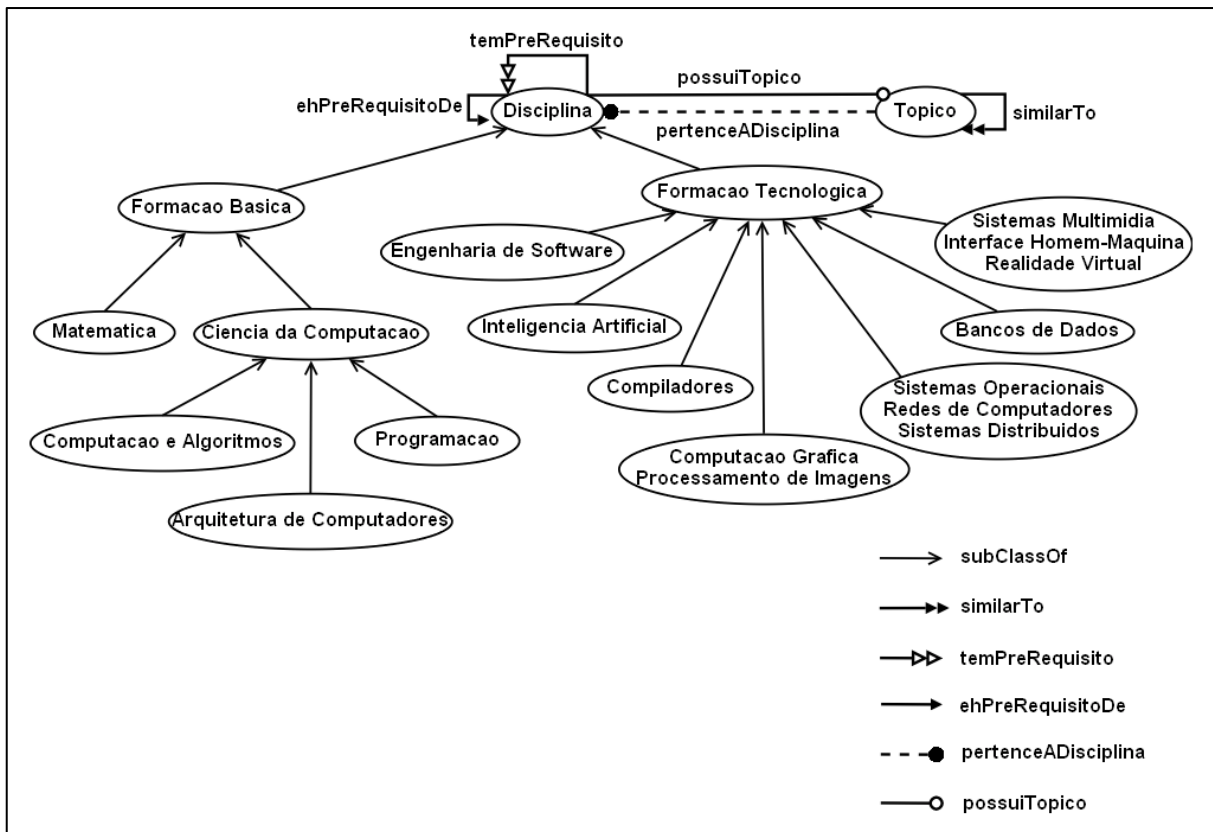


Figura 5.3 – Ontologia difusa sobre disciplinas da área de computação e informática.

Quanto aos relacionamentos da ontologia, foram definidos relacionamentos transitivos representando os requisitos das disciplinas (*temPreRequisito* – *ehPreRequisitoDe*). Os requisitos recomendados, que não são obrigatórios mas possuem um conteúdo que fundamenta outras disciplinas, são difusos, ou seja, são sub-relacionamentos de *fuz:FuzzyRelation* da meta-ontologia difusa. Assim, disciplinas que representam requisitos recomendados possuem um determinado grau dentro do intervalo  $[0,1]$ , enquanto os requisitos obrigatórios possuem grau de pertinência máximo ( $\mu = 1$ ). Além de requisitos, as disciplinas possuem tópicos de ementa (*possuiTopico* – *pertenceADisciplina*), que corresponde a um relacionamento todo-parte no domínio, sendo, portanto, sub-relacionamentos de *fuz:hasPart* e *fuz:partOf* respectivamente. Deste modo, é possível inferir disciplinas próximas entre si em função de seus tópicos de ementa, através da análise de proximidade todo-parte. Por fim, os

tópicos de diferentes disciplinas podem ser similares entre si com um determinado grau de similaridade através do relacionamento difuso *similarTo* (sub-relacionamento de *fuz:FuzzyRelation* da meta-ontologia difusa).

### 5.3.3. Mapeamento

Uma vez definidos o banco de dados e a ontologia difusa, é necessário estabelecer o mapeamento entre os seus elementos de modo que o sistema FOQuE seja capaz de representar os conceitos expandidos em termos de tabelas e atributos do banco de dados. Considerando a ontologia e o banco de dados utilizados nos testes, foram elaboradas as seguintes correspondências:

- A tabela *ingres\_disciplinagrad*, que representa a entidade *Disciplina* do esquema conceitual da Figura 5.2, corresponde às subclasses e instâncias da classe *Disciplina* da ontologia. Assim, o atributo *nome\_discip* (nome da disciplina) da tabela *ingres\_disciplinagrad* é mapeado para a propriedade *rdfs:label* que rotula os nomes das subclasses e instâncias da classe *Disciplina*:

$$rdfs:subClassOf(x, Disciplina), rdfs:label(x,y) \rightarrow ingres\_disciplinagrad(x), nome\_discip(x,y)$$

$$Disciplina(x), rdfs:label(x,y) \rightarrow ingres\_disciplinagrad(x), nome\_discip(x,y)$$

- A entidade *Tópico*, representada pela tabela *ingres\_ementagrad*, está associada às instâncias da classe *Topico* da ontologia. Definiu-se, então, a regra de mapeamento entre o atributo *topico\_ementa* (descrição do tópico) da tabela *ingres\_ementagrad* e a propriedade *rdfs:label* que contém a descrição das instâncias da classe *Topico* da ontologia:

$$Topico(x), rdfs:label(x,y) \rightarrow ingres\_ementagrad(x), topico\_ementa(x,y)$$

- A tabela *ingres\_requisitograd* corresponde aos relacionamentos transitivos *temPreRequisito* e *ehPreRequisitoDe*, através de seus atributos *coddiscip\_requisito* e *codrequisito\_requisito* que constituem chaves estrangeiras para a tabela *ingres\_disciplinagrad* (entidade *Disciplina*). Assim, de acordo com a abordagem citada na Seção 4.4 para associar um atributo de tabela com um relacionamento da ontologia, foram definidas as seguintes regras de mapeamento:

$$temPreRequisito(x,y) \rightarrow ingres\_requisitograd(z), coddiscip\_requisito(z,x)$$

$ehPreRequisitoDe(x,y) \rightarrow ingres\_requisitograd(z), codrequisito\_requisito(z,x)$

- No caso de homônimos, consideraram-se as disciplinas da UFSCar e da USP que possuem o mesmo nome, mas representam conceitos diferentes na ontologia porque possuem pré-requisitos e tópicos de ementa distintos dependendo da instituição. Para remover a ambigüidade dos dados utilizou-se o atributo *inst\_discip* da tabela *ingres\_disciplinagrad*, que especifica a qual instituição uma disciplina pertence. Assim, foram definidas regras de mapeamento para cada disciplina que possui o mesmo nome tanto para a UFSCar quanto para a USP São Carlos. Um exemplo é a disciplina *banco de dados*:

$Banco\_Dados\_UFSCAR \rightarrow ingres\_disciplinagrad(x), nome\_discip(x, 'BANCO DE DADOS'), inst\_discip(x, 'UFSCAR')$

$Banco\_Dados\_USP \rightarrow ingres\_disciplinagrad(x), nome\_discip(x, 'BANCO DE DADOS'), inst\_discip(x, 'USP/ICMC')$

sendo que *Banco\_Dados\_UFSCAR* e *Banco\_Dados\_USP* são as instâncias da ontologia que representam respectivamente a disciplina *banco de dados* oferecida pela UFSCar e pela USP São Carlos. Outras disciplinas que também possuem esse tipo de mapeamento por serem consideradas conceitos homônimos com relação às diferentes instituições são *redes de computadores*, *inteligência artificial*, *arquitetura de computadores* e *computação gráfica*.

Vale ressaltar que as regras de mapeamento devem ser construídas por especialistas do domínio que conhecem tanto o banco de dados quanto a ontologia para que sejam definidas regras adequadas para a expansão de consultas do sistema FOQuE. A partir das regras de mapeamento apresentadas nesta seção, o sistema FOQuE pode expandir as consultas realizadas ao banco de dados ilustrado na Seção 5.3.1, com base na ontologia difusa da Seção 5.3.2. A próxima seção (Seção 5.3.4) descreve algumas consultas submetidas ao protótipo do sistema FOQuE, detalhando as expansões semânticas efetuadas e os resultados obtidos para o banco de dados, ontologia difusa e mapeamentos considerados nos testes.

### 5.3.4. Consultas realizadas

Para mostrar as expansões semânticas e suas contribuições para melhorar os resultados das consultas, foram realizadas diversas consultas ao protótipo do sistema FOQuE considerando o banco de dados, a ontologia difusa e as regras de mapeamento das Seções 5.3.1 a 5.3.3. A seguir, são ilustradas consultas para cada tipo de expansão e, ao final, consultas que envolvem várias expansões.

#### 5.3.4.1 Homônimos

Suponha que o usuário especifique uma consulta por disciplinas pertencentes à categoria *Inteligência artificial*, ilustrada na Figura 5.4.

<pre>select ingres_disciplinagrado.coddiscip_discip, ingres_disciplinagrado.nome_discip, ingres_disciplinagrado.inst_discip from ingres_disciplinagrado where ingres_disciplinagrado.nome_discip = 'INTELEGENCIA ARTIFICIAL';</pre>		
Degrees:		
Similarity	<input type="text" value="0"/>	Closeness
	<input type="text" value="0"/>	Membership
		<input type="text" value="0.1"/>

Figura 5.4 – Consulta por disciplinas da categoria *Inteligência artificial*.

De acordo com a ontologia difusa e o mapeamento, o termo *Inteligência artificial* é homônimo porque corresponde a três conceitos diferentes: classe *Inteligência artificial*, que descreve uma categoria de disciplinas da computação; instância *Inteligência artificial UFSCAR*, que representa a disciplina de inteligência artificial oferecida pela UFSCar; e instância *Inteligência artificial USP*, correspondente à disciplina oferecida pela USP. A Figura 5.5 mostra estes conceitos da ontologia difusa.

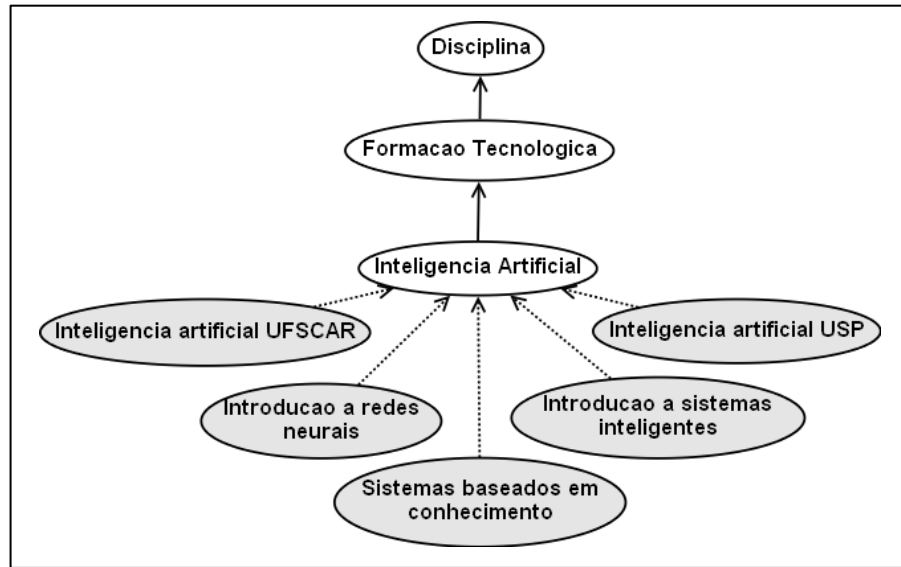


Figura 5.5 – Conceitos da ontologia difusa relacionados ao termo homônimo *Inteligência artificial*.

Assim, o sistema FOQuE mostra as três opções de homônimos ao usuário (árvore de conceitos situada ao lado esquerdo da Figura 5.6), que então escolhe o contexto mais apropriado, ou seja, a classe *Inteligência artificial* que corresponde à categoria de disciplinas *Inteligência artificial*. Como o conceito escolhido é uma classe, também é feita a expansão de classes, que analisa todas as instâncias de *Inteligência artificial* para reformular a consulta. A Figura 5.6 mostra a consulta expandida e os resultados obtidos.

Expanded Query

```

select ingres_disciplinagrado.coddisc_discip, ingres_disciplinagrado.nome_discip, ingres_disciplinagrado.inst_discip
from ingres_disciplinagrado where ((ingres_disciplinagrado.nome_discip = 'INTELEGENCIA ARTIFICIAL')
OR (((((ingres_disciplinagrado.nome_discip = 'INTELEGÊNCIA ARTIFICIAL UFSCAR')
OR (ingres_disciplinagrado.nome_discip = 'INTRODUCAO A REDES NEURAIIS')
OR (ingres_disciplinagrado.nome_discip = 'INTRODUCAO A SISTEMAS INTELIGENTES'))
OR (ingres_disciplinagrado.nome_discip = 'SISTEMAS BASEADOS EM CONHECIMENTO'))
OR (ingres_disciplinagrado.nome_discip = 'INTELEGÊNCIA ARTIFICIAL USP'))))
  
```

Homonyms

- class DISCIPLINA
  - class FORMACAO TECNOLGI
    - class INTELEGENCIA ARTIF
      - individual INTELEGÊNCI
      - individual INTELEGÊNCI

original membership transitivity similarity closeness

Original query results:

original	membership	transitivity	similarity	closeness
022705	INTELEGENCIA ARTIFICIAL			UFSCAR
SCE0163	INTELEGENCIA ARTIFICIAL			USP/ICMC

original membership transitivity similarity closeness

Result augmented by membership to INTELEGENCIA ARTIFICIAL

original	membership	transitivity	similarity	closeness	Membership degree
SCE0211	INTRODUCAO A REDES NEURAIIS			USP/ICMC	1.0
SCE0234	INTRODUCAO A SISTEMAS INTELIGENTES			USP/ICMC	1.0
022713	SISTEMAS BASEADOS EM CONHECIMENTO			UFSCAR	1.0

Figura 5.6 – Consulta expandida para recuperar disciplinas da classe *Inteligência artificial*.

Comparando as respostas resultantes da expansão com as obtidas pela consulta original, é possível observar melhores resultados principalmente com relação à revocação, pois a consulta expandida retorna disciplinas pertencentes à categoria *Inteligência artificial* que não são obtidas sem a expansão. É o caso das disciplinas *Introdução a redes neurais*,

*Sistemas baseados em conhecimento e Introdução a sistemas inteligentes*, que somente são recuperadas em virtude da expansão de classes. A Tabela 5.1 mostra os resultados em termos de revocação e precisão, obtidos tanto para a consulta original da Figura 5.4 quanto para a consulta expandida.

**Tabela 5.1 – Revocação e precisão dos resultados obtidos pela consulta original e pela consulta expandida.**

	<b>Revocação</b>	<b>Precisão</b>
<b>Consulta original</b>	$\frac{2}{5} = 40\%$	$\frac{2}{2} = 100\%$
<b>Consulta expandida</b>	$\frac{5}{5} = 100\%$	$\frac{5}{5} = 100\%$

Na Tabela 5.1, a revocação é obtida através da divisão da quantidade de respostas relevantes recuperadas pelo total de respostas que são relevantes para a consulta. A precisão, por sua vez, resulta da divisão do total de respostas relevantes que foram recuperadas pelo total de respostas obtidas. Assim, a consulta expandida referente ao contexto escolhido pelo usuário (classe *Inteligência artificial*) obtém melhores resultados em termos de revocação, visto que possibilita recuperar todas as disciplinas pertencentes à categoria *Inteligência artificial*, e não somente aquelas que possuem exatamente esse nome. Além disso, mesmo com o aumento significativo da revocação, os resultados em termos de precisão não são prejudicados, uma vez que todas as disciplinas recuperadas são pertinentes ao contexto da consulta do usuário.

#### 5.3.4.2 Classes

Para mostrar uma expansão de classes que envolve graus de pertinência, considere a consulta por disciplinas pertencentes à categoria *Programação* com  $\text{minMembership} = 0.65$  (Figura 5.7).

```
select ingres_disciplinagrado.coddiscip_discip, ingres_disciplinagrado.nome_discip, ingres_disciplinagrado.inst_discip
from ingres_disciplinagrado
where ingres_disciplinagrado.nome_discip = 'PROGRAMACAO';
```

Degrees:  
 Similarity  Closeness  Membership

Figura 5.7 – Consulta por disciplinas da categoria *Programação*.

Na ontologia difusa, algumas disciplinas possuem um determinado grau de pertinência para as classes às quais pertencem, representando a quantidade de horas-aula para uma categoria de disciplinas. Por exemplo, segundo o relatório de avaliação do MEC para o curso de Bacharelado em Ciência da Computação da UFSCar [UFSCAR, 2000], a disciplina *Estruturas de Dados* possui 40 horas-aula de *Programação* e 20 horas-aula de *Computação e Algoritmos*. Assim, pode-se representar na ontologia difusa que *Estruturas de Dados* pertence à classe *Programação* com  $\mu_{Programação}(Estruturas\ de\ Dados) = 0.66$  e à classe *Computação e Algoritmos* com  $\mu_{Computação\ e\ Algoritmos}(Estruturas\ de\ Dados) = 0.34$ . Para obter esses valores de horas-aula, utilizou-se como referência o relatório de avaliação do MEC [UFSCAR, 2000] para as disciplinas da UFSCar, e estimou-se um valor de horas-aula para as disciplinas de computação da USP, com base no conteúdo de suas ementas. A Figura 5.8 mostra as disciplinas pertencentes à classe *Programação* da ontologia difusa e seus respectivos graus de pertinência em função da distribuição de horas-aula.

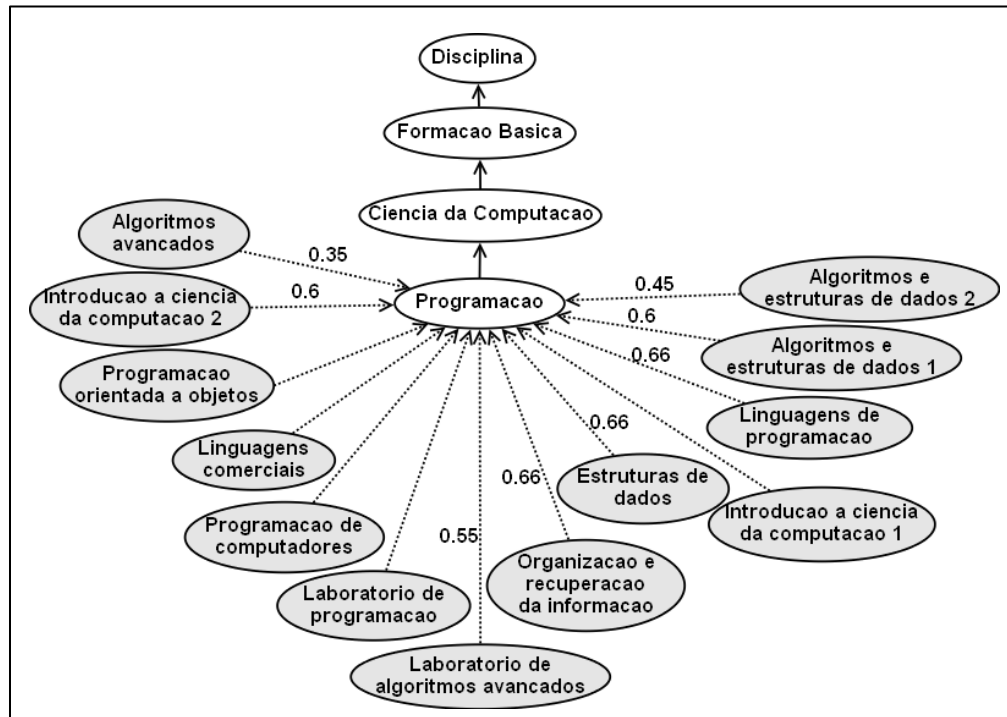


Figura 5.8 – Disciplinas pertencentes à classe *Programação*.

Com base na ontologia difusa e no mapeamento, o sistema FOQuE expande a consulta por disciplinas pertencentes à categoria *Programação* segundo a estratégia apresentada na Seção 4.6.2. A Figura 5.9 mostra a consulta expandida e seus resultados.

```

select ingres_disciplinagrado.coddiscip_discip, ingres_disciplinagrado.nome_discip, ingres_disciplinagrado.inst_discip
from ingres_disciplinagrado where ((ingres_disciplinagrado.nome_discip = 'PROGRAMACAO')
OR (((((((ingres_disciplinagrado.nome_discip = 'INTRODUCAO A CIENCIA DA COMPUTACAO 1')
OR (ingres_disciplinagrado.nome_discip = 'ESTRUTURAS DE DADOS'))
OR (ingres_disciplinagrado.nome_discip = 'LINGUAGENS DE PROGRAMACAO'))
OR (ingres_disciplinagrado.nome_discip = 'PROGRAMACAO ORIENTADA A OBJETOS'))
OR (ingres_disciplinagrado.nome_discip = 'LABORATORIO DE PROGRAMACAO'))

```

original	membership	transitivity	similarity	closeness
Result augmented by membership to PROGRAMACAO				
				Membership degree
SCE0180	INTRODUCAO A CIENCIA DA COMPUTACAO 1		USP/ICMC	1.0
020672	LINGUAGENS COMERCIAIS		UFSCAR	1.0
020940	LABORATORIO DE PROGRAMACAO		UFSCAR	1.0
025917	LINGUAGENS COMERCIAIS		UFSCAR	1.0
025062	LABORATORIO DE PROGRAMACAO		UFSCAR	1.0
025020	PROGRAMACAO DE COMPUTADORES		UFSCAR	1.0
SCE0213	PROGRAMACAO ORIENTADA A OBJETOS		USP/ICMC	1.0
022624	ORGANIZACAO E RECUPERACAO DA INFORMACAO		UFSCAR	0.66
025100	LINGUAGENS DE PROGRAMACAO		UFSCAR	0.66
025208	ESTRUTURAS DE DADOS		UFSCAR	0.66
025259	ESTRUTURAS DE DADOS		UFSCAR	0.66
020966	ESTRUTURAS DE DADOS		UFSCAR	0.66
020117	LINGUAGENS DE PROGRAMACAO		UFSCAR	0.66
022667	ORGANIZACAO E RECUPERACAO DA INFORMACAO		UFSCAR	0.66
020010	ESTRUTURAS DE DADOS		UFSCAR	0.66

Figura 5.9 – Consulta expandida para recuperar disciplinas da classe *Programação*.

Conforme ilustrado na Figura 5.9, a consulta expandida obtém as disciplinas cujo grau de pertinência para a classe *Programação* seja maior ou igual a *minMembership*. Além disso, o resultado é ordenado de forma decrescente de acordo com os graus de pertinência, para que o usuário identifique as respostas mais relevantes. A consulta original (Figura 5.7) não retorna respostas, pois o banco de dados não possui disciplinas de nome *Programação*, mas disciplinas que pertencem a esta classe. Deste modo, os resultados obtidos com a expansão proporcionam uma melhoria significativa tanto para a revocação quanto para a precisão, visto que a consulta original retorna nenhuma resposta enquanto a expandida é capaz de recuperar todas as disciplinas requisitadas pelo usuário.

### 5.3.4.3 Similaridade

A ontologia difusa utilizada nos testes possui relacionamentos de similaridade entre tópicos de disciplinas da UFSCar e da USP, para que seja possível analisar a proximidade todo-parte entre disciplinas das diferentes instituições com base nos tópicos de



ementa. Assim, entrevistaram-se especialistas do domínio, no caso os professores de computação da UFSCar, para definir graus de similaridade apropriados entre os tópicos. A Tabela 5.2 mostra alguns tópicos similares entre si, as disciplinas às quais pertencem e os graus de similaridade correspondentes.

**Tabela 5.2 – Similaridade entre tópicos de disciplinas.**

Tópico			Tópico similar			$\mu_{similarTo}$
ID	Descrição	Disciplina	ID	Descrição	Disciplina	
<i>topico1</i>	Implementação de Sistemas	Laboratório de banco de dados (UFSCar)	<i>topico2</i>	Aspectos de implementação dos SGBDs relacionais (indexação, métodos de acesso e "bufferização")	Banco de dados (USP)	1.0
<i>topico2</i>	Aspectos de implementação dos SGBDs relacionais (indexação, métodos de acesso e "bufferização")	Banco de dados (USP)	<i>topico3</i>	Projeto de Implementação: Componentes do Projeto de Implementação; Projeto Físico: Passos do Projeto Físico; Considerações sobre o Projeto Físico	Projeto de banco de dados (UFSCar)	0.7
<i>topico4</i>	Modelagem de Dados, Modelos de Dados, Modelos de Dados Hierárquicos e de Redes	Banco de dados (UFSCar)	<i>topico5</i>	Modelos conceituais e modelos externos	Banco de dados (USP)	1.0
<i>topico5</i>	Modelos conceituais e modelos externos	Banco de dados (USP)	<i>topico6</i>	Conceitos Avançados de Modelagem de Dados	Projeto de banco de dados (UFSCar)	0.75

Além dos relacionamentos de similaridade ilustrados na Tabela 5.2, o sistema FOQuE é capaz de inferir graus de similaridade com base na composição *max-min*, conforme apresentado na Seção 4.6.3. Desta forma, é possível inferir que  $\mu_{similarTo}(topico1, topico3) = 0.7$  e  $\mu_{similarTo}(topico4, topico6) = 0.75$ . Para ilustrar a expansão por similaridade, suponha que o usuário deseje consultar por disciplinas que contêm o tópico *Implementação de Sistemas* com  $minSimilarity = 0.7$  (Figura 5.10).

```

select D.coddiscip_discip, D.nome_discip, D.inst_discip, E.topico_ementa
from ingres_disciplinagrad D, ingres_ementagrad E
where E.topico_ementa = 'Implementacao de Sistemas'
AND D.coddiscip_discip = E.coddiscip_ementa;

```

Degrees:

Similarity  Closeness  Membership

```

select D.coddiscip_discip, D.nome_discip, D.inst_discip, E.topico_ementa
from ingres_disciplinagrad D, ingres_ementagrad E
where (((E.topico_ementa = 'Implementacao de Sistemas')
OR ((E.topico_ementa = 'Aspectos de implementacao dos SGBDs relacionais (indexacao, metodos de acesso e "buff
OR (E.topico_ementa = 'Projeto de Implementacao: Componentes do Projeto de Implementacao; Projeto Fisico: Pas
AND (D.coddiscip_discip = E.coddiscip_ementa))

```

original membership transitivity similarity closeness

Original query results:

025224	LABORATORIO DE BANCO DE DADOS	UFSCAR	Implementacao de Sistemas
--------	-------------------------------	--------	---------------------------

original membership transitivity similarity closeness

Result augmented by similarity to Implementacao de Sistemas

SCE0179	BANCO DE DADOS	USP/ICMC	Aspectos de implementacao dos SGBDs rela
025232	PROJETO DE BANCO DE DADOS	UFSCAR	Projeto de Implementacao: Componentes d

Figura 5.10 – Consulta por disciplinas que contém o tópico *Implementação de Sistemas*.

Os resultados apresentados na Figura 5.10 mostram que a consulta original obtém apenas *topico1*, enquanto a consulta expandida permite recuperar os tópicos similares *topico2* e *topico3*, sendo que este último foi considerado em função das inferências lógicas realizadas durante a expansão. Deste modo, é possível obter respostas relevantes que não correspondem necessariamente ao termo especificado na consulta original. Além disso, as informações adicionais podem auxiliar o usuário a compreender melhor o domínio em questão, caso ele não conheça previamente os tópicos similares. Pode-se concluir, portanto, que a consulta expandida por similaridade melhora a revocação dos resultados, sem prejudicar a precisão, pois obtém respostas mais abrangentes respeitando o parâmetro de similaridade mínima especificado pelo usuário.

#### 5.3.4.4 Proximidade todo-parte

Considerando o banco de dados e a ontologia difusa utilizados nos testes, a expansão por proximidade todo-parte pode ser realizada ao tratar os tópicos de ementa como partes das disciplinas. Conforme citado anteriormente, os tópicos de diferentes disciplinas podem ser similares entre si, possibilitando que sejam calculados graus de proximidade todo-parte. Desta forma, as disciplinas que possuem uma quantidade significativa de tópicos em

comum são consideradas semanticamente próximas entre si e, portanto, podem ser recomendadas como respostas relevantes para a consulta do usuário.

Para mostrar um exemplo de consulta expandida por proximidade todo-parte, suponha que um aluno deseje consultar por informações referentes à disciplina *Sistemas distribuídos*, fornecendo os parâmetros  $minMembership = 0.0$ ,  $minSimilarity = 0.5$  e  $minCloseness = 0.2$ . A partir destas variáveis, o sistema FOQuE calcula os graus de proximidade para as disciplinas da ontologia difusa, baseando-se nos tópicos de ementa e no algoritmo citado na Seção 4.6.4. Assim, a Tabela 5.3 apresenta as disciplinas próximas a *Sistemas distribuídos* e os respectivos graus de proximidade todo-parte calculados com base na estratégia da média ponderada (Seção 4.6.4).

**Tabela 5.3 – Disciplinas próximas a *Sistemas distribuídos* (estratégia da média ponderada).**

<b>Disciplina</b>	<b>Instituição</b>	<b>Grau de proximidade todo-parte</b>
Sistemas computacionais distribuídos	USP/ICMC	0.54545456
Computação distribuída	USP/ICMC	0.5
Programação concorrente	USP/ICMC	0.2222222
Sistemas computacionais de tempo real	USP/ICMC	0.2
Linguagens e programação concorrente	UFSCar	0.2

Além da estratégia com base na média ponderada, também foram testadas outras formas para calcular a proximidade todo-parte, como  $\min(x, y)$  e a média aritmética  $ma = \frac{x + y}{2}$ , com  $x = \frac{comum\_il}{total\_il}$  e  $y = \frac{comum\_cl}{total\_cl}$ , definidos na Seção 4.6.4. No caso da abordagem utilizando mínimo, as respostas obtidas não são apropriadas em algumas situações. Por exemplo, suponha que a disciplina  $d1$  tenha um total de 10 tópicos, sendo 3 em comum com a disciplina  $d2$ , que também contém 10 tópicos ao total. Assim,  $prox(d1, d2) = \min(0.3, 0.3) = 0.3$ . Considere então a disciplina  $d3$  com 3 tópicos ao total, sendo todos eles comuns à disciplina  $d2$ . O resultado de  $prox(d2, d3) = \min(0.3, 1.0) = 0.3$ , que é o mesmo resultado de  $prox(d1, d2)$ , embora a disciplina  $d3$  seja mais próxima a  $d2$  que  $d1$ . Deste modo, concluiu-se que a abordagem utilizando mínimo não é adequada para o domínio de disciplinas da computação porque pode retornar graus de proximidade que não correspondem à semântica do domínio.

Quanto à abordagem baseada em média aritmética, os resultados obtidos são em geral bastante parecidos com os resultados da estratégia da média ponderada. A Tabela 5.4

mostra os graus de proximidade calculados com base na média aritmética para o exemplo da consulta por *Sistemas distribuídos*.

**Tabela 5.4 – Disciplinas próximas a *Sistemas distribuídos* (estratégia da média aritmética).**

<b>Disciplina</b>	<b>Instituição</b>	<b>Grau de proximidade todo-parte</b>
Sistemas computacionais distribuídos	USP/ICMC	0.55
Computação distribuída	USP/ICMC	0.53333336
Programação concorrente	USP/ICMC	0.225
Sistemas computacionais de tempo real	USP/ICMC	0.2
Linguagens e programação concorrente	UFSCar	0.2

Embora ambas abordagens obtenham resultados próximos para *Sistemas distribuídos*, observou-se uma situação em que os graus de proximidade calculados com base na média aritmética e na média ponderada possuem valores significativamente diferentes. É o caso da disciplina *Laboratório de compiladores*, que possui um único tópico em sua ementa. Quando comparada à disciplina *Introdução à compilação*, que possui 10 tópicos ao total, ambas disciplinas apresentam um tópico em comum, ou seja, o único tópico da disciplina *Laboratório de compiladores*. Ao utilizar a média aritmética para calcular o grau de proximidade entre essas disciplinas, obtém-se  $\text{prox}(\text{Laboratório de compiladores}, \text{Introdução à compilação}) = 0.5555556$ , enquanto que a abordagem baseada na média ponderada obtém  $\text{prox}(\text{Laboratório de compiladores}, \text{Introdução à compilação}) = 0.2$ . Uma proximidade de 55% entre as disciplinas citadas não é um resultado apropriado com relação à semântica do domínio, uma vez que os tópicos em comum representam apenas 10% para *Introdução à compilação*. Deste modo, utilizou-se a estratégia baseada na média ponderada nos testes, pois seus resultados são semanticamente mais adequados por considerarem a quantidade total de tópicos como peso da média ponderada.

Assim, voltando à consulta pela disciplina *Sistemas distribuídos*, a Figura 5.11 mostra os resultados recuperados por proximidade todo-parte, calculada com base na abordagem que utiliza a média ponderada.

```

select ingres_disciplinagrad.coddiscip_discip, ingres_disciplinagrad.nome_discip,
ingres_disciplinagrad.inst_discip
from ingres_disciplinagrad
where ingres_disciplinagrad.nome_discip = 'SISTEMAS DISTRIBUIDOS';

```

Degrees:

Similarity  Closeness  Membership

```

where ((ingres_disciplinagrad.nome_discip = 'SISTEMAS DISTRIBUIDOS')
OR (((((ingres_disciplinagrad.nome_discip = 'PROGRAMACAO CONCORRENTE')
OR (ingres_disciplinagrad.nome_discip = 'SISTEMAS COMPUTACIONAIS DE TEMPO REAL'))
OR (ingres_disciplinagrad.nome_discip = 'LINGUAGENS E PROGRAMACAO CONCORRENTE'))
OR (ingres_disciplinagrad.nome_discip = 'COMPUTACAO DISTRIBUIDA'))
OR (ingres_disciplinagrad.nome_discip = 'SISTEMAS COMPUTACIONAIS DISTRIBUIDOS'))))

```

original membership transitivity similarity closeness

Original query results:

025321	SISTEMAS DISTRIBUIDOS	UFSCAR
--------	-----------------------	--------

original membership transitivity similarity closeness

Result augmented by closeness to SISTEMAS DISTRIBUIDOS

			Closeness degree
SCE0218	SISTEMAS COMPUTACIONAIS DISTRIBUIDOS	USP/ICMC	0.54545456
SCE0706	COMPUTACAO DISTRIBUIDA	USP/ICMC	0.5
SCE0217	PROGRAMACAO CONCORRENTE	USP/ICMC	0.22222222
SCE0708	SISTEMAS COMPUTACIONAIS DE TEMPO REAL	USP/ICMC	0.2
025135	LINGUAGENS E PROGRAMACAO CONCORRENTE	UFSCAR	0.2

Figura 5.11 – Consulta por disciplinas próximas a Sistemas distribuídos.

A consulta expandida da Figura 5.11 permite recuperar informações semanticamente próximas ao conceito fornecido pelo usuário, sendo que as respostas mais relevantes são listadas em ordem decrescente com relação ao grau de proximidade. Observa-se, portanto, que a consulta expandida por proximidade todo-parte também contribui para melhorar a revocação dos resultados, assim como o exemplo de consulta ilustrado para a expansão por similaridade. Além disso, as respostas obtidas por proximidade todo-parte não afetam a precisão, pois somente são recuperados os dados que satisfazem aos parâmetros de expansão especificados.

#### 5.3.4.5 Transitividade

Para o domínio considerado nos testes, a expansão por transitividade é realizada com relação aos requisitos de uma disciplina. Os relacionamentos *temPreRequisito* e *ehPreRequisitoDe* da ontologia difusa são transitivos, ou seja, dados os relacionamentos *d1 temPreRequisito d2* e *d2 temPreRequisito d3*, pode-se inferir que *d1 temPreRequisito d3*.

Estes relacionamentos podem ser difusos, contendo um grau de pertinência que representa o quanto um pré-requisito recomendado é relevante para uma determinada disciplina. Assim como os relacionamentos de similaridade entre tópicos, os graus de pertinência entre um pré-requisito recomendado e uma disciplina foram definidos por especialistas do domínio, no caso os professores de computação da UFSCar. Por exemplo, a disciplina *Linguagens formais e autômatos* da UFSCar é um pré-requisito recomendado de *Construção de compiladores* com  $\mu_{ehPreRequisitoDe}(Linguagens\ formais\ e\ autômatos, Construção\ de\ compiladores) = 0.3$ . No caso de pré-requisitos obrigatórios,  $\mu_{ehPreRequisitoDe}$  assume o valor máximo ( $\mu_{ehPreRequisitoDe} = 1.0$ ). Desta forma, é possível expandir consultas envolvendo pré-requisitos de disciplinas, a partir dos relacionamentos transitivos *temPreRequisito* e *ehPreRequisitoDe* da ontologia difusa.

A Figura 5.12 mostra um exemplo de consulta que pode ser expandida por transitividade para recuperar os todos os pré-requisitos da disciplina *Laboratório de compiladores*.

<pre>select R.coddiscip_discip, R.nome_discip, R.inst_discip from ingres_disciplinagrad D, ingres_disciplinagrad R, ingres_requisitograd Rel where D.nome_discip = 'LABORATORIO DE COMPILADORES' AND D.coddiscip_discip = Rel.coddiscip_requisito AND R.coddiscip_discip = Rel.codrequisito_requisito;</pre>		
Degrees:		
Similarity	<input type="text" value="1.0"/>	Membership
Closeness	<input type="text" value="1.0"/>	<input type="text" value="0.3"/>

Figura 5.12 – Consulta por pré-requisitos da disciplina *Laboratório de compiladores*.

Como o banco de dados utilizado nos testes mantém somente os pré-requisitos diretos das disciplinas, a consulta da Figura 5.12 não é capaz de recuperar todos os pré-requisitos de *Laboratório de compiladores*, sendo necessário expandir a consulta para contemplar todos os níveis de transitividade do relacionamento difuso *temPreRequisito*. Assim, depois de analisar as condições da consulta original, o mapeamento e a ontologia difusa para identificar que a consulta referencia relacionamentos transitivos, é necessário definir quantos níveis de transitividade existem para a disciplina *Laboratório de compiladores* e o relacionamento *temPreRequisito*. Como o usuário não conhece previamente quais os níveis de transitividade possíveis, o sistema FOQuE verifica esta informação na ontologia difusa de domínio, para então expandir a consulta de forma transparente. A Figura 5.13 mostra quais os requisitos da disciplina *Laboratório de compiladores*, além dos respectivos níveis de transitividade.

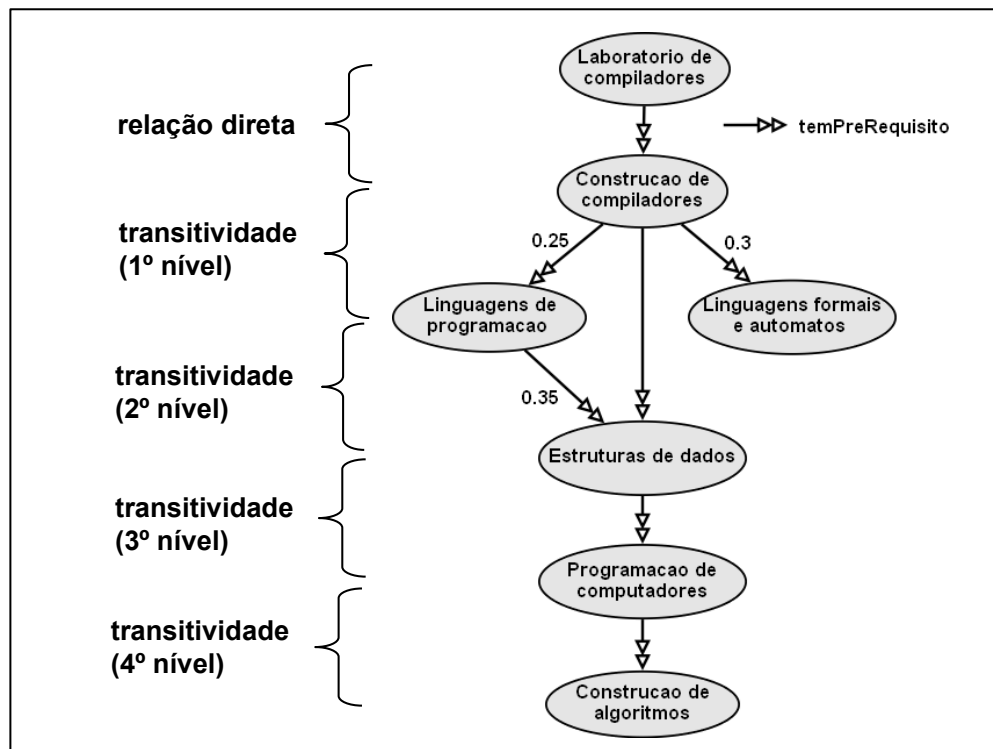


Figura 5.13 – Níveis de transividade para a disciplina *Laboratório de compiladores* e o relacionamento *temPreRequisito*.

Segundo a Figura 5.13, a disciplina *Laboratório de compiladores* possui no máximo quatro níveis de transividade para o relacionamento *temPreRequisito*, além da relação direta com o pré-requisito *Construção de compiladores*. Portanto, a consulta expandida deve atingir até o quarto nível de transividade, de modo que todos os requisitos da disciplina sejam devidamente recuperados. Assim, o sistema FOQuE baseia-se na abordagem apresentada na Seção 4.6.5 para obter a consulta expandida da Figura 5.14, com  $minMembership = 0.3$ ,  $minSimilarity = 1.0$  e  $minCloseness = 1.0$ .

```

select R.coddiscip_discip, R.nome_discip, R.inst_discip
from ingres_disciplinagrad D, ingres_disciplinagrad R, ingres_requisitograd Rel,
ingres_requisitograd Rel_1, ingres_requisitograd Rel_1_2,
ingres_requisitograd Rel_1_2_3, ingres_requisitograd Rel_1_2_3_4
where (((D.nome_discip = 'LABORATORIO DE COMPILADORES')
AND (R.nome_discip <> 'LINGUAGENS DE PROGRAMACAO'))
AND (D.coddiscip_discip = Rel.coddiscip_requisito)
AND ((Rel.codrequisito_requisito = Rel_1.coddiscip_requisito)
AND ((Rel_1.codrequisito_requisito = Rel_1_2.coddiscip_requisito)
AND ((Rel_1_2.codrequisito_requisito = Rel_1_2_3.coddiscip_requisito)
AND ((Rel_1_2_3.codrequisito_requisito = Rel_1_2_3_4.coddiscip_requisito)
AND (Rel_1_2_3_4.codrequisito_requisito = R.coddiscip_discip))))))

UNION

(select R.coddiscip_discip, R.nome_discip, R.inst_discip
from ingres_disciplinagrad D, ingres_disciplinagrad R, ingres_requisitograd Rel,
ingres_requisitograd Rel_1, ingres_requisitograd Rel_1_2, ingres_requisitograd Rel_1_2_3
where (((D.nome_discip = 'LABORATORIO DE COMPILADORES')
AND (R.nome_discip <> 'LINGUAGENS DE PROGRAMACAO'))
AND (D.coddiscip_discip = Rel.coddiscip_requisito)
AND ((Rel.codrequisito_requisito = Rel_1.coddiscip_requisito)
AND ((Rel_1.codrequisito_requisito = Rel_1_2.coddiscip_requisito)
AND ((Rel_1_2.codrequisito_requisito = Rel_1_2_3.coddiscip_requisito)
AND (Rel_1_2_3.codrequisito_requisito = R.coddiscip_discip))))))

UNION

(select R.coddiscip_discip, R.nome_discip, R.inst_discip
from ingres_disciplinagrad D, ingres_disciplinagrad R, ingres_requisitograd Rel,
ingres_requisitograd Rel_1, ingres_requisitograd Rel_1_2
where (((D.nome_discip = 'LABORATORIO DE COMPILADORES')
AND (R.nome_discip <> 'LINGUAGENS DE PROGRAMACAO'))
AND (D.coddiscip_discip = Rel.coddiscip_requisito)
AND ((Rel.codrequisito_requisito = Rel_1.coddiscip_requisito)
AND ((Rel_1.codrequisito_requisito = Rel_1_2.coddiscip_requisito)
AND (Rel_1_2.codrequisito_requisito = R.coddiscip_discip))))))

UNION

(select R.coddiscip_discip, R.nome_discip, R.inst_discip
from ingres_disciplinagrad D, ingres_disciplinagrad R, ingres_requisitograd Rel,
ingres_requisitograd Rel_1
where (((D.nome_discip = 'LABORATORIO DE COMPILADORES')
AND (R.nome_discip <> 'LINGUAGENS DE PROGRAMACAO'))
AND (D.coddiscip_discip = Rel.coddiscip_requisito)
AND ((Rel.codrequisito_requisito = Rel_1.coddiscip_requisito)
AND (Rel_1.codrequisito_requisito = R.coddiscip_discip))))

UNION

(select R.coddiscip_discip, R.nome_discip, R.inst_discip
from ingres_disciplinagrad D, ingres_disciplinagrad R, ingres_requisitograd Rel
where (((D.nome_discip = 'LABORATORIO DE COMPILADORES')
AND (R.nome_discip <> 'LINGUAGENS DE PROGRAMACAO'))
AND (D.coddiscip_discip = Rel.coddiscip_requisito)
AND (R.coddiscip_discip = Rel.codrequisito_requisito))))))

```

Figura 5.14 – Consulta expandida por transitividade para a disciplina *Laboratório de compiladores* e o relacionamento *temPreRequisito*.

Embora a consulta expandida seja bastante custosa em termos de junções e sub-consultas, ela possibilita recuperar todos os pré-requisitos da disciplina *Laboratório de compiladores* de forma transparente ao usuário, que não precisa saber previamente quantos



níveis de transitividade estão envolvidos na consulta. A cada sub-consulta, é possível recuperar todos os requisitos pertencentes a um determinado nível de transitividade. Por exemplo, a sub-consulta 1 recupera todos os requisitos do 4º nível, a sub-consulta 2 do 3º nível, e assim por diante até a sub-consulta 5, que corresponde à relação direta. Além disso, como  $\text{minMembership} = 0.3$ , o requisito *Linguagens de programação* não deve ser recuperado porque  $\mu_{\text{ehPreRequisitoDe}}(\text{Linguagens de programação}, \text{Laboratório de compiladores}) = 0.25$ , segundo as inferências de composição *max-min*. Portanto, adicionou-se a condição  $R.\text{nome\_discip} \langle \rangle \text{Linguagens de programação}$  às sub-consultas da consulta expandida. A Figura 5.15 apresenta os resultados da consulta expandida por transitividade.

original	membership	transitivity	similarity	closeness
<b>Original query results:</b>				
R.coddiscip_discip	R.nome_discip	R.inst_discip	Membership degree	
025119	CONSTRUCAO DE COMPILADORES	UFSCAR	1.0	
original	membership	transitivity	similarity	closeness
<b>Result augmented by transitivity to LABORATORIO DE COMPILADORES</b>				
Transitive relationship: pré-requisito				
R.coddiscip_discip	R.nome_discip	R.inst_discip	Membership degree	
025020	PROGRAMACAO DE COMPUTADORES	UFSCAR	1.0	
025054	CONSTRUCAO DE ALGORITMOS	UFSCAR	1.0	
025208	ESTRUTURAS DE DADOS	UFSCAR	1.0	
020265	LINGUAGENS FORMAIS E AUTOMATOS	UFSCAR	0.3	

**Figura 5.15 – Pré-requisitos da disciplina *Laboratório de compiladores*.**

Os resultados da Figura 5.15 mostram que a consulta original recupera apenas o pré-requisito referente à relação direta, sendo que os demais requisitos somente foram recuperados em função da expansão por transitividade efetuada pelo sistema FOQuE. As respostas adicionais são classificadas em ordem decrescente de pertinência associada ao relacionamento transitivo, permitindo, assim, que o usuário identifique quais as respostas mais relevantes. Por fim, pode-se observar que a expansão por transitividade também proporciona benefícios com relação à revocação, sem comprometer a precisão, visto que todos os resultados adicionais são semanticamente relevantes, de acordo com a pertinência mínima estabelecida.

#### 5.3.4.6 Consultas envolvendo vários tipos de expansão

Depois de ilustrar consultas para cada tipo de expansão realizada pelo sistema FOQuE, foram elaboradas consultas envolvendo diversos tipos de expansão, a fim de recuperar informações interessantes com relação ao domínio de aplicação considerado nos testes. Nas consultas que referenciam nomes de alunos, considerou-se o nome fictício *João* para preservar a identidade dos alunos de computação da UFSCar.

Um exemplo de consulta interessante para o domínio de aplicação é a consulta por notas de alunos nos pré-requisitos de uma disciplina, que pode auxiliar na tomada de decisão para a formação de turmas. Assim, por exemplo, alunos que tiraram notas baixas nos pré-requisitos podem ser alocados para turmas diferentes daqueles que tiraram boas notas. Diante deste contexto, considere a consulta pelas notas do aluno *João* em todos pré-requisitos da disciplina *Construção de compiladores*, com parâmetros *minMembership* = 1.0, *minSimilarity* = 1.0 e *minCloseness* = 0.2. A Figura 5.16 mostra a consulta expandida pelo sistema FOQuE. Nesta consulta, as disciplinas *Laboratório de compiladores* e *Introdução à compilação* foram expandidas por proximidade à disciplina *Construção de compiladores* com base na análise dos tópicos de ementa e nos parâmetros especificados. Além disso, a consulta expandida é capaz de atingir quatro níveis de transitividade, visto que o caminho de transitividade mais longo corresponde ao caminho entre a disciplina *Laboratório de compiladores* e o pré-requisito *Construção de algoritmos* (Figura 5.13). Por fim, adicionaram-se condições do tipo *nome\_discip* <> '*nome\_disciplina*' para os pré-requisitos que não satisfazem ao parâmetro *minMembership* nos diversos níveis de transitividade analisados. Os resultados obtidos em função das expansões realizadas são apresentados na Figura 5.17.



original	membership	transitivity	similarity	closeness
Original query results:				
discip.nome_discip		discipreq.nome_discip	insc.nota_insc	Membership degree
CONSTRUCAO DE COMPILADORES		ESTRUTURAS DE DADOS	6.0	1.0
original	membership	transitivity	similarity	closeness
Result augmented by closeness to CONSTRUCAO DE COMPILADORES				
discip.nome_discip		discipreq.nome_discip	insc.nota_insc	Closeness de
LABORATORIO DE COMPILADORES		CONSTRUCAO DE COMPILADORES	6.5	0.22222222
LABORATORIO DE COMPILADORES		ESTRUTURAS DE DADOS	6.0	0.22222222
LABORATORIO DE COMPILADORES		PROGRAMACAO DE COMPUTADORES	7.3	0.22222222
LABORATORIO DE COMPILADORES		CONSTRUCAO DE ALGORITMOS	8.6	0.22222222
original	membership	transitivity	similarity	closeness
Result augmented by transitivity to CONSTRUCAO DE COMPILADORES				
Transitive relationship: pré-requisito				
discip.nome_discip		discipreq.nome_discip	insc.nota_insc	Membership
CONSTRUCAO DE COMPILADORES		PROGRAMACAO DE COMPUTADORES	7.3	1.0
CONSTRUCAO DE COMPILADORES		CONSTRUCAO DE ALGORITMOS	8.6	1.0
Result augmented by transitivity to LABORATORIO DE COMPILADORES				
Transitive relationship: pré-requisito				
discip.nome_discip		discipreq.nome_discip	insc.nota_insc	Membership c
LABORATORIO DE COMPILADORES		ESTRUTURAS DE DADOS	6.0	1.0
LABORATORIO DE COMPILADORES		PROGRAMACAO DE COMPUTADORES	7.3	1.0
LABORATORIO DE COMPILADORES		CONSTRUCAO DE ALGORITMOS	8.6	1.0

Figura 5.17 – Notas de um aluno nos pré-requisitos da disciplina *Construção de compiladores*.

Os resultados mostram que a consulta original retorna somente as notas dos pré-requisitos associados pela relação direta, enquanto a consulta expandida permite recuperar as notas de todos os pré-requisitos pertencentes aos diversos níveis de transitividade. Também são recuperados todos os requisitos da disciplina *Laboratório de compiladores*, expandida por proximidade, uma vez que a expansão por transitividade considera todos os conceitos previamente expandidos. Vale ressaltar, ainda, que todos os pré-requisitos recuperados satisfazem ao parâmetro *minMembership*.

Outra consulta interessante para o domínio é a consulta por notas de alunos nas diversas disciplinas pertencentes a uma determinada categoria. A partir dos resultados desta consulta, professores podem, por exemplo, selecionar alunos de iniciação científica que tenham um bom desempenho nas disciplinas relacionadas a uma determinada área de pesquisa. Com base neste contexto, considere a consulta pelas notas do aluno *João* em todas as disciplinas pertencentes à categoria *Arquitetura de computadores*, com parâmetros *minMembership = 1.0*, *minSimilarity = 1.0* e *minCloseness = 1.0*. Como o termo *Arquitetura de computadores* está associado a três conceitos da ontologia (classe *Arquitetura de*

computadores, disciplina *Arquitetura de computadores UFSCar* e *Arquitetura de computadores USP/ICMC*), é possível realizar a expansão por homônimos. Depois que o contexto referente à classe *Arquitetura de computadores* for escolhido, o sistema FOQuE pode efetuar a expansão de classes, ao adicionar condições OR para todas as instâncias da classe *Arquitetura de computadores*. A Figura 5.18 mostra a consulta expandida e a Figura 5.19 apresenta os resultados obtidos.

```

select DISCIPLINA.nome_discip, INSCRICAO.nota_insc
from ingres_alunograd ALUNO, ingres_inscricaoograd INSCRICAO,
     ingres_disciplinograd DISCIPLINA
where (((ALUNO.nomealu_alug = 'JOAO')
AND (ALUNO.codalu_alug = INSCRICAO.codalu_insc))
AND (INSCRICAO.coddiscip_insc = DISCIPLINA.coddiscip_discip))
AND ((DISCIPLINA.nome_discip = 'ARQUITETURA DE COMPUTADORES')
OR (((((((((((DISCIPLINA.nome_discip = 'LABORATORIO DE ORGANIZACAO BASICA DE COMPUTADORES')
OR (DISCIPLINA.nome_discip = 'LABORATORIO DE ARQUITETURA DE COMPUTADORES'))
OR (DISCIPLINA.nome_discip = 'LOGICA DIGITAL'))
OR (DISCIPLINA.nome_discip = 'LABORATORIO DE ELEMENTOS DE LOGICA DIGITAL'))
OR (DISCIPLINA.nome_discip = 'ELEMENTOS DE LOGICA DIGITAL 2'))
OR (DISCIPLINA.nome_discip = 'ORGANIZACAO BASICA DE COMPUTADORES'))
OR (DISCIPLINA.nome_discip = 'MICROPROCESSADORES E MICROCONTROLADORES'))
OR (DISCIPLINA.nome_discip = 'ELEMENTOS DE LOGICA DIGITAL 1'))
OR (DISCIPLINA.nome_discip = 'ORGANIZACAO DE COMPUTADORES DIGITAIS 1'))
OR (DISCIPLINA.nome_discip = 'ARQUITETURA DE COMPUTADORES UFSCAR'))
OR (DISCIPLINA.nome_discip = 'LABORATORIO DE LOGICA DIGITAL'))
OR (DISCIPLINA.nome_discip = 'ARQUITETURA AVANCADA DE COMPUTADORES'))
OR (DISCIPLINA.nome_discip = 'ARQUITETURA DE COMPUTADORES USP'))
OR (DISCIPLINA.nome_discip = 'ARQUITETURAS AVANCADAS DE COMPUTADORES')))))

```

Figura 5.18 – Consulta por notas de um aluno nas disciplinas da classe *Arquitetura de computadores*.

Original query results:		
DISCIPLINA.nome_discip	INSCRICAO.nota_insc	
ARQUITETURA DE COMPUTADORES	4.2	
ARQUITETURA DE COMPUTADORES	6.4	

Result augmented by membership to ARQUITETURA DE COMPUTADORES		
DISCIPLINA.nome_discip	INSCRICAO.nota_insc	Memb
LOGICA DIGITAL	5.4	1.0
LOGICA DIGITAL	6.0	1.0
ORGANIZACAO BASICA DE COMPUTADORES	6.0	1.0
LABORATORIO DE ARQUITETURA DE COMPUTADORES	8.0	1.0
MICROPROCESSADORES E MICROCONTROLADORES	3.0	1.0
MICROPROCESSADORES E MICROCONTROLADORES	6.6	1.0
LOGICA DIGITAL	4.0	1.0
LABORATORIO DE LOGICA DIGITAL	7.8	1.0
LABORATORIO DE ORGANIZACAO BASICA DE COMPUTADORES	8.5	1.0
LABORATORIO DE ORGANIZACAO BASICA DE COMPUTADORES	0.0	1.0

Figura 5.19 – Notas de um aluno nas disciplinas da classe *Arquitetura de computadores*.

Os resultados da Figura 5.19 ilustram que as modificações efetuadas sobre a consulta original possibilitam recuperar as notas de todas as disciplinas pertencentes à classe *Arquitetura de computadores* de acordo com ontologia difusa. Deste modo, enquanto a consulta original obtém somente as notas das disciplinas que possuem exatamente o nome

*Arquitetura de computadores*, a consulta expandida recupera todas as informações requisitadas considerando o contexto apropriado.

Por fim, foi elaborada uma consulta para checar se um aluno da UFSCar pode cursar uma determinada disciplina da USP São Carlos, ao verificar se ele já cursou todos os pré-requisitos correspondentes na UFSCar. Assim, é possível saber quais disciplinas os alunos de computação da UFSCar podem cursar na USP com base nas disciplinas em que eles já foram aprovados na UFSCar. Para que o sistema FOQuE expanda esta consulta, ela deve ser dividida em duas etapas: a primeira para recuperar todos os pré-requisitos de uma disciplina da USP, e a segunda para verificar se o aluno foi aprovado nas disciplinas da UFSCar que são semanticamente próximas aos pré-requisitos identificados. Esta divisão em etapas ocorre porque as expansões efetuadas pelo sistema FOQuE baseiam-se no termo especificado na consulta, ou seja, na seqüência de caracteres fornecida pelo usuário, sendo necessário especificar o nome de cada pré-requisito para obter as disciplinas por proximidade todo-parte. Assim, considere a consulta que verifica se o aluno *João* pode cursar a disciplina *Visualização computacional* da USP. A Figura 5.20 mostra a consulta expandida para obter todos os pré-requisitos desta disciplina, considerando a expansão por transitividade, e a Figura 5.21 apresenta seus resultados.

```

select REQUISITO.coddiscip_discip, REQUISITO.nome_discip, DISCIPLINA.nome_discip
from ingres_disciplinagrad REQUISITO, ingres_disciplinagrad DISCIPLINA,
     ingres_requisitograd REL, ingres_requisitograd REL_1,
     ingres_requisitograd REL_1_2
where (((DISCIPLINA.nome_discip = 'VISUALIZACAO COMPUTACIONAL')
AND (DISCIPLINA.coddiscip_discip = REL.coddiscip_requisito))
AND ((REL.codrequisito_requisito = REL_1.coddiscip_requisito)
AND ((REL_1.codrequisito_requisito = REL_1_2.coddiscip_requisito)
AND (REL_1_2.codrequisito_requisito = REQUISITO.coddiscip_discip))))
UNION
(select REQUISITO.coddiscip_discip, REQUISITO.nome_discip, DISCIPLINA.nome_discip
from ingres_disciplinagrad REQUISITO, ingres_disciplinagrad DISCIPLINA,
     ingres_requisitograd REL, ingres_requisitograd REL_1
where (((DISCIPLINA.nome_discip = 'VISUALIZACAO COMPUTACIONAL')
AND (DISCIPLINA.coddiscip_discip = REL.coddiscip_requisito))
AND ((REL.codrequisito_requisito = REL_1.coddiscip_requisito)
AND (REL_1.codrequisito_requisito = REQUISITO.coddiscip_discip))))
UNION
(select REQUISITO.coddiscip_discip, REQUISITO.nome_discip, DISCIPLINA.nome_discip
from ingres_disciplinagrad REQUISITO, ingres_disciplinagrad DISCIPLINA,
     ingres_requisitograd REL
where (((DISCIPLINA.nome_discip = 'VISUALIZACAO COMPUTACIONAL')
AND (DISCIPLINA.coddiscip_discip = REL.coddiscip_requisito))
AND (REQUISITO.coddiscip_discip = REL.codrequisito_requisito))))

```

Figura 5.20 – Consulta por pré-requisitos da disciplina *Visualização computacional*.

original	membership	transitivity	similarity	closeness
Original query results:				
REQUISITO.coddiscip_discip	REQUISITO.nome_discip	Membership degree		
SCE0201	COMPUTACAO GRAFICA	1.0		
SCE0213	PROGRAMACAO ORIENTADA A OBJETOS	1.0		
original	membership	transitivity	similarity	closeness
Result augmented by transitivity to VISUALIZACAO COMPUTACIONAL				
Transitive relationship: pré-requisito				
REQUISITO.coddiscip_discip	REQUISITO.nome_discip	Membership		
SCE0181	INTRODUCAO A CIENCIA DA COMPUTACAO 2	1.0		
SCE0182	ALGORITMOS E ESTRUTURAS DE DADOS 1	1.0		
SCE0183	ALGORITMOS E ESTRUTURAS DE DADOS 2	1.0		
SCE0180	INTRODUCAO A CIENCIA DA COMPUTACAO 1	1.0		

Figura 5.21 – Pré-requisitos da disciplina *Visualização computacional*.

A consulta expandida da Figura 5.20 considera dois níveis de transitividade, pois a ontologia difusa possui no máximo dois níveis de transitividade para a disciplina *Visualização computacional* e o relacionamento transitivo *temPreRequisito*. Assim, para cada pré-requisito recuperado (Figura 5.21), o sistema FOQuE deve realizar a expansão por proximidade todo-parte para verificar se o aluno *João* foi aprovado em disciplinas próximas oferecidas pela UFSCar. A Figura 5.22 mostra a consulta expandida para recuperar as notas do aluno *João* nas disciplinas da UFSCar que são próximas ao pré-requisito *Introdução a ciência da computação 1*, com parâmetros  $minMembership = 1.0$ ,  $minSimilarity = 1.0$  e  $minCloseness = 0.6$ .

```
select REQUISITO.coddiscip_discip, REQUISITO.nome_discip, INSCRICAO.nota_insc
from ingres_disciplinograd REQUISITO, ingres_alunograd ALUNO,
ingres_inscriaograd INSCRICAO, ingres_resultadoinscriaograd RESULTADO
where ALUNO.nomealu_alug = 'JOAO' and RESULTADO.descricao_result = 'Aprovado'
```

Degrees:

Similarity  Closeness  Membership

```
select REQUISITO.coddiscip_discip, REQUISITO.nome_discip, INSCRICAO.nota_insc
from ingres_disciplinograd REQUISITO, ingres_alunograd ALUNO,
ingres_inscriaograd INSCRICAO, ingres_resultadoinscriaograd RESULTADO
where (((((ALUNO.nomealu_alug = 'JOAO') AND (RESULTADO.descricao_result = 'Aprovado'))
AND (INSCRICAO.resultado_insc = RESULTADO.cod_result)) AND (ALUNO.codalu_alug = INSCRICAO.codalu_insc))
AND (INSCRICAO.coddiscip_insc = REQUISITO.coddiscip_discip))
AND ((REQUISITO.nome_discip = 'INTRODUCAO A CIENCIA DA COMPUTACAO 1')
```

original	membership	transitivity	similarity	closeness
Result augmented by closeness to INTRODUCAO A CIENCIA DA COMPUTACAO 1				
REQUISITO.coddiscip_discip	REQUISITO.nome_discip	INSCRICAO.nota_insc	Closeness degree	
025054	CONSTRUCAO DE ALGORITMOS	7.8	0.8235294	
025062	LABORATORIO DE PROGRAMACAO	9.3	0.6666667	

Figura 5.22 – Notas de um aluno em disciplinas próximas a *Introdução a ciência da computação 1*.

Portanto, pode-se recomendar a disciplina *Visualização computacional* da USP São Carlos para o aluno *João*, caso ele tenha sido aprovado em todas as disciplinas da UFSCar que são semanticamente próximas aos pré-requisitos dessa disciplina.

#### 5.4. DISCUSSÃO DOS RESULTADOS

Considerando o domínio escolhido, que descreve categorias, requisitos e ementas de disciplinas da área de computação, os testes apresentaram diversos exemplos de consultas expandidas pelo sistema FOQuE que permitiram obter respostas semanticamente próximas aos termos especificados pelo usuário. Assim, foi possível mostrar a aplicabilidade da abordagem de expansão de consultas do sistema FOQuE em um domínio de aplicação e ilustrar quais as contribuições proporcionadas para os resultados das consultas.

Em geral, observou-se que o sistema FOQuE obtém melhores resultados em termos de revocação quando comparados aos resultados das consultas originais, conforme ilustrado pelas consultas executadas durante os testes. Isto ocorre principalmente porque os resultados das consultas originais baseiam-se no casamento de palavras-chave, enquanto as respostas expandidas resultam da análise da semântica de domínio representada por ontologias difusas. Os testes também mostraram que, mesmo com o aumento da revocação, a precisão das respostas não é prejudicada, pois o sistema FOQuE recupera apenas os dados que satisfazem aos parâmetros de expansão definidos para as consultas. Portanto, os resultados obtidos a partir dos testes mostram que as expansões semânticas realizadas pelo sistema FOQuE possibilitam que sejam recuperadas respostas adicionais semanticamente relevantes para os requisitos do usuário, de acordo com os parâmetros de pertinência, similaridade e proximidade mínima definidos por ele.

Por fim, os testes mostraram que as expansões por proximidade todo-parte e por transitividade tiveram um papel importante com relação ao domínio de aplicação, pois possibilitaram a elaboração de consultas que recuperam informações interessantes tanto para professores quanto para alunos da área de computação. Um exemplo é a recomendação de disciplinas que alunos da UFSCar podem cursar na USP, com base na análise dos pré-requisitos transitivos das disciplinas e na proximidade todo-parte entre pré-requisitos, obtida a partir dos tópicos de ementa. A expansão de homônimos também é significativa para o domínio escolhido, visto que permitiu que fossem recuperadas disciplinas de uma



determinada categoria da área de computação, o que não seria possível com as consultas originais. Por outro lado, observaram-se algumas limitações, como o fato de que os graus de pertinência e de similaridade foram definidos manualmente, enquanto alguns sistemas existentes possuem formas automatizadas para defini-los. Além disso, os testes também ilustraram que alguns tipos de expansão geram consultas bastante complexas, como no caso da expansão por transitividade. Dependendo da quantidade de níveis de transitividade a serem atingidos, a consulta expandida pode conter muitas sub-consultas e junções, aumentando assim o custo da consulta e comprometendo seu desempenho. Desta forma, trabalhos futuros podem considerar estas limitações para melhorar as expansões semânticas realizadas pelo sistema FOQuE.

## 5.5. CONSIDERAÇÕES FINAIS

Neste capítulo foram apresentados os testes realizados para avaliar a aplicabilidade do sistema FOQuE, suas contribuições e limitações. Apresentou-se o protótipo implementado, utilizado para realizar diversas consultas a um banco de dados contendo informações sobre disciplinas e alunos de computação da UFSCar, além de dados das disciplinas de computação da USP São Carlos. As expansões semânticas realizadas foram detalhadas, comparando-se os resultados recuperados sem expansão com as respostas expandidas. Por fim, apresentou-se uma discussão dos pontos fortes e das limitações do sistema FOQuE, com base nos resultados obtidos nos testes.

No próximo capítulo, serão apresentadas as conclusões, onde serão descritos os resultados alcançados, as principais contribuições deste trabalho de mestrado e os trabalhos futuros.

## 6. CONCLUSÕES

### 6.1. RESULTADOS ALCANÇADOS

A partir do sistema FOQuE, foi possível realizar diversos tipos de expansão semântica de consultas baseando-se em informações imprecisas representadas por ontologias difusas, a fim de obter resultados tanto relevantes quanto abrangentes. Novos tipos de expansão foram apresentados, como a expansão por proximidade todo-parte e por transitividade, contribuindo para a recuperação de respostas semanticamente relevantes que não seriam obtidas pelas consultas originais. As expansões de homônimos, classes e similaridade também possibilitaram melhorar os resultados das consultas, sempre de acordo com os parâmetros de expansão especificados pelo usuário. O sistema FOQuE possui, ainda, o diferencial de classificar as respostas adicionais conforme o tipo de expansão efetuada e apresentá-las de modo que os resultados mais relevantes com relação ao grau difuso considerado na expansão sejam visualizados em primeiro lugar. Assim, o usuário é capaz de identificar quais as expansões realizadas e quais os dados mais significativos para a consulta.

Os testes realizados apresentaram diversos exemplos de consultas expandidas que permitiram demonstrar a aplicabilidade da abordagem de expansão de consultas do sistema FOQuE em um domínio de aplicação. Os testes revelaram que o sistema FOQuE obteve melhores resultados em termos de revocação quando comparados aos resultados das consultas originais, pois estes se baseiam no casamento de palavras-chave, enquanto as respostas expandidas resultam da análise da semântica de domínio representada por ontologias difusas. Os testes ainda mostraram que o aumento da revocação não comprometeu a precisão das respostas, visto que foram recuperados somente os dados que satisfazem aos parâmetros de expansão. Além disso, observou-se que as expansões semânticas possibilitaram a recuperação de informações interessantes ao domínio de aplicação utilizado nos testes como, por exemplo, a recomendação de disciplinas que alunos de computação da UFSCar podem cursar na USP São Carlos, com base nas expansões por transitividade dos pré-requisitos das disciplinas e na proximidade todo-parte entre disciplinas. Deste modo, os resultados obtidos a partir dos testes ilustraram que as expansões semânticas realizadas pelo sistema FOQuE possibilitaram recuperar respostas adicionais semanticamente relevantes aos

requisitos do usuário, de acordo com os parâmetros de pertinência, similaridade e proximidade mínima definidos por ele.

## 6.2. CONTRIBUIÇÕES

As principais contribuições deste trabalho foram:

- A criação de uma meta-ontologia difusa, que permite incorporar conceitos de lógica difusa a ontologias específicas de domínio, sem impedir o uso de máquinas de inferência existentes para as linguagens OWL Lite e OWL DL. Também foram implementados alguns mecanismos de raciocínio sobre conceitos e relacionamentos difusos com base na máquina de inferência (*reasoner*) do *framework Jena*, para possibilitar a inferência de graus difusos de acordo com as definições da lógica difusa;
- A definição da expansão de consulta por proximidade todo-parte, que permite obter conceitos semanticamente próximos entre si com base na análise das partes em comum que os constituem. Os cálculos para determinar os graus de proximidade levam em consideração relacionamentos difusos como a pertinência das partes para o conceito que representa o “todo” e a similaridade entre as partes;
- A definição da expansão de consulta por transitividade, que possibilita recuperar conceitos que estejam associados entre si de forma indireta por relacionamentos transitivos não explícitos no banco de dados. Como os relacionamentos transitivos podem ser difusos, este tipo de expansão considera graus de pertinência, diferentemente da expansão por transitividade baseada na lógica clássica;
- A construção de um protótipo do sistema FOQuE, que expande consultas SQL a bancos de dados relacionais de forma não intrusiva: basta especificar a ontologia difusa que descreve a semântica dos dados e o mapeamento entre os elementos da ontologia e do banco de dados para que as consultas sejam semanticamente expandidas. Além disso, aplicações Java em geral podem acessar as funcionalidades do sistema FOQuE, pois estas foram implementadas como uma API de Java (API *FOQuE.jar*);

- A classificação dos resultados quanto ao tipo de expansão semântica, que permite identificar quais as expansões realizadas e os conceitos envolvidos, ordenados por relevância com relação ao grau difuso considerado na expansão (pertinência, similaridade ou proximidade todo-parte).

### 6.3. TRABALHOS FUTUROS

Pretende-se dar continuidade ao trabalho desenvolvido considerando os seguintes aspectos: testes, expansão de consulta, análise da consulta e ontologia difusa.

#### 6.3.1. Testes

Pretende-se avaliar o desempenho do sistema FOQuE, uma vez que este tipo de avaliação não foi realizada durante os testes apresentados. Devem ser realizados, também, baterias de testes envolvendo outros domínios de aplicação mais complexos, contendo diversos conceitos e relacionamentos difusos, com o intuito de validar a abordagem de expansão de consultas do sistema FOQuE. Os testes podem considerar não somente bancos de dados convencionais, mas também sistemas de integração de dados, para analisar possíveis extensões do sistema FOQuE para expandir consultas a sistemas que acessam diversas fontes de dados.

#### 6.3.2. Expansão de consulta

Com relação às expansões de consulta, pretende-se melhorar a abordagem para expandir consultas por transitividade, com o intuito de reduzir a complexidade das consultas expandidas. Pode-se, por exemplo, adicionar condições do tipo *atributo <> literal* somente para as sub-consultas correspondentes ao nível de transitividade ao qual pertence o conceito que representa o termo *literal*. É possível, ainda, optar por explicitar no banco de dados os relacionamentos com muitos níveis de transitividade, para evitar a geração de muitas sub-consultas e junções que podem comprometer o desempenho das consultas expandidas. Além disso, é possível estender a expansão por transitividade para outros tipos de relacionamento, além daqueles que correspondem a auto-relacionamentos no banco de dados.

Outro tipo de expansão que pode ser melhorada é a expansão por homônimos, para recomendar ao usuário qual o termo homônimo mais apropriado, sem que seja necessário consultá-lo para escolher o contexto adequado. Os trabalhos de Khan, McLeod & Hovy [KHAN; MCLEOD; HOVY, 2004] e de Parry [PARRY, 2004a, 2004b] caminham neste sentido e podem ser utilizados como base para desenvolver uma estratégia que possibilite ao sistema FOQuE identificar automaticamente qual o contexto desejado pelo usuário em consultas ambíguas.

Além de melhorar os tipos de expansão semântica já realizados pelo sistema FOQuE, pretende-se também incorporar outros como, por exemplo, a expansão de consultas com base em regras específicas de domínio, permitindo que relacionamentos específicos de domínio sejam analisados para inferir informações e reformular as consultas.

### **6.3.3. Análise da consulta**

Pretende-se estender o módulo de análise da consulta para que sejam interpretadas mais construções da linguagem SQL além das condições *atributo = atributo* e *atributo = literal*. Assim, podem ser consideradas funções de agregação, consultas aninhadas, operadores de junção (OUTER JOIN), entre outras construções.

### **6.3.4. Ontologia difusa**

Pretende-se incorporar a representação de funções de pertinência e modificadores na meta-ontologia difusa, além de considerar mais definições da lógica difusa como, por exemplo, a união, intersecção e o complemento de classes difusas. Além disso, outro ponto a ser pesquisado se refere aos graus de pertinência em hierarquias de classes e de relacionamentos difusos.

Como os graus de conceitos e relacionamentos difusos da ontologia utilizada pelo sistema FOQuE são especificados manualmente, trabalhos futuros podem utilizar técnicas automáticas ou semi-automáticas para obter graus de pertinência e de similaridade. Pode-se considerar, por exemplo, o *framework* FOGA [THO; HUI; CAO, 2004] [THO *et al.*, 2006], além de algumas abordagens citadas na Seção 3.4, como base para desenvolver uma

estratégia para definir (semi-)automaticamente os graus difusos das ontologias utilizadas pelo sistema FOQuE.

Por fim, pretende-se aplicar e avaliar a meta-ontologia difusa e os mecanismos de inferência sobre conceitos e relacionamentos difusos desenvolvidos neste trabalho em outros contextos, como a mineração de regras de associação difusas realizada pelo algoritmo *Extended Semantically Similar Data Miner* [ESCOVAR; YAGUINUMA; BIAJIZ, 2006].

## 7. REFERÊNCIAS

ABULAISH, M.; DEY, L. Biological ontology enhancement with fuzzy relations: a text-mining framework. In: IEEE/WIC/ACM INTERNATIONAL CONFERENCE ON WEB INTELLIGENCE, 2005, Compiègne, France. **Proceedings...** Washington, DC: IEEE Computer Society, 2005. p. 379-385.

BERNERS-LEE, T.; HENDLER, J.; LASSILA, O. The Semantic Web. **Scientific American**, v. 284, n. 5, p. 34-43, 2001.

BIDAULT, A.; FROIDEVAUX, C.; SAFAR, B. Repairing Queries in a Mediator Approach. In: EUROPEAN CONFERENCE ON ARTIFICIAL INTELLIGENCE (ECAI), 14., 2000, Berlin, Germany. **Proceedings...** Amsterdam: IOS Press, 2000. p. 406-410.

BUCHE, P. et al. Fuzzy querying of incomplete, imprecise, and heterogeneously structured data in the relational model using ontologies and rules. **IEEE Transactions on Fuzzy Systems**, v. 13, n. 3, p. 373-383, 2005.

CALVANESE, D.; DE GIACOMO, G. Data Integration: A Logic-Based Perspective. **AI Magazine**, v. 26, n. 1, p. 59-70, 2005.

CARROLL, J. J. et al. Jena: implementing the semantic web recommendations. In: INTERNATIONAL WORLD WIDE WEB CONFERENCE, 13., 2004, New York, NY, USA. **Proceedings...** New York: ACM Press, 2004. p. 74-83.

CHEN, G. **Fuzzy Logic in Data Modeling**. Norwell: Kluwer Academic Publishers, 1998.

CHEN, G.; WEI, Q. Fuzzy association rules and the extended mining algorithms. **Information Sciences**, v. 147, n. 1-4, p. 201-228, 2002.

CORBY, O. et al. Searching the Semantic Web: Approximate Query Processing Based on Ontologies. **IEEE Intelligent Systems**, v. 21, n. 1, p. 20-27, 2006.

DUBOIS, D.; PRADE, H. **Fuzzy Sets and Systems: Theory and Applications**. 1st. ed. New York: Academic Press, 1980.

ESCOVAR, E. L. G.; YAGUINUMA, C. A.; BIAJIZ, M. Using Fuzzy Ontologies to Extend Semantically Similar Data Mining. In: SIMPÓSIO BRASILEIRO DE BANCO DE DADOS, 21., 2006, Florianópolis, SC, Brasil. **Anais...** Florianópolis: UFSC, 2006. p. 16-30.

FIKES, R.; HAYESB, P.; HORROCKS, I. OWL-QL - a language for deductive query answering on the Semantic Web. **Web Semantics: Science, Services and Agents on the World Wide Web**, v. 2, n. 1, p. 19-29, 2004.

GAO, M.; LIU, C. Extending OWL by Fuzzy Description Logic. In: IEEE INTERNATIONAL CONFERENCE ON TOOLS WITH ARTIFICIAL INTELLIGENCE (ICTAI), 17., 2005, Hong Kong. **Proceedings...** Washington: IEEE Computer Society, 2005. p. 562-567.

GRUBER, T. R. A Translation Approach to Portable Ontology Specifications. **Knowledge Acquisition**, v. 5, n. 2, p. 199-220, 1993.

GUARINO, N. Formal Ontology and Information Systems. In: INTERNATIONAL CONFERENCE ON FORMAL ONTOLOGIES IN INFORMATION SYSTEMS, 1998, Trento, Italy. **Proceedings...** Amsterdam: IOS Press, 1998. p. 3-15.

GUARINO, N.; MASOLO, C.; VETERE, G. OntoSeek: content-based access to the Web. **IEEE Intelligent Systems**, v. 14, n. 3, p. 70-80, 1999.

HORROCKS, I.; PATEL-SCHNEIDER, P. F. Reducing OWL Entailment to Description Logic Satisfiability. In: INTERNATIONAL SEMANTIC WEB CONFERENCE (ISWC), 2003, Sanibel – Captiva Islands, Florida, USA. **Proceedings...** Berlin/Heidelberg: Springer, 2003. p. 17-29.

KHAN, L.; MCLEOD, D.; HOVY, E. Retrieval effectiveness of an ontology-based model for information selection. **The VLDB Journal**, v. 13, n. 1, p. 71-85, 2004.

KLIR, G. J.; YUAN, B. **Fuzzy Sets and Fuzzy Logic - Theory and Applications**. Upper Saddle River: Prentice Hall PTR, 1995.

LEE, C.-S.; JIAN, Z.-W.; HUANG, L.-K. A Fuzzy Ontology and Its Application to News Summarization. **IEEE Transactions on Systems, Man, and Cybernetics Part B**, v. 35, n. 5, p. 859-880, 2005.

MATTOS, D.; MOURA, A. M. D. C.; CAVALCANTI, M. C. ROSA+: Um Repositório de Objetos de Aprendizagem com Suporte a Inferência e Regras. In: SIMPÓSIO BRASILEIRO DE BANCO DE DADOS, 2006, Florianópolis, SC. **Anais...** Florianópolis: UFSC, 2006. p. 295-309.

MINISTÉRIO DA EDUCAÇÃO (MEC). **Diretrizes Curriculares de Cursos da Área de Computação e Informática**. Brasília: MEC/SESu/CEEInf, 1999. Disponível em: <<http://www.inf.ufrgs.br/mec/ceeinf.diretrizes.html>>. Acesso em: 15 jan. 2007.

MIAN, P. G.; FALBO, R. A. Supporting Ontology Development with ODEd. **Journal of the Brazilian Computer Society**, v. 9, n. 2, p. 57-76, 2003.

NARDI, D.; BRACHMAN, R. J. An Introduction to Description Logics. In: Baader, F. et al. **The Description Logic Handbook**. New York: Cambridge University Press, 2003. p. 5-44

NECIB, C. B.; FREYTAG, J. C. Using Ontologies for Database Query Reformulation. In: EAST EUROPEAN CONFERENCE ON ADVANCES IN DATABASES AND INFORMATION SYSTEMS (ADBIS), 8., 2004, Budapest, Hungary. **Proceedings...** 2004. p. 1-19.

\_\_\_\_\_. Query Processing Using Ontologies. In: INTERNATIONAL CONFERENCE ON ADVANCED INFORMATION SYSTEMS ENGINEERING (CAISE), 17., 2005a, Porto, Portugal. **Proceedings...** 2005a. p. 167-186.



\_\_\_\_\_. Semantic query transformation using ontologies. In: INTERNATIONAL DATABASE ENGINEERING AND APPLICATION SYMPOSIUM (IDEAS), 9., 2005b, Montreal, Canada. **Proceedings...** 2005b. p. 187- 199.

NOY, N. F. e MCGUINNESS, D. L. **Ontology Development 101: A Guide to Creating Your First Ontology**. Stanford University. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880. 2001.

ORCHARD, R. Fuzzy Reasoning in Jess: The FuzzyJ Toolkit and Fuzzy Jess. In: INTERNATIONAL CONFERENCE ON ENTERPRISE INFORMATION SYSTEMS, 3., 2001, Setubal, Portugal. **Proceedings...** 2001. p. 533-542.

PARRY, D. Fuzzification of a standard ontology to encourage reuse. In: IEEE INTERNATIONAL CONFERENCE ON INFORMATION REUSE AND INTEGRATION (IRI), 2004a, Las Vegas, NV, USA. **Proceedings...** Piscataway: IEEE Systems, Man, and Cybernetics Society, 2004a. p. 582-587.

\_\_\_\_\_. A fuzzy ontology for medical document retrieval. In: AUSTRALASIAN WORKSHOP ON DATA MINING AND WEB INTELLIGENCE, 2004b, Dunedin, New Zealand. **Proceedings...** Darlinghurst: Australian Computer Society, 2004b. p. 121-126.

PEAT, H. J.; WILLETT, P. The limitations of term co-occurrence data for query expansion in document retrieval systems. **Journal of the American Society for Information Science**, v. 42, n. 5, p. 378 - 383, 1999.

SMITH, M. K.; WELTY, C.; MCGUINNESS, D. L. **W3C Proposed Recommendation: OWL Web Ontology Language Guide**. Disponível em: <<http://www.w3.org/TR/2004/REC-owl-guide-20040210>>. Acesso em: 5 jan. 2006.

STOILLOS, G. et al. Uncertainty and the Semantic Web. **IEEE Intelligent Systems**, v. 21, n. 5, p. 84-87, 2006.

\_\_\_\_\_. The Fuzzy Description Logic f-SHIN. In: INTERNATIONAL WORKSHOP ON UNCERTAINTY REASONING FOR THE SEMANTIC WEB, 2005, Galway, Ireland. **Proceedings...** 2005.

STOJANOVIC, N. On the query refinement in the ontology-based searching for information. **Information Systems**, v. 30, n. 7, p. 543-563, 2005.

STRACCIA, U. A Fuzzy Description Logic for the Semantic Web. In: Sanchez, E. **Fuzzy Logic and the Semantic Web**. Elsevier, 2006. p. 73-90

\_\_\_\_\_. **The fuzzy DL System**. Disponível em: <<http://gaia.isti.cnr.it/~straccia/software/fuzzyDL/fuzzyDL.html>>. Acesso em: 20 mar. 2007.

THO, Q. T.; HUI, S. C.; CAO, T. H. FOGA: A Fuzzy Ontology Generation Framework for Scholarly Semantic Web. In: ECML/PKDD WORKSHOP ON KNOWLEDGE DISCOVERY AND ONTOLOGIES, 2004, Pisa, Italy. **Proceedings...** 2004. p. 37-48.

THO, Q. T. et al. Automatic Fuzzy Ontology Generation for Semantic Web. **IEEE Transactions on Knowledge and Data Engineering** v. 18, n. 6, p. 842- 856, 2006.

UNIVERSIDADE FEDERAL DE SÃO CARLOS (UFSCAR). **Relatório de Avaliação do Curso de Bacharelado em Ciência da Computação**. São Carlos:UFSCar/Departamento de Computação, 2000. 164 p. Etapa de Auto-Avaliação.

USCHOLD, M.; GRÜNINGER, M. Ontologies and Semantics for Seamless Connectivity. **SIGMOD Record**, v. 33, n. 4, p. 58-64, 2004.

WACHE, H. et al. Ontology-based integration of information - a survey of existing approaches. In: IJCAI-01 WORKSHOP: ONTOLOGIES AND INFORMATION SHARING, 2001, Seattle, USA. **Proceedings...** 2001. p. 108-117.

WIDYANTORO, D. H.; YEN, J. A fuzzy ontology-based abstract search engine and its user studies. In: IEEE INTERNATIONAL CONFERENCE ON FUZZY SYSTEMS, 10., 2001, Melbourne, Vic., Australia. **Proceedings...** Piscataway: IEEE Computer Society, 2001. p. 1291-1294.

ZADEH, L. A. Fuzzy Sets. In: Yager, R. R. et al. **Fuzzy sets and applications: Selected Papers by L.A. Zadeh**. New York: Wiley-Interscience, 1987a. p. 29-44

\_\_\_\_\_. Similarity Relations and Fuzzy Orderings. In: Yager, R. R. et al. **Fuzzy Sets and Applications: Select Papers by L. A. Zadeh**. New York: Wiley-Interscience, 1987b. p. 81-104