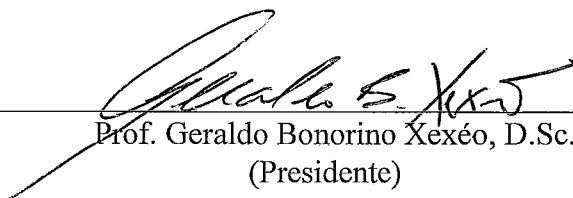


FERRAMENTAS DE MANIPULAÇÃO NEBULOSA DE DADOS COM
APLICAÇÃO EM SISTEMAS DE INFORMAÇÃO GEOGRÁFICA

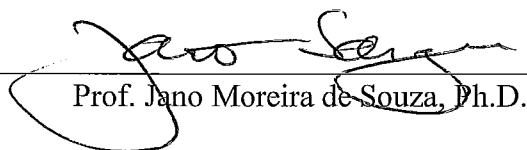
André Luiz Braga

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS
PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE
FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS
NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM
ENGENHARIA DE SISTEMAS E COMPUTAÇÃO .

Aprovada por:



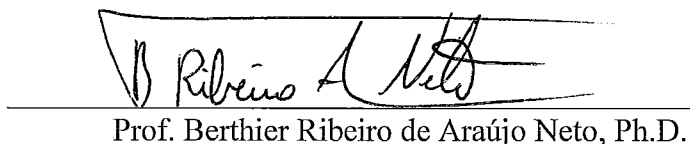
Prof. Geraldo Bonorino Xexéo, D.Sc.
(Presidente)



Prof. Jano Moreira de Souza, Ph.D.



Prof. Flávio Fonseca Nobre, Ph.D.



Prof. Berthier Ribeiro de Araújo Neto, Ph.D.

RIO DE JANEIRO, RJ - BRASIL

MAIO DE 1998

BRAGA, ANDRÉ LUIZ

Ferramentas de Manipulação Nebulosa de
Dados com Aplicação em Sistemas de
Informação Geográfica [Rio de Janeiro] 1998

X, 160 p. 29,7 cm (COPPE/UFRJ, M.Sc.,
Engenharia de Sistemas e Computação, 1998)

Tese - Universidade Federal do Rio de
Janeiro, COPPE

1. Sistemas Nebulosos
 2. Bancos de Dados Nebulosos
 3. Sistemas de Informação Geográfica
- I. COPPE/UFRJ II. Título (série)

DEDICATÓRIA

“A todos os pesquisadores, investigadores, estudiosos, inventores, observadores, curiosos, artistas, sonhadores, enfim todas as pessoas que procuram trazer à humanidade o conhecimento, seu crescimento e transformação.” (ALB e SAFA)

AGRADECIMENTOS

Ao Prof. Flávio Fonseca Nobre, pela sua orientação e suporte no desenvolvimento e aplicação deste trabalho.

A todos os membros da equipe de desenvolvimento do SIGEPI, em especial à Rejane Sobrino Pinheiro e João Alexandre Lopes.

A todos os membros do Programa de Engenharia Biomédica que colaboraram com idéias e discussões nas aplicações deste trabalho.

Ao Prof. Geraldo Xexéo pela sua orientação e incentivo de novas idéias e descobertas.

Ao Prof. Claudio Esperança pelas suas opiniões e auxílio.

À banca em especial ao prof. Berthier Ribeiro pelo sua atenção e interesse neste trabalho.

A CAPES pelo auxílio.

Ao CNPq pelo auxílio.

À secretária do Programa de Engenharia de Sistemas e Computação(PESC) em especial à Claudia.

A Patricia Leal secretária da linha de banco de dados pelo seu auxílio.

A Mercedes, bibliotecária do Centro de Documentação do PESC.

À minha noiva Soraia, pelo seu apoio e auxílio, principalmente na revisão do texto deste trabalho.

À minha mãe por todo o incentivo e apoio ao meu estudo e desenvolvimento ao longo de todos esses anos.

Ao meu pai, que postumamente sempre esteve me acompanhando e me incentivando.

A todos os meus amigos e colegas pela paciência e alegria proporcionada nos momentos de intensivo trabalho e estudo.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

FERRAMENTAS DE MANIPULAÇÃO NEBULOSA DE DADOS COM APLICAÇÃO
EM SISTEMAS DE INFORMAÇÃO GEOGRÁFICA

André Luiz Braga

Maio/1998

Orientador: Prof. Geraldo Bonorino Xexéo

Programa: Engenharia de Sistemas e Computação

Este trabalho implementa um sistema que utiliza a teoria nebulosa para manipulação de dados. Ele permite que os dados, em forma convencional, sejam convertidos em dados nebulosos através de um modelo definido pelo usuário do sistema. Isso é feito por uma linguagem de definição e consulta implementada como extensão do SQL. A definição do modelo permite ainda a criação de regras sobre variáveis do sistema. O sistema é aplicado em um SIG(Sistema de Informação Geográfica) onde verificamos sua capacidade de consulta e representação de dados espaciais. Propomos também um modelo capaz de manipular, de forma nebulosa, as incertezas ligadas às características geométricas dos dados espaciais.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

FUZZY MANIPULATION TOOLS OF DATA WITH APPLICATION IN
GEOGRAPHIC INFORMATION SYSTEMS

André Luiz Braga

May/1998

Advisor: Prof. Geraldo Bonorino Xexéo

Department: Computing and Systems Engineering

This work implements a system that uses the fuzzy theory for manipulation of data. He allows that the data, in conventional form, be transformed into fuzzy data through a model defined by the user of the system. That is made by a definition and query language implemented as extension of SQL. The definition of the model also allows the creation of rules over variables of the system. The system is applied in a GIS(Geographic Information System) where we verified its query capabilities and presentation of spatial data. We also propose a fuzzy model to manipulate, the uncertainties of the geometric characteristics of the spatial data.

ÍNDICE:

1. Introdução	2
1.1 Motivação	2
1.2 Objetivos	4
1.3 Organização do texto	6
2. Sistemas Nebulosos	8
2.1 Introdução	8
2.2 Conjuntos Nebulosos	8
2.2.1 Representação	10
2.2.2 Características	12
2.2.3 Operações	15
2.2.4 Variáveis Linguísticas	18
2.2.5 Operações de Agregação	20
2.2.6 Princípio da Extensão	22
2.2.7 Números Nebulosos	24
2.2.8 Aritmética Nebulosa	27
2.2.9 Comparações entre conjuntos nebulosos	28
2.3 Lógica Nebulosa	35
2.3.1 Proposições Lógicas Nebulosas	35
2.3.2 Regras e inferência nebulosa	38
2.4 Bancos de Dados	40
2.4.1 Modelo Relacional Nebuloso	40
2.4.2 Processamento de Consultas	45
3. Sistemas de Informação Geográfica	54
3.1 Funcionalidades	54
3.1.1 Representação	56
3.2 Modelos de Dados	56
3.2.1 Modelos	56
3.2.2 Paradigma Relacional	59
3.2.3 Paradigma Orientado para Objetos	60
3.3 Estruturas de dados	60
3.3.1 Tesselação Espacial	61
3.3.2 Formato Vetorial	61
3.4 Processamento de Consultas	61
3.5 Observações	62
3.6 Ferramentas de Manipulação Nebulosa	63
3.6.1 Incertezas em Bancos de Dados Espaciais	64
3.6.2 Manipulação de Dados	65
3.6.3 Mapeamento espacial e geométrico	68
4. Um Sistema de Banco de Dados Nebuloso	77
4.1 Arquitetura	77
4.2 Esquema Nebuloso	81
4.2.1 Elementos Básicos	82
4.2.2 Regras	90
4.3 Sistema de Controle	94
4.3.1 Saída e representação	95

4.3.2	Processamento de Consultas	96
4.4	Utilização e expansões	101
5.	<i>Manipulação Espacial Nebulosa</i>	104
5.1	Modelagem Espacial	104
5.1.1	Modelagem geométrica e extensional	104
5.1.2	Modelagem Semântica	106
5.1.3	Objetos Espaciais Nebulosos	107
5.1.4	Modelagem Semântica Nebulosa	111
5.2	Visão nebulosa de objetos espaciais precisos	113
5.2.1	Pontos Espaciais Nebulosos	115
5.2.2	Segmentos de Reta Nebulosos	115
5.2.3	Áreas de Contorno Nebulosas.	116
6.	<i>Linguagem de Definição e Consulta</i>	119
6.1	Definição do modelo	119
6.1.1	Definição de interfaces de dados	120
6.1.2	Definição de componentes	122
6.1.3	Definição conexão de componentes	124
6.1.4	Definição de regras.	126
6.2	Execução	129
6.3	Relatórios de erro	132
6.4	Estrutura	133
7.	<i>Aplicação: Dados de análise de Malária</i>	136
7.1	Malária	136
7.1.1	Ciclo de vida	137
7.1.2	Espécies	137
7.1.3	Ambiente de desenvolvimento	138
7.1.4	Tratamento e controle	138
7.2	Análise e representação de Dados Convencionais	139
7.2.1	Descrição dos dados	139
7.2.2	Modelagem	141
7.2.3	Descrição das análises	146
7.3	Conclusões	152
8.	Conclusões	154
9.	Referências Bibliográficas	158
10.	Gramática utilizada no Parser do sistema	167

Índice das Figuras:

Figura 2-1 - Conceito nítido "quente"	9
Figura 2-2 - Conceito nebuloso "quente"	10
Figura 2-3 - Exemplo de representação de um conjunto nebuloso	11
Figura 2-4 - Exemplo de suporte e supremo	12
Figura 2-5 - Exemplo de corte-alfa	13
Figura 2-6 - Exemplo de conjunto nebuloso não convexo	14
Figura 2-7 - Exemplo de variável lingüística	19
Figura 2-8 - Exemplo de número nebuloso	25
Figura 2-9 - Número nítido.....	25
Figura 2-10 - Intervalo de valores	26
Figura 2-11 - Intervalo nebuloso	26
Figura 2-12 - Número Nebuloso.....	27
Figura 2-13 - Abordagens de arquitetura nebulosa	46
Figura 2-14 - Arquitetura em visões.....	47
Figura 3-1 - Exemplo de tesselação.....	59
Figura 3-2 - Números nebulosos "Em torno de" e Noroeste	71
Figura 4-1 - Descrição geral do mapeamento de dados nebuloso.....	78
Figura 4-2 - Arquitetura geral do sistema.....	79
Figura 4-3 - Variável nebulosa	79
Figura 4-4 - Variável solução	80
Figura 4-5 - Variável nebulosa pura	80
Figura 4-6 - Diagrama de objetos do esquema nebuloso	82
Figura 4-7 - Objetos de interface de dados.....	83
Figura 4-8 - Objetos de variáveis lingüísticas	85
Figura 4-9 - Objetos de termos lingüísticos.....	88
Figura 4-10 - Objetos de modificadores.....	90
Figura 4-11 - Objetos de proposições.....	91
Figura 4-12 - Objetos de operadores	93
Figura 4-13 - Objetos de condições.....	93
Figura 4-14 - Estrutura genérica de uma proposição.....	94
Figura 4-15 - Objetos de regras	94
Figura 4-16 - Sistema de controle	95
Figura 5-1 - Domínio Espacial P	107
Figura 5-2 - Ponto nebuloso	109
Figura 5-3 - Representação de objetos nebulosos	113
Figura 5-4 - Segmento de reta nebuloso.....	116
Figura 7-1 - Distribuição espacial dos dados	139
Figura 7-2 - Diagrama conceitual de coleta de informação	140
Figura 7-3 - Mapa da área a ser pesquisada	143
Figura 7-4 - Número de casos de cada Plasmodium	147
Figura 7-5 - Representação cromática do numero de casos positivos.....	148
Figura 7-6 - Representação de áreas selecionadas	150
Figura 7-7 - Representação de visitas.....	151

Índice das Tabelas:

Tabela 2-1 - Resumo dos operadores de comparação.....	34
Tabela 2-2 - Relação Idade.....	43
Tabela 2-3 - Relações de similaridade.....	43
Tabela 2-4 - (R1)Relação projetada.....	50
Tabela 2-5 - Exemplo de classificação.....	51
Tabela 2-6 - Classificação pela álgebra F-G.....	52
Tabela 3-1 - Aspectos funcionais de um GIS.....	55
Tabela 3-2 - Classificação de dados geográficos.....	58
Tabela 3-3 - Alguns tipos de consultas feitas em SIGs.....	58
Tabela 6-1 - Comandos de criação de interfaces de dados.....	121
Tabela 6-2 - Exemplo de criação de tabelas.....	122
Tabela 6-3 - Comandos de criação de termos linguísticos.....	122
Tabela 6-4 - Comandos de criação de modificadores.....	123
Tabela 6-5 - Comandos de criação de variáveis linguísticas.....	123
Tabela 6-6 - Comandos de criação de variáveis solução.....	124
Tabela 6-7 - Comandos de conexão de componentes.....	125
Tabela 6-8 - Exemplo de criação de componentes nebulosos.....	125
Tabela 6-9 - Exemplo de conexão de componentes.....	126
Tabela 6-10 - Comandos de criação de proposições.....	127
Tabela 6-11 - Comandos de criação de condições.....	128
Tabela 6-12 - Comandos de criação de regras.....	128
Tabela 6-13 - Exemplo de criação de regras.....	129
Tabela 6-14 - Comandos de execução de consultas.....	130
Tabela 6-15 - Exemplo de execução de consultas de projeção.....	131
Tabela 6-16 - Exemplo de execução de consultas com projeção e seleção.....	132
Tabela 6-17 - Exemplo de execução de consultas com mínimo de relevância.....	132
Tabela 6-18 - Exemplo de relatório de execução dos comandos da linguagem.....	133
Tabela 6-19 - Exemplo de relatório de erros na criação do modelo nebuloso.....	133
Tabela 6-20 - Lista de palavras-chave da linguagem.....	134
Tabela 7-1 - Percentuais de casos positivos.....	149

Capítulo I

1. Introdução

As entidades do mundo real são geralmente descritas através de modelos que não levam em conta o grau de ambigüidade da lógica humana. Um grande número de técnicas visam melhor modelar as incertezas conceituais do mundo real. Este trabalho trata da utilização de uma destas técnicas, a teoria nebulosa, em bancos de dados, mostrando sua aplicação em Sistemas de Informação Geográfica.

1.1 Motivação

Nas teorias clássicas, um conceito pode ser verdadeiro ou falso, como por exemplo, um elemento pode pertencer ou não a um conjunto, uma proposição pode ser verdadeira ou falsa, etc. Na teoria nebulosa, um conceito pode assumir valores intermediários entre o verdadeiro e o falso. Os “graus de verdade” podem ser vistos, por exemplo, como *possibilidade* de um evento ocorrer, *pertinência* de um elemento a um conjunto ou a *relevância* de uma parte para o todo. Ela se divide em duas áreas principais de estudo: a teoria dos conjuntos nebulosos, como extensão da teoria clássica dos conjuntos, e a lógica nebulosa, também como extensão da lógica clássica.

A teoria dos conjuntos nebulosos fundamenta-se em um elemento poder possuir graus de pertinência a um determinado conjunto, resultando, em consequência, em uma extensão das operações clássicas entre conjuntos.

Na lógica nebulosa uma proposição do tipo: “João é Alto” possui graus de verdade ou relevância, dessa forma as operações e regras de inferência devem ser igualmente estendidas de forma a produzir deduções lógicas também com níveis de relevância.

Muitas áreas de aplicação têm sido alvo da teoria nebulosa, entre elas os sistemas de bancos de dados. Existem vários trabalhos teóricos a respeito, apesar de não haver um grande número de implementações. No modelo de dados nebuloso, os atributos das entidades de dados possuem valores conceituais associados a graus de relevância, como por exemplo, o atributo IDADE poderia assumir o valor “Jovem” com digamos, 30% de relevância, ou “Velho” com 80% de relevância.

A maioria dos trabalhos implementados ou sugeridos baseiam-se em bancos dados que já possuem valores nebulosos, contudo, a maioria dos bancos de dados são constituídos de dados convencionais ou exatos. A teoria nebulosa permite a utilização

de vários recursos interessantes como o estabelecimento de regras nebulosas entre variáveis. Essa opção é bastante utilizada em sistemas de controle mas pouco utilizada em propostas de bancos de dados nebulosos.

Os Sistemas de Informação Geográfica(SIGs) são um tipo de sistema de informação usado em todas as áreas onde se modelam, manipulam, analisam e representam dados do mundo real ao longo do espaço(Ramirez,1994). Eles são geralmente constituídos de uma base de dados convencional com dados associados a pontos ou regiões do espaço, denominada de base espacial.

Devido a grande necessidade de fidelidade entre o tratamento dos dados e a visão humana dos mesmos existem vários trabalhos aplicando a teoria nebulosa na representação e análise dos dados espaciais. Eles têm dois focos principais, um baseado no tratamento dos dados convencionais que são associados às entidades geográficas e outro na modelagem nebulosa dos dados geométricos. No primeiro caso, enfatiza-se a consulta e representação destes dados de forma nebulosa. Uma das muitas consultas feitas neste tipo de SIG é tentar saber quais as áreas do espaço que atendam a um certo predicado nebuloso, com por exemplo:

“Selecione todas as cidades onde a população é alta”

No outro enfoque, a modelagem nebulosa dos dados geométricos, os tipos de consulta nebulosa a dados espaciais são voltados às características espaciais dos dados, como por exemplo:

“Selecione as cidades perto de Itatiba”

Não existem muitos trabalhos, nos dois enfoques, voltados à manipulação nebulosa de dados já existentes sob formato convencional, ou seja preciso. Estas condições nos levaram a desenvolver um sistema de manipulação nebulosa de dados sobre um banco de dados convencional, possibilitando assim a modelagem, consulta e representação de dados de forma nebulosa porém sem interferir na estrutura original dos dados tratados. Escolhemos os SIGs para mostrar a contribuição deste tipo de tratamento, onde podemos usufruir das vantagens da representação nebulosa de dados, bem como explorar novas possibilidades de modelagem nebulosa através dos dados espaciais.

1.2 Objetivos

O especialistas de diversas áreas possuem conhecimentos de ordem conceitual sobre os dados do mundo real. Os dados coletados contudo, possuem em geral formato numericamente preciso¹ e torna-se difícil inferir estes conhecimentos diretamente sobre eles.

O objetivo deste trabalho foi desenvolver uma ferramenta capaz de manipular, consultar e fazer análises em bancos de dados precisos de forma nebulosa. Tal ferramenta possibilitará que especialistas possam criar e testar vários modelos de dados nebulosos dentro do seu próprio conceito da realidade.

O sistema é acoplado ao banco de dados, permitindo criar uma “visão” dos dados de forma nebulosa. Isto é feito através da modelagem das variáveis do banco e das relações entre elas. As entidades do mundo real são modeladas na forma de variáveis nebulosas, cujos valores podem assumir valores conceituais, ou lingüísticos. Uma variável “Temperatura”, por exemplo, poderia ser modelada como “Quente”, “Morna” e “Fria”.

O conhecimento dos especialistas sobre as relações entre as variáveis podem ser incorporadas ao modelo através de regras. Uma regra infere um valor nebuloso a uma variável de acordo com os valores de outras variáveis, como por exemplo:

“Se o número de portadores é alto o risco de contágio é elevado”

O sistema permite consultas ao banco de dados através do modelo nebuloso, onde os resultados em forma nebulosa, como os valores conceituais e suas relevâncias, serão enviados de volta para o banco de dados.

Para permitir uma modelagem adequada dos bancos de dados, representando os mais diversos tipos de realidades, propomos uma linguagem de definição e consulta nebulosa. Esta linguagem é baseada no SQL. Com isto pretendemos tornar simples a utilização do sistema como uma extensão do banco de dados ao qual ele está acoplado.

Dentre as várias propostas de linguagens para manipulação e consulta de dados nebulosos(Li,Liu,1990, Petry,1996, Petry e Cobb,1995, Shenoi et al,1990, Berthier e Muntz,1995), poucas tentam manter maior fidelidade ao padrão do SQL no que se refere a manipulação nebulosa dos dados. Isto se deve ao fato de que pesquisadores de várias

¹ Mesmo se conceitualmente imprecisos, isto é, com erros, os dados são guardados em uma forma simbólica precisa, em geral números.

áreas de conhecimento tentam adequar a linguagem aos seus jargões próprios. A linguagem implementada aqui é construída sobre um conjunto de comandos básicos.

O sistema foi incorporado ao Sistema de Informação Geográfica SIGEPI(Braga, Nobre, et al. 1994), onde os dados a serem modelados são provenientes de bases de dados espaciais, onde serão feitas consultas. Os dados resultantes das análises nebulosas são enviados novamente para o sistema, possibilitando mostrar as capacidades de representação e visualização espacial dos valores nebulosos.

Os dados espaciais, manipulados pelos SIGs, possuem características complexas que requerem um tratamento próprio. Pretendemos mostrar as possibilidades de extensão da manipulação nebulosa de dados deste tipo. Este recurso possibilita uma modelagem e visualização de conhecimento espacial sobre os dados, mostrando também uma outra dimensão de estudo de tratamento nebuloso de dados. Assim, podem ser feitas consultas de ordem conceitual aplicadas às características espaciais dos dados, como por exemplo:

“Selecione áreas de incidência de Malária perto de Itatiba”

Uma consulta deste tipo, com representação nebulosa no espaço, mostraria os graus de relevância da doença ao longo de toda a área, por meio de padrões de representação, como cores.

O sistema foi desenvolvido utilizando o paradigma da orientação por objetos e implementado como uma biblioteca de objetos agregados num esquema principal. A biblioteca de objetos permite que cada elemento definido do modelo seja compartilhado por vários outros objetos. Isto proporciona que um elemento nebuloso definido como “Alto”, por exemplo, tenha as mesmas características em todas as variáveis que o utilizam, mantendo assim a consistência do modelo.

O sistema pode ser usado independentemente, definindo a sua entrada de dados externa ou incorporado internamente a outros sistemas. Seguindo esta última forma, aproveitamo-nos da arquitetura aberta do Sistema de Informação Geográfica SIGEPI(Braga, Nobre, et al. 1994).

A área de saúde é constituída de especialistas cujo conhecimento está distante da manipulação de dados mais formal. Uma das linhas de pesquisa na área é a vigilância epidemiológica, onde os dados em sua maioria possuem referência e distribuição espacial. Pretendemos então mostrar a contribuição dessa ferramenta nesta

área, utilizando uma base de dados espacial de ocorrência de malária inserida dentro do SIGEPI, estendido com a biblioteca de manipulação nebulosa de dados implementada.

1.3 Organização do texto

Este trabalho está organizado em 7 capítulos, incluindo esta introdução. Os dois capítulos seguintes se destinam a uma revisão bibliográfica da teoria utilizada.

O capítulo 2 faz uma revisão em sistemas nebulosos e mostra as suas aplicações em bancos de dados.

O capítulo 3 faz uma revisão de Sistemas de Informação Geográfica e os trabalhos de aplicação de sistemas nebulosos.

O capítulo 4 se destina a descrição da biblioteca e sua implementação.

O capítulo 5 descreve uma extensão nebulosa de um modelo para entidades espaciais.

O capítulo 6 descreve a linguagem de definição e consulta implementada.

O capítulo 7 descreve a utilização do sistema numa aplicação prática usando dados de incidência de Malária.

O Capítulo 8 apresenta as conclusões e considerações finais do trabalho.

Capítulo II

2. Sistemas Nebulosos

2.1 Introdução

O conceito de sistemas nebulosos, mais precisamente a lógica nebulosa, foi introduzida por Zadeh(1965). Colocado inicialmente como uma extensão da lógica tradicional, ele se baseia em que uma proposição lógica não é necessariamente verdadeira ou falsa, mas possui, de fato, graus de verdade. Quando dizemos que o tempo está “ensolarado” não queremos dizer que não haja nuvens. Da mesma forma, em um tempo nublado podemos ter sol. Geralmente aceitamos como ensolarado um céu com até 20% ou 30% de nuvens(Klir,1995). Como podemos então definir os limites de nossas proposições ? Pode-se dizer que, quando o ser humano utiliza tais termos, ele infere um grau de verdade. Dentro de uma região tropical, por exemplo, pode-se aceitar nuvens em maior quantidade, visto que uma pouca quantidade de sol poderia ser suficiente para as atividades que dele dependem. Um meteorologista já estabeleceria, talvez limites mais rígidos e precisos. Os valores que definem o grau de verdade de uma proposição podem não ser conscientes, mas de alguma forma definidos pelo contexto social, lingüístico, ou uma base de referência experimental. A definição e a modelagem destes termos e valores constituem a base da teoria e do estudo dos sistemas nebulosos.

O estudo e a pesquisa nessa área possuem vários enfoques. Um dos mais explorados inicialmente em aplicações práticas foi o controle de processos (Cox,1994a,1994b). Os fundamentos teóricos se dividem em teoria dos conjuntos nebulosos e em lógica nebulosa. A teoria dos conjuntos nebulosos é uma extensão da teoria dos conjuntos, representando um enfoque mais amplo. Aqui estão incluídos os estudos de aritmética, programação, topologia, teoria de grafos e análise de dados. A lógica nebulosa tem por princípio estender a lógica clássica de forma a modelar e incluir a incerteza e a imprecisão do raciocínio humano[Zadeh,1965,1988].

2.2 Conjuntos Nebulosos

Para melhor compreendermos os conjuntos nebulosos devemos observar o conceito geral de conjuntos. Tomemos como exemplo a seguinte proposição: “a água está quente”. Modelamos esta proposição definindo seus parâmetros. Representamos a temperatura da água pela variável T , a qual chamamos de universo de discurso

(Cox,1994b, Klir,1995). Definimos então **A** como sendo o conjunto de elementos do *universo de discurso T* tal que a condição “quente” seja verdadeira. Este conjunto é chamado de *conjunto nítido* e pode ser definido como:

$$A \subseteq T, \text{ onde } A = \{x \mid x \in [50, 90]\}.$$

Esta é uma das formas de representar o *conjunto nítido*. Outra forma é definir uma função que retorne sim ou não para a certeza que tenhamos sobre um elemento pertencer ou não ao conjunto **A**. Ela é chamada de *função característica*. Uma função χ_A seria tal que,

$$\chi_A(x) = \begin{cases} 1 & \text{para } x \in A \\ 0 & \text{para } x \notin A \end{cases} \quad \text{ou mais formalmente, } \chi_A: X \rightarrow \{0, 1\}$$

Na Figura 2-1 podemos ver a representação do conceito nítido “quente”, onde caso a temperatura esteja entre 50 e 90 graus, indica que temos 100% de certeza de que a água está quente, caso contrário, não está.

Esta modelagem tem muitas aplicações e utilidades específicas, contudo para muitos casos, sofre de perda e distorção da informação. Neste caso específico, podemos ver que uma temperatura de 50°C é considerada quente e um valor imediatamente inferior de 49.9°C já é considerado fria.

Por outro lado, podemos verificar que dentro do raciocínio humano as afirmações tem “graus de certeza”. Quando medimos uma temperatura, sabemos que o termo “quente” está associado àquela temperatura através de um grau de relevância.

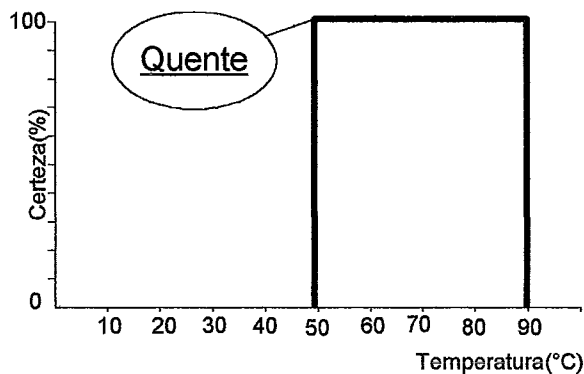


Figura 2-1 - Conceito nítido "quente"

Um *conjunto nebuloso* é então definido por um grupo de elementos de um universo de discurso X tais que, cada elemento pertence ao conjunto com um grau de pertinência (Klir, 1995). A função característica que associa cada elemento a um grau de relevância ou pertinência é chamada de *função de pertinência*. Esta função é normalizada assumindo valores reais no intervalo $[0,1]$ e formalmente descrita por:

$\mu_A : X \rightarrow [0,1]$, onde A é um conjunto nebuloso e X é um universo de discurso.

Tomando como base o exemplo acima, na Figura 2-2 podemos ver uma representação do conjunto nebuloso **QUENTE**. A função de pertinência μ_{QUENTE} associa o quanto a temperatura é considerada relevantemente “quente”. Podemos ter, por exemplo, as temperaturas 40, 50 e 60, com as pertinências 0.2, 0.4 e 0.6, respectivamente.

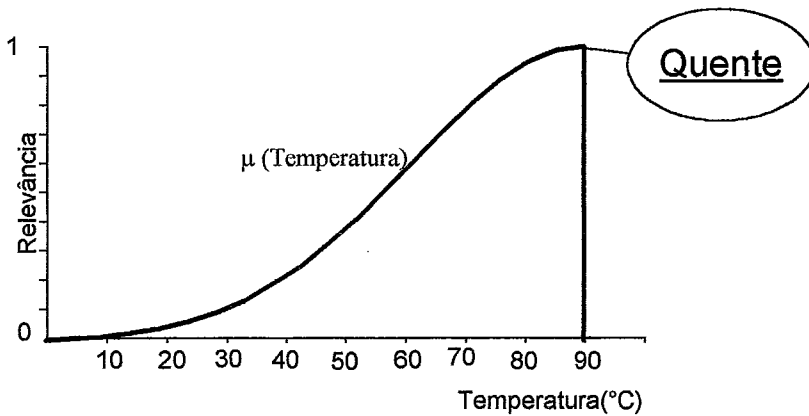


Figura 2-2 - Conceito nebuloso "quente"

2.2.1 Representação

Existem muitas formas de representar *conjuntos nebulosos* (Belchior, 1997). A primeira forma, mais geral, é considerar cada elemento do conjunto nebuloso A como um par ordenado $(x, \mu_A(x))$ onde x é um elemento nítido do universo de discurso X e $\mu_A(x)$ é valor da pertinência para o elemento x ou, mais formalmente,

$$A = \{(x, \mu_A(x)) \mid x \in X\} \quad (\text{Belchior, 1997})$$

Nesta forma de representação podemos ter conjuntos finitos:

$$A = \{(x_1, \mu_A(x_1)), (x_2, \mu_A(x_2)), \dots (x_n, \mu_A(x_n))\}$$

ou, no caso de um conjunto não finito,

$\mathbf{A} = \{(x, \mu_{\mathbf{A}}(x)) \mid \mu_{\mathbf{A}}(x) = f(x) \text{ e } x \in X\}$, onde $f(x)$ pode também ser uma função contínua.

Outra forma utilizada é definir diretamente \mathbf{A} como sendo a função de mapeamento do universo de discurso X no valor normalizado de pertinência (Klir, 1995). Nesta forma de representação, o conjunto nebuloso se confunde com a sua função de pertinência:

$$\mathbf{A} : X \rightarrow [0,1]$$

Temos por exemplo um conjunto nebuloso \mathbf{A} definido pelas curvas do gráfico da Figura 2-3:

$$\mathbf{A}(x) = \begin{cases} x - 1, & x \in [1,2[\\ 3 - x, & x \in [2,3] \\ 0, & \text{de outro modo} \end{cases}$$

Ainda outra forma utilizada é dispor os pares de valores nítidos e sua pertinência separados por barras (/)(Li, Liu, 1990):

$$\mathbf{A} = \mu_{\mathbf{A}}(x_1)/x_1 + \mu_{\mathbf{A}}(x_2)/x_2 + \dots + \mu_{\mathbf{A}}(x_n)/x_n$$

Se \mathbf{A} é um conjunto finito e contável, então podemos escrever:

$$\mathbf{A} = \sum_{i=1}^n \mu_{\mathbf{A}}(x_i) / x_i, \text{ por outro lado, se } \mathbf{A} \text{ é infinito, } \mathbf{A} = \int_x \mu_{\mathbf{A}}(x_i) / x_i$$

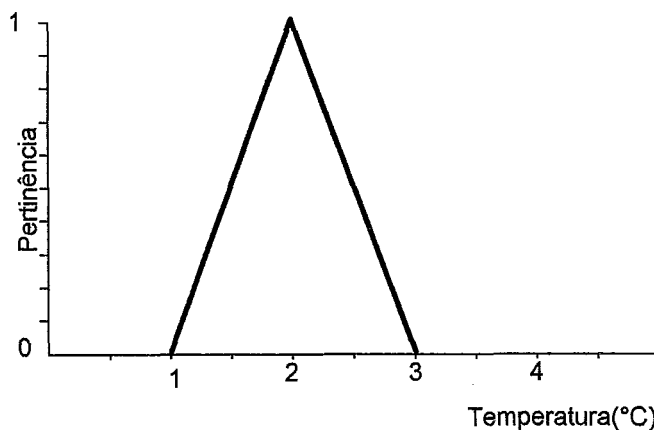


Figura 2-3 - Exemplo de representação de um conjunto nebuloso

2.2.2 Características

Os conjuntos nebulosos possuem várias propriedades básicas. Descreveremos aqui as mais fundamentais (Belchior,1997, Li,Liu,1990, Cox,1994a,1994b, Klir,1995).

O conjunto suporte, ou simplesmente suporte, corresponde ao intervalo pertencente ao universo de discurso X no qual a pertinência($\mu(x)$) seja maior do que zero. Como pode-se ver na Figura 2-4 o suporte de um conjunto nebuloso A é formalmente descrito como:

$$supp(A) = \{ x \in X \mid \mu_A(x) > 0 \}$$

O maior grau de pertinência que um conjunto nebuloso pode assumir é chamado de supremo ou altura. É formalmente definido como:

$$h(A) = \sup A(x) , x \in X$$

Quando o supremo de um conjunto nebuloso é igual a 1, o conjunto é dito *normal*.

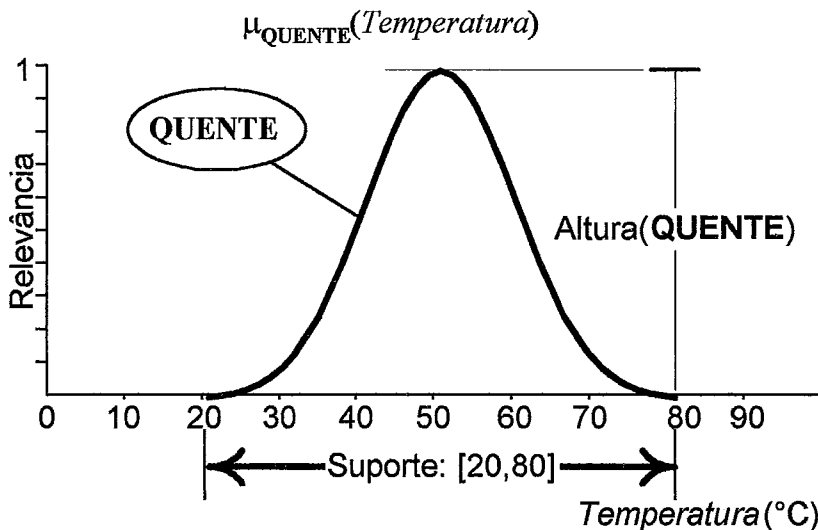


Figura 2-4 - Exemplo de suporte e supremo

Os *cortes-alfa* fazem parte de outro conceito importante dos conjuntos nebulosos. Por vezes torna-se necessário estabelecer limites de relevância mínima para os quais um conceito seja válido(Cox,1994b), valores normalizados que, quando associados a um conjunto nebuloso, modificam a sua pertinência. Qualquer valor abaixo do corte-alfa torna-se zero. Mais formalmente, uma vez aplicado um corte-alfa α num conjunto nebuloso A temos que,

$$A_\alpha = \begin{cases} \mu(x), & \text{se } \mu(x) \geq \alpha \\ 0, & \mu(x) < \alpha \end{cases}$$

Na Figura 2-5 podemos ver uma representação da aplicação de um corte-alfa em 0.5 num conjunto nebuloso **QUENTE**. As linhas tracejadas representam o conjunto nebuloso original. Uma das importantes propriedades dos cortes-alfa é a alteração do suporte(Cox,1994b). Na Figura 2-5 verificamos os dois suportes em contraste.

Várias formas de representação são adotadas. Em geral utilizam-se subíndices ou superíndices da forma A_α (Belchior,1997, Li,Liu,1990) ou ${}^\alpha A$ (Klir,1995)

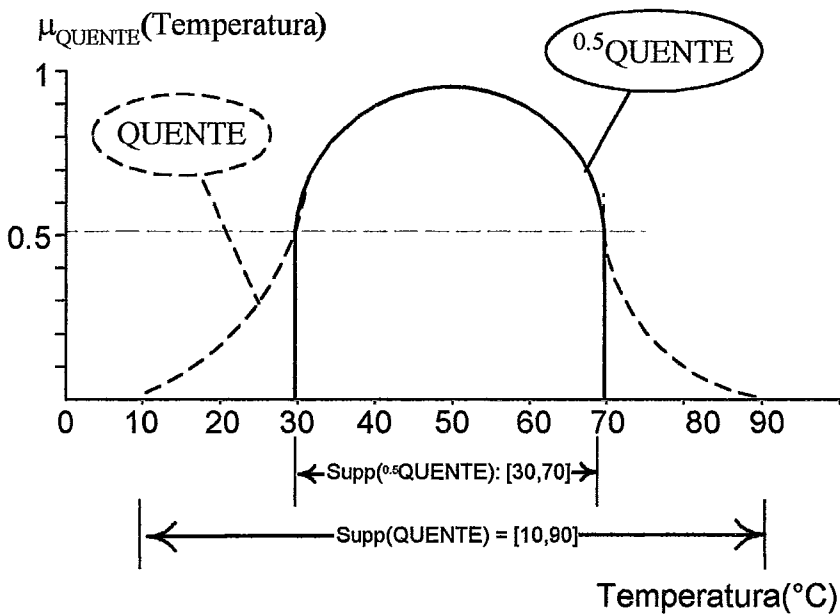


Figura 2-5 - Exemplo de corte-alfa

A convexidade é um importante conceito para várias questões de consistência matemática. Ele é generalizado dentro da teoria dos conjuntos nebulosos, fornecendo base para várias formulações matemáticas bem como interpretações conceituais(Klir,1995, Belchior,1997).

Para um conjunto nebuloso A_α dizemos que, dados $(x_1, x_2) \in X$, $\lambda \in [0,1]$ e $\alpha \in (0,1]$, A_α é convexo se e somente se, para todo corte α ,

$$\mu_A(\lambda x_1 + (1-\lambda)x_2) \geq \min[\mu_A(x_1), \mu_A(x_2)]$$

A extensão deste conceito para os conjuntos nebulosos é interpretada das seguinte forma: “A pertinência entre dois elementos de um conjunto nebuloso deve

ser maior ou igual que a menor pertinência entre estes dois elementos”. De um outro modo, podemos ver a convexidade como a inexistência de máximos locais na função de pertinência.

Uma das necessidades de convexidade pode ser vista de um ponto de vista conceitual. Em alguns casos queremos ter apenas um valor nítido para onde a relevância cresça. Na Figura 2-6 podemos ver que a interpretação do conjunto nebuloso “Jovem” parece ter duas idades mais relevantes distintas, onde entre elas, o indivíduo deixa de ser jovem, o que é uma incoerência.

A *cardinalidade* de um conjunto nítido é definida como o número de elementos pertencentes ao conjunto. No caso dos conjuntos nebulosos, cada elemento tem um grau de pertinência associado. Desta forma, a cardinalidade, dita escalar, de um conjunto nebuloso é dada pela soma das pertinência de seus elementos. Para um conjunto nebuloso finito A definido sobre X , a cardinalidade $|A|$ seria,

$$|A| = \sum_{x \in X} \mu_A(x), \text{ para um conjunto infinito, } |A| = \int_x \mu_A(x) dx$$

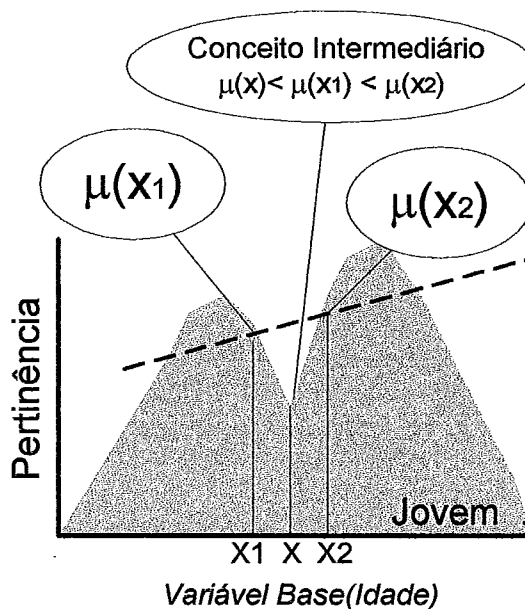


Figura 2-6 - Exemplo de conjunto nebuloso não convexo

A cardinalidade escalar é similar à área da curva de pertinência, contudo podemos definir uma cardinalidade relativa ao universo de discurso X . Isto nos proporciona comparar a cardinalidade de conjuntos nebulosos independentemente.

Utiliza-se a notação $\|A\|$ (Belchior,1997) para a cardinalidade relativa, onde $\|A\| = |A|/|X|$

2.2.3 Operações

A teoria dos conjuntos nítidos contém 3 operações básicas. São as operações de *complemento*, *interseção* e *união*. Ela é baseada nos conceito de pertinência, ou não, de um elemento aos conjuntos.

A extensão da teoria nebulosa destes operadores é disposta de forma a manter a sua consistência semântica(Belchior,1997, Li,Liu,1990, Cox,1994a,1994b, Klir,1995). Vários autores propõem conjuntos de funções para realizar estas operações. As operações de interseção e união possuem classes de funções denominadas de *t-norma* e *t-conorma*, respectivamente. Alguns axiomas devem ser impostos para estas funções(Klir,1995, Belchior,1997). Discutiremos aqui o conceito referente a cada um, especificando a sua forma para cada operação distinta. Usaremos a notação $\mathcal{I}(A(x),B(x))$ para definir genericamente uma *t-norma*, $\mathcal{U}(A(x),B(x))$ para uma *t-conorma*, e $\mathcal{C}(A(x))$ para o complemento genérico. Os termos $A(x)$ e $B(x)$ retornam o valor de pertinência para algum $x \in X$, onde X é um universo de discurso. Utilizamos a notação 1 e 0 para identificar conjuntos cuja pertinência seja sempre 1 ou 0 para todo x . Os axiomas de 1 a 5 são necessários para caracterização dos operadores. Os axiomas 6 e 7 são por vezes desejáveis.

1. Contorno - As restrições de contorno são relativas a manter a compatibilidade com a teoria dos conjuntos nítidos. Deve-se garantir que, quando um elemento x pertencer ou não a um conjunto A de forma plena, ou seja, com pertinência $\mu_A = 1$ ou $\mu_A = 0$, respectivamente, as operações devem obedecer às regras clássicas.

- **Axioma 1a** - $\mathcal{I}(A(x),1) = A(x)$ e $\mathcal{I}(A(x),0) = 0$. Para todo $x \in X$
- **Axioma 1b** - $\mathcal{U}(A(x),0) = A(x)$ e $\mathcal{U}(A(x),1) = 1$. Para todo $x \in X$
- **Axioma 1c** - $\mathcal{C}(0) = 1$ e $\mathcal{C}(1) = 0$.

2. Monotonicidade - Este principio mostra como a função deve se comportar quando os valores aplicados a ela são incrementados ou decrementados. Uma função deve,

por exemplo, ser sempre crescente com o crescimento dos seus valores de entrada ou sempre decrescente com o decrescimento dos valores de entrada.

Para o caso das *t-normas* e *t-conormas*, o axioma vem a partir do cálculo com intervalos. Ele indica que a união ou interseção não cresce acima de seu limite superior de intervalo. Os complementos, por outro lado, devem garantir o decrescimento da função com o crescimento de x .

- **Axioma 2a** - se $\mathbf{B}_1(x) \leq \mathbf{B}_2(x)$ implica que, $\mathcal{I}(\mathbf{A}(x), \mathbf{B}_1(x)) \leq \mathcal{I}(\mathbf{A}(x), \mathbf{B}_2(x))$. Para todo $x \in X$ e igualmente,
- **Axioma 2b** - se $\mathbf{B}_1(x) \leq \mathbf{B}_2(x)$ implica que, $\mathcal{U}(\mathbf{A}(x), \mathbf{B}_1(x)) \leq \mathcal{U}(\mathbf{A}(x), \mathbf{B}_2(x))$. Para todo $x \in X$

Existem alguns casos onde desejamos maior rigor na monotonicidade das funções, em função disto, alguns autores (Klir,1995, Belchior,1997) propõe uma monotonicidade estrita, considerando também o limite inferior do intervalo:

- **Axioma 2a'** - se $\mathbf{A}_1(x) \leq \mathbf{A}_2(x)$ e $\mathbf{B}_1(x) \leq \mathbf{B}_2(x)$ implica que, $\mathcal{I}(\mathbf{A}_1(x), \mathbf{B}_1(x)) \leq \mathcal{I}(\mathbf{A}_2(x), \mathbf{B}_2(x))$. Para todo $x \in X$ e igualmente,
 - **Axioma 2b'** - se $\mathbf{A}_1(x) \leq \mathbf{A}_2(x)$ e $\mathbf{B}_1(x) \leq \mathbf{B}_2(x)$ implica que, $\mathcal{U}(\mathbf{A}_1(x), \mathbf{B}_1(x)) \leq \mathcal{U}(\mathbf{A}_2(x), \mathbf{B}_2(x))$. Para todo $x \in X$.
 - **Axioma 2c** - se $\mathbf{A}(x) \leq \mathbf{B}(x)$ implica que, $\mathcal{C}(\mathbf{B}(x)) \leq \mathcal{C}(\mathbf{A}(x))$. Para todo $x \in X$.
- 3. Comutatividade** - Aplica-se somente às *t-normas* e *t-conormas*. Ela garante um principio básico de operação com conjuntos e a propriedade de simetria.
- **Axioma 3a** - $\mathcal{I}(\mathbf{A}(x), \mathbf{B}(x)) = \mathcal{I}(\mathbf{B}(x), \mathbf{A}(x))$, para todo $x \in X$.
 - **Axioma 3b** - $\mathcal{U}(\mathbf{A}(x), \mathbf{B}(x)) = \mathcal{U}(\mathbf{B}(x), \mathbf{A}(x))$, para todo $x \in X$.
- 4. Associatividade** - Aplica-se somente as *t-normas* e *t-conormas* e garantimos o principio básico dos conjuntos nítidos, ela estende a operação com os conjuntos nebulosos para mais de dois conjuntos.
- **Axioma 4a** - $\mathcal{I}(\mathcal{I}(\mathbf{A}(x), \mathbf{B}(x)), \mathbf{C}(x)) = \mathcal{I}(\mathbf{A}(x), \mathcal{I}(\mathbf{B}(x), \mathbf{C}(x)))$, para todo $x \in X$.

- **Axioma 4b** - $\mathcal{U}(\mathcal{U}(A(x), B(x)), C(x)) = \mathcal{U}(A(x), \mathcal{U}(B(x), C(x)))$, para todo $x \in X$.
- 5. **Continuidade** - Garante que pequenas variações de x não acarretarão em grandes variações da função.
- 6. **Involução** - Este princípio se aplica somente ao operador de complemento. Garante que uma vez aplicado o operador sobre si mesmo, retorna ao valor original.
- **Axioma 6a** - $\mathcal{C}(\mathcal{C}(A(x))) = A(x)$, para todo $x \in X$.
- 7. **Idempotência** - É um requisito relativo a limites quando a função é aplicada a valores iguais.
- **Axioma 6a** - $\mathcal{I}(A(x), A(x)) = A(x)$, para todo $x \in X$.
- **Axioma 6b** - $\mathcal{U}(A(x), A(x)) = A(x)$, para todo $x \in X$.

Os operadores possuem um grupo de funções padrão propostas por Zadeh(1965). Eles são baseados em operadores de máximo e mínimo e atendem a um grupo de restrições padrão, como são a única classe de funções idempotentes(Klir,1995). Alguns exemplos:

- **Padrão:**

$$\mathcal{I}(A(x), B(x)) = \min(\mu_A(x), \mu_B(x))$$

$$\mathcal{U}(A(x), B(x)) = \max(\mu_A(x), \mu_B(x))$$

$$\mathcal{C}(A(x)) = 1 - \mu_A(x)$$

- **Produto e Soma Algébrica**

$$\mathcal{I}(A(x), B(x)) = \mu_A(x) \cdot \mu_B(x)$$

$$\mathcal{U}(A(x), B(x)) = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x)$$

- **Diferença e Soma Limitada**

$$\mathcal{I}(A(x), B(x)) = \max(0, \mu_A(x) + \mu_B(x) - 1)$$

$$\mathcal{Z}(\mathbf{A}(x), \mathbf{B}(x)) = \min(1, \mu_{\mathbf{A}}(x) + \mu_{\mathbf{B}}(x))$$

- **Interseção e União Robusta:**

$$\mathcal{Z}(\mathbf{A}(x), \mathbf{B}(x)) = \begin{cases} \mu_{\mathbf{A}}(x) & \text{quando } \mu_{\mathbf{B}}(x) = 1 \\ \mu_{\mathbf{B}}(x) & \text{quando } \mu_{\mathbf{A}}(x) = 1 \\ 0, & \text{de outro modo} \end{cases}$$

$$\mathcal{Z}(\mathbf{A}(x), \mathbf{B}(x)) = \begin{cases} \mu_{\mathbf{A}}(x) & \text{quando } \mu_{\mathbf{B}}(x) = 0 \\ \mu_{\mathbf{B}}(x) & \text{quando } \mu_{\mathbf{A}}(x) = 0 \\ 1, & \text{de outro modo} \end{cases}$$

As propriedades básicas dos conjuntos são estendidas para a aplicação dos *cortes-alfa*, permitindo a ampliação das capacidades de manipulação dos conjuntos nebulosos.

2.2.4 Variáveis Lingüísticas

O conceito de *variável lingüística* é amplamente exposto(Cox,1994b, Zadeh, 1988) e mesmo formalmente definido(Klir,1995, Belchior,1997) por quase todos os autores. A variável lingüística é, em termos mais simples, uma variável cujo domínio, ou valores, são termos da linguagem referentes a um certo contexto(Zadeh, 1988). Podemos dizer, por exemplo, que a variável lingüística “Idade” pode assumir os valores “Jovem”, “Adulto” ou “Velho”.

Aplicando estes conceitos à teoria nebulosa, os valores(Jovem, Adulto, Velho) seriam nomes de conjuntos nebulosos, denominados *termos lingüísticos*, aos quais estariam associados a um universo de discurso *IDADE*(Figura 2-7). A variável nítida Idade, chamada também de variável base(Klir,1995, Belchior,1997), tem seus valores classificados pelos conjuntos nebulosos. Eles nos permitem melhor adaptar conceitos genéricos às variáveis lingüísticas.

Este processo de classificação é similar a trabalhar com a variável através de intervalos(Klir,1995). Contudo, a extensão fornecida pela manipulação nebulosa nos dá maior nível de informação. Uma das razões se deve à relevância em relação aos valores da variável base. A flexibilidade fornecida pela função de pertinência nos permite fazer operações com os intervalos mantendo uma maior adequação com os conceitos do mundo real.

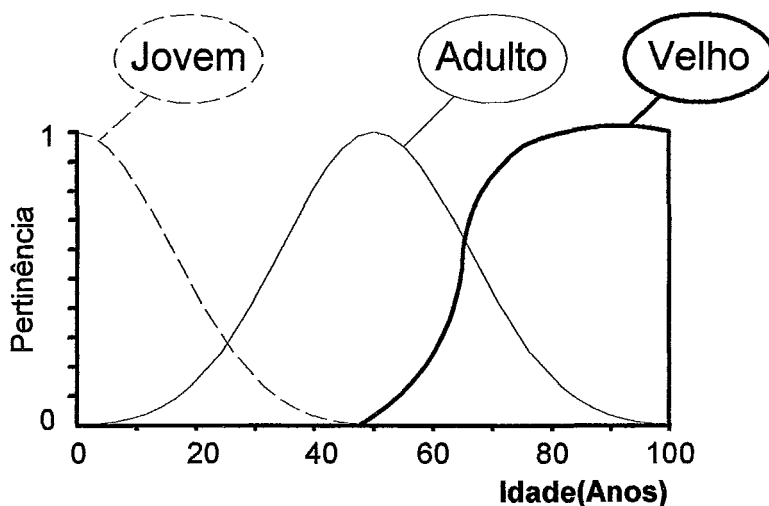


Figura 2-7 - Exemplo de variável lingüística

Para a construção de uma variável lingüística (Klir, 1995, Belchior, 1997) de nome V , definida sobre um universo de discurso X , devemos ter um conjunto de termos lingüísticos T com gramática de geração g e uma regra semântica m . A gramática g define a sintaxe dos termos lingüísticos. Ela pode ser constituída, simplesmente de uma lista de termos, como por exemplo,

$termo := Jovem \mid Adulto \mid Velho$

Ou então regras de formação mais complexas que envolvam modificadores ou operadores, do tipo,

$termo := Modificador termo \mid$

$Não termo \mid$

$Jovem \mid Adulto \mid Velho$

$Modificador := Muito \mid Pouco \mid Razoavelmente$

Estas regras permitem a formação sintática de termos lingüísticos mais complexos como “Muito Velho”, “Não Jovem”, etc.

A regra semântica m define como os termos lingüísticos estarão associados a seus conjuntos nebulosos ou funções de pertinência. Ela assinala o grupo de termos $t \in T$ a um grupo de conjuntos nebulosos $\mathbf{F}(X)$, definido sobre um universo de discurso X , da forma:

$$m: T \rightarrow \mathbf{F}(X)$$

Para um termo qualquer t associado a uma variável base u , teríamos,

$$m(t) = \{u, \mu_t(u) \mid u \in X\}$$

No exemplo da(Figura 2-7) teríamos as regras semânticas representadas pelas funções de pertinência μ_{JOVEM} , μ_{ADULTO} e μ_{VELHO} , as quais caracterizam os conjuntos nebulosos $\mathbf{F}(X)$.

Usa-se definir o conjunto de elementos definidos acima como uma quintupla (V, T, X, g, m) , a qual caracteriza a definição formal de uma variável lingüística(Klir,1995).

2.2.5 Operações de Agregação

Em muitos casos, torna-se necessário combinarmos dois conceitos ou termos lingüísticos diferentes que, de alguma forma, se aplicam parcial ou simultaneamente a um mesmo atributo. Para isto é necessário unificar estes conjuntos em um novo conjunto. Esta operação é chamada de *agregação*(Belchior,1997, Klir, 1995) e é definida por uma função de transferência h que combina n conjuntos nebulosos em um único conjunto:

$$h: [0,1]^n \rightarrow [0,1]$$

Um conjunto agregado de n conjuntos nebulosos $C = h(A_1, A_2, \dots, A_n)$, é definido por:

$$C = \{(x, \mu_C(x)) \mid \mu_C(x) = h(\mu_{A_1}(x), \mu_{A_2}(x), \dots, \mu_{A_n}(x))\}$$

Existem várias funções de transferência propostas na literatura(Belchior,1997), cada uma delas é mais ou menos adequada ao problema específico tratado. Uma função de agregação deve atender pelo menos 3 dos 5 axiomas(Klir, 1995):

- **Axioma 1 - Condições de contorno.** $h(0, 0, \dots, 0) = 0$ e $h(1, 1, \dots, 1) = 1$.
- **Axioma 2 - Crescimento monotônico.** Se para qualquer par de tuplas $\mu_{A_1}(x), \mu_{A_2}(x), \dots, \mu_{A_n}(x)$ e $\mu_{B_1}(x), \mu_{B_2}(x), \dots, \mu_{B_n}(x)$, tal que $i = 1, 2, \dots, n, \mu \in [0, 1]$, se $\mu_{A_i}(x) \leq \mu_{B_i}(x)$, então $h(\mu_{A_1}(x), \mu_{A_2}(x), \dots, \mu_{A_n}(x)) \leq h(\mu_{B_1}(x), \mu_{B_2}(x), \dots, \mu_{B_n}(x))$.
- **Axioma 3 - Continuidade.** A função de transferência h deve ser contínua, ou seja, variações infinitesimais dos argumentos não pode gerar grande variações na função.
- **Axioma 4 - Simetria.** Garante que para qualquer permutação p em N , $h(A_1, A_2, \dots, A_n) = h(A_{p(1)}, A_{p(2)}, \dots, A_{p(n)})$, desta forma mantemos a igualdade de importância entre os conjuntos nebulosos. Em alguns casos, quando isto não é desejado, este axioma pode ser relaxado.
- **Axioma 5 - idempotência.** $h(\mu_A(x), \mu_A(x), \dots, \mu_A(x)) = \mu_A(x)$. Este axioma é uma generalização do axioma 1.

As operações de união e interseção nebulosas são igualmente funções de agregação. Qualquer função h que atenda ao axioma 5 e ao axioma 2, responderá a inequação:

$$\min(\mu_{A_1}(x), \mu_{A_2}(x), \dots, \mu_{A_n}(x)) \leq h(\mu_{A_1}(x), \mu_{A_2}(x), \dots, \mu_{A_n}(x)) \leq \max(\mu_{A_1}(x), \mu_{A_2}(x), \dots, \mu_{A_n}(x))$$

Desta forma, qualquer função que esteja entre as funções de união e interseção, atende aos critérios axiomáticos. Uma classe de funções que atende o critério de idempotência é a de funções de média, cuja sua forma genérica é:

$$h(\mu_{A_1}(x), \mu_{A_2}(x), \dots, \mu_{A_n}(x)) = \left(\frac{\mu_{A_1}^\beta(x) + \mu_{A_2}^\beta(x) + \dots + \mu_{A_n}^\beta(x)}{n} \right)^{1/\beta}$$

Podemos ver que quando $\beta=1$, temos a média aritmética, quando $\beta=2$, temos o valor médio quadrático e para $\beta=-1$ temos a média harmônica.

Existe uma outra classe de funções permitindo um maior ajuste entre as funções de máximo e mínimo. São as OWA(Ordered Weighted Average), ou função de média ponderada, que é definida por:

$$h(\mu_{A_1}(x), \mu_{A_2}(x), \dots, \mu_{A_n}(x)) = w_1\mu_{B_1}(x) + w_2\mu_{B_2}(x) + \dots + w_n\mu_{B_n}(x)$$

Onde $\mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_n$ são a permutação de $\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_n$ de forma que $\mathbf{B}_i \geq \mathbf{B}_j$ para qualquer $j > i$ e $i, j = 1, 2, \dots, n$, ou seja os valores de pertinência são ordenados de forma decrescente. O vetor $\mathbf{w} = \langle w_1, w_2, \dots, w_n \rangle$ deve tal que,

$$w_i \in [0,1] \quad \text{e} \quad \sum_{i=1}^n w_i = 1.$$

A OWA pode representar a operação de mínimo, nesse caso $\mathbf{w} = \langle 0, 0, \dots, 1 \rangle$ ou para representar a operação de máximo, onde $\mathbf{w} = \langle 1, 0, \dots, 0 \rangle$. Para outros casos, OWA tem a possibilidade de realizar uma configuração flexível entre a operação máx e mín.

2.2.6 Princípio da Extensão

Os sistemas e modelos matemáticos estão baseados em relações que mapeam um conjunto de variáveis de um universo em outro. Estas relações são comumente representadas por funções f da forma:

$$f: X \rightarrow Y, \text{ onde,}$$

X e Y representam dois universos quaisquer. Seja x um elemento do universo X e $y = f(x)$ uma função que mapea x em $y \in Y$. A função característica que descreve a relação R representando y seria (Ross, 1995):

$$\chi_R(x, y) = \begin{cases} 1, & y = f(x) \\ 0, & y \neq f(x) \end{cases}$$

Devemos agora considerar que tenhamos subconjuntos de X que serão mapeados em subconjuntos de Y , dessa forma:

$$f: \mathbf{P}(X) \rightarrow \mathbf{P}(Y),$$

onde $\mathbf{P}(\cdot)$ representa o conjunto potência. Um subconjunto nebuloso $\mathbf{A} \subset X$ é mapeado num subconjunto $\mathbf{B} \subset Y$ da forma: $\mathbf{B} = f(\mathbf{A}) = \{y \mid \forall x \in \mathbf{A}, y = f(x)\}$. Em termos mais gerais, devemos supor que mais de um elemento de Y seja mapeado a partir de um mesmo elemento de X , dessa forma, a função característica de \mathbf{B} seria,

$$\chi_B(y) = \bigvee_{y=f(x)} \chi_A(x)$$

onde $\vee_{y=f(x)}$ representa o valor máximo para todos os valores de x que levem a y . A função característica em conjuntos precisos apenas indica se um valor pertence ou não a um conjunto. Zadeh(1975) elaborou uma extensão para $\chi_B(y)$ de forma a lidar com valores intermediários de pertinência:

$$\mu_B(y) = \vee_{y=f(x)} \mu_A(x) = \text{máx}_{y=f(x)} \mu_A(x) \text{ onde,}$$

$x \in \mathbf{A}$ e $y \in \mathbf{B}$ são agora elementos de conjuntos nebulosos e a função $f: F(X) \rightarrow F(Y)$, mapea conjuntos nebulosos de X em conjuntos nebulosos de Y . Supondo ainda que a função f possa mapear um conjunto de n universos X_1, X_2, \dots, X_n , tal que

$$f: F(X_1, X_2, \dots, X_n) \rightarrow F(Y), \text{ onde } y = f(x_1, x_2, \dots, x_n)$$

A função de pertinência $\mu_B(y)$ de $\mathbf{B} = f(\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_n)$ será dada por:

$$\mu_B(y) = \text{max}_{y=f(x_1, x_2, \dots, x_n)} \{ \min [\mu_{A_1}, \mu_{A_2}, \dots, \mu_{A_n}] \}$$

Este é chamado de princípio da extensão e podemos entendê-lo da seguinte forma: dada uma função qualquer f mapeando x_1, x_2, \dots, x_n em y , a função de pertinência dos elementos de y é a maior pertinência para todos os valores x_1, x_2, \dots, x_n do domínio que geram este mesmo y . A função de máximo é utilizada pois estamos nos baseando em valores discretos, para funções contínuas, a equação fica:

$$\mu_B(y) = \text{sup}_{y=f(x_1, x_2, \dots, x_n)} \{ \min [\mu_{A_1}, \mu_{A_2}, \dots, \mu_{A_n}] \}$$

O princípio da extensão nos garante várias propriedades que mantém a consistência matemática de várias operações com conjuntos nítidos quando estendidas para conjuntos nebulosos(Klir, 1995).

- I. $f(\mathbf{A}_i) = \emptyset$ se e somente se $\mathbf{A}_i = \emptyset$
- II. Se $\mathbf{A}_1 \subseteq \mathbf{A}_2$, então $f(\mathbf{A}_1) \subseteq f(\mathbf{A}_2)$
- III. $f(\bigcup \mathbf{A}_i) = \bigcup f(\mathbf{A}_i)$, onde $i = 1, 2, \dots, n$
- IV. $f(\bigcap \mathbf{A}_i) \subseteq \bigcap f(\mathbf{A}_i)$, onde $i = 1, 2, \dots, n$
- V. Se $\mathbf{B}_1 \subseteq \mathbf{B}_2$, então $f^{-1}(\mathbf{B}_1) \subseteq f^{-1}(\mathbf{B}_2)$
- VI. $f^{-1}(\bigcup \mathbf{B}_i) = \bigcup f^{-1}(\mathbf{B}_i)$, onde $i = 1, 2, \dots, n$

$$VII. f^{-1}(\bigcap \mathbf{B}_i) = \bigcap f^{-1}(\mathbf{B}_i), \text{ onde } i = 1, 2, \dots, n$$

$$VIII. \overline{f^{-1}(\mathbf{B})} = f^{-1}(\overline{\mathbf{B}})$$

$$IX. \mathbf{A} \subseteq f^{-1}(f(\mathbf{A}))$$

$$X. \mathbf{B} \supseteq f(f^{-1}(\mathbf{B}))$$

2.2.7 Números Nebulosos

Os números são utilizados para quantificar atributos físicos da realidade. Estes atributos estão por vezes associados a imprecisão ou mesmo conceitos humanos vagos. Em alguns casos, por exemplo um especialista pode dar a temperatura de uma área como sendo “em torno de 25 graus”. O método clássico de tratar com as imprecisões físicas é colocar uma barra de erro no valor. Diz-se que a temperatura é $20^{\circ}\text{C} \pm 10^{\circ}\text{C}$. Esta manipulação de números por intervalos também é utilizada para operar com números nebulosos(Klir, 1995).

A desvantagem para o processo de manipulação por intervalos é que reduz uma parte da informação. Se a temperatura válida é de 10 a 30, é pouco consistente dizer, para a maioria das aplicações, que a temperatura de 9.99°C é inválida e passa a ser válida 0.01°C depois.

Neste ponto, a teoria nebulosa nos permite definir um conjunto nebuloso “próximo de” (Figura 2-8), que permite expressar o grau de relevância da temperatura com a proximidade de seu valor central, levando assim em conta a informação de imprecisão de uma forma mais adequada.

A curva da Figura 2-8 é apenas um caso particular para a representação de um número nebuloso. Um número nebuloso pode ser visto como uma generalização dos números reais, cuja função de pertinência seria uma singularidade no ponto correspondente ao número(Figura 2-9).

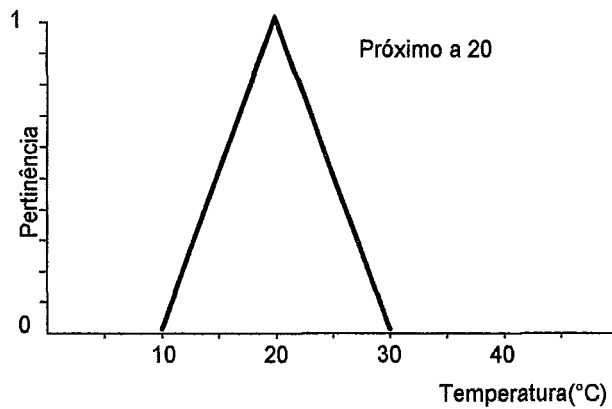


Figura 2-8 - Exemplo de número nebuloso

Um valor de intervalo clássico $[a, b]$ pode ser representado igualmente representado por uma função de pertinência do tipo:

$$\mu = \begin{cases} 1, & \text{se } x \in [a_1, b] \\ 0, & \text{se } x \notin [a_1, b] \end{cases}$$

Na Figura 2-10 podemos ver a representação da função característica de um intervalo. Esta função pode ser estendida para definir um intervalo nebuloso(Figura 2-11).

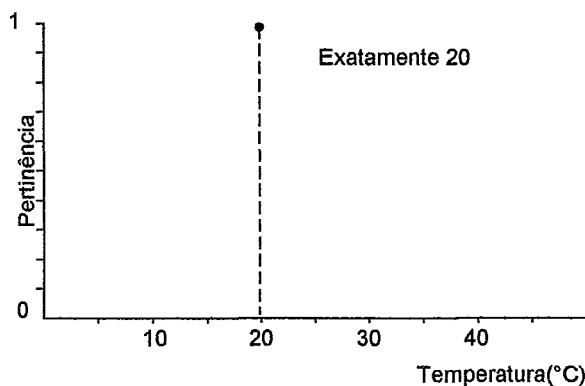


Figura 2-9 - Número nítido

Vários tipos de curvas são propostas na literatura(Belchior,1997, Cox,1994b, Klir,1995). Uma classe delas são as curvas “em forma de sino” (Klir,1995). Um exemplo pode ser visto na Figura 2-12.

Existem vários tipos, como a curva *Beta*, *Gaussiana* e a mais usada, a curva *PI*(Cox,1994b). Cada curva tem uma característica diferente, como por exemplo a

mudança de ponto de inflexão ou largura. Isto tem por finalidade melhor adaptar a curva ao conceito de imprecisão do número. Algumas curvas podem não ser simétricas(Klir,1995).

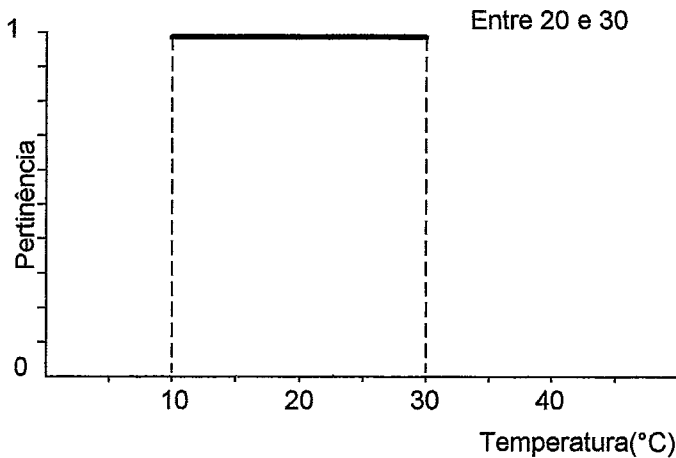


Figura 2-10 - Intervalo de valores

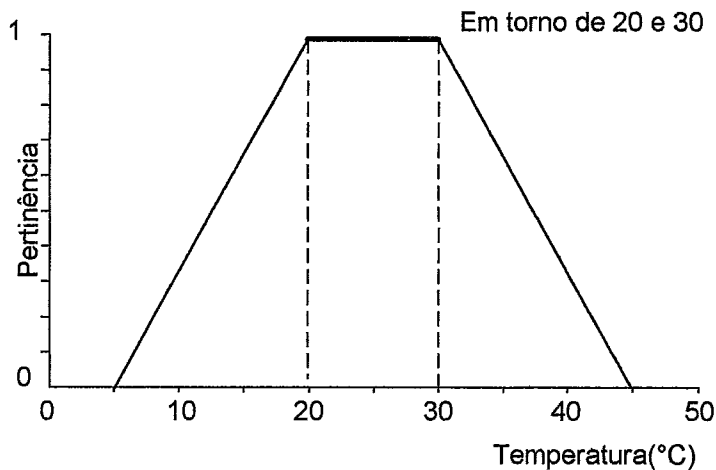


Figura 2-11 - Intervalo nebuloso

Para um conjunto nebuloso A ser adequadamente caracterizado como um número nebuloso, cuja função de pertinência deve ser dada por: $\mu_A: \mathfrak{R} \rightarrow [0,1]$, ele deve atender a certas restrições:

i) A deve se normal - ou seja, $\sup \mu_A = 1$. Isto pode ser facilmente compreendido se considerarmos que um número nebuloso é uma generalização dos números nítidos, ou seja, deve haver algum valor para o qual o grau de relevância mantenha a propriedade original do número nítido.

ii) A deve ser um intervalo fechado e convexo para todo corte-alfa $\alpha \in (0,1]$. Esta propriedade impede contradição na interpretação do número nebuloso. Ao ser convexo, o conjunto nebuloso converge para o mesmo valor com o crescimento da função de pertinência, tanto pela direita, como pela esquerda.

iii) O suporte de ${}^{0+}A$ deve ser limitado.

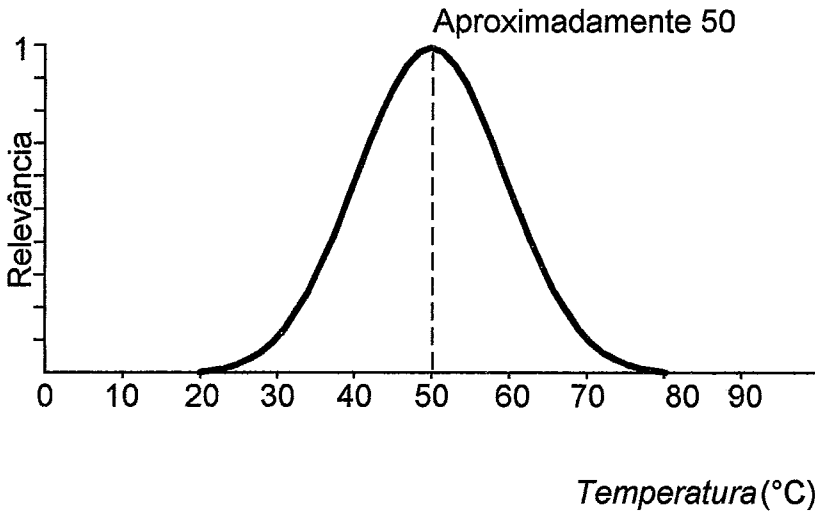


Figura 2-12 - Número Nebuloso

As funções de pertinência podem ser genericamente definidas como conjuntos tipo-lr (Belchior, 1997, Klir, 1995). Através da definição das funções como lr, dizemos que um conjunto A é um número nebuloso, se e somente se, existir um intervalo fechado (restrição ii) $[a,b] \neq \emptyset$ tal que,

$$\mu_A(x) \begin{cases} 1 & \text{para } x \in [a,b] \\ l(x) & \text{para } x \in [-\infty, a] \\ r(x) & \text{para } x \in [b, \infty] \end{cases}$$

Sendo $\mu = 1$ para $x \in [a,b]$, garantimos a restrição (i). As funções l e r , esquerda e direita respectivamente, são ambas monotônicas. Para cada função l crescente, ou r decrescente, a monotonicidade garante a restrição (ii).

2.2.8 Aritmética Nebulosa

Através da extensão fornecida pela teoria nebulosa, podemos incorporar o conceito da incerteza aos números, melhorando assim a representatividade da informação. Para manipular estes números, torna-se necessário estender também as

operações com os números. As operações entre conjuntos vistas anteriormente tentavam apenas estender e manter a consistência das operações clássicas entre conjuntos, aqui desejamos estender operações clássicas entre números. Antes os conjuntos nebulosos eram vistos como agregação de elementos associados a uma dada pertinência, aqui eles serão vistos como números nebulosos.

Um dos métodos utilizados para estender as operações aritméticas é utilizar a manipulação de intervalos(Klir,1995) onde, dada uma operação nebulosa qualquer

$$* \in \{+, -, \times, \div\} \quad (A * B) = \bigcup_{\alpha \in (0,1]} \alpha (A * B) \text{ onde,}$$

$\alpha(A * B)$ é definido através de operações com intervalos.

Outra forma de se calcular uma operação aritmética nebulosa é através do princípio da extensão(Seção 2.2.6). Uma operação aritmética z qualquer entre dois operandos nítidos x e y pode ser escrita por $z = f(x,y)$. Dados dois conjuntos nebulosos A e B , onde $x \in A$ e $y \in B$, a extensão nebulosa da função f , pelo princípio da extensão fica:

$$(A * B)(y) = \{ (z, \mu(z)) \mid \mu(z) = \sup_{y=f(x,y)} \{ \min [\mu_A(x), \mu_B(y)] \} \}$$

2.2.9 Comparações entre conjuntos nebulosos

As características próprias dos conjuntos nebulosos trazem a necessidade de se re-definir as operações básicas de comparação: $=$, \neq , $<$, $>$, \geq e \leq . Estas operações, quando aplicadas a valores precisos, retornam verdadeiro ou falso de acordo com o resultado da comparação. Quando se manipulam com conjuntos nebulosos este resultado pode assumir valores intermediários indicando a pertinência da operação.

São sugeridos (Li & Liu,1990) 4 métodos de cálculo de igualdade entre conjuntos nebulosos: inclusão, similaridade, distância semântica e compatibilidade.

Inclusão: Indica o quanto um conjunto nebuloso A está inserido num conjunto B . Utiliza-se como forma de se definir igualdade entre conjuntos nebulosos. A relação de inclusão, definida por Zadeh,(1965) é escrita como:

$$A \subseteq B \text{ se e somente se } \forall x, \mu_A(x) \leq \mu_B(x)$$

O conceito de Zadeh é muito restrito, outras formas, como proposta por Sanchez(1977) em Li,Liu(1995), podem ser utilizadas:

$$I(\mathbf{A},\mathbf{B}) = \frac{\|\mathbf{A} \cap \mathbf{B}\|}{\|\mathbf{A}\|}$$

Similaridade: A indicação de igualdade entre conjuntos nebulosos, também proposta por Zadeh(1965) é:

$$\mathbf{A} = \mathbf{B} \text{ se e somente se } \mathbf{A} \subseteq \mathbf{B} \text{ e } \mathbf{B} \subseteq \mathbf{A}$$

Este método, como a inclusão, retorna somente verdadeiro ou falso. Muitos autores preferem definir a similaridade como um índice nebuloso que indica o grau de semelhança entre os conjuntos (Sanchez,1977 em Li,Liu,1990):

$$S(\mathbf{A},\mathbf{B}) = \frac{\|\mathbf{A} \cap \mathbf{B}\|}{\|\mathbf{A} \cup \mathbf{B}\|}$$

Distância Semântica: É outra forma de se avaliar o grau de semelhança entre conjuntos nebulosos. Em Li,Liu(1990) são apresentadas 4 propostas:

a) Distancia de *Hamming*:

$$S_d(\mathbf{A},\mathbf{B}) = \sum_i^n |\mu_A(x_i) - \mu_B(x_i)|, \text{ para um conjunto nebuloso discreto e finito}$$

ou,

$$S_d(\mathbf{A},\mathbf{B}) = \int_a^b |\mu_A(x_i) - \mu_B(x_i)| dx \quad \text{ caso contrário.}$$

b) Distancia de *Hamming* ponderada:

$$S_d(\mathbf{A},\mathbf{B}) = \sum_i^n w(x_i) |\mu_A(x_i) - \mu_B(x_i)|, \text{ para um conjunto nebuloso discreto e}$$

finito ou,

$$S_d(\mathbf{A},\mathbf{B}) = \int_a^b w(x_i) |\mu_A(x_i) - \mu_B(x_i)| dx \quad \text{ caso contrário, onde}$$

o termo $w(x_i)$ é o peso de x_i

c) Distancia euclidiana:

$$S_d(\mathbf{A}, \mathbf{B}) = \sum_i^n |\mu_A(x_i) - \mu_B(x_i)|^2,$$

d) Distancia de *Chebyshev*:

$$S_d(\mathbf{A}, \mathbf{B}) = \max_{1 \leq i \leq n} |\mu_A(x_i) - \mu_B(x_i)|$$

Compatibilidade: O grau de compatibilidade entre conjuntos é o único que fornece a semelhança em termos de um conjunto nebuloso. A pertinência de um elemento à um conjunto nebuloso pode ser interpretada como o grau de compatibilidade do elemento ao conjunto. A idéia da compatibilidade entre conjuntos é feita estendendo-se este conceito pelo princípio da extensão, logo $\mu_A(\mathbf{B})$ sendo o grau de compatibilidade de \mathbf{A} com o conjunto nebuloso \mathbf{B} :

$$\mu_A(\mathbf{B}) = \mu_T(u) = \sup_{u = \mu_A(x)} \mu_B(x) \quad \forall \in [0,1]$$

Várias outras formas propostas são baseadas nos princípios acima ou extensões dos mesmos. Três tipos de comparações podem surgir utilizando princípios ou conjuntos nebulosos:

1) Comparações nebulosas entre números precisos

As comparações clássicas ($=$, \neq , $<$, $>$, \geq e \leq) entre números precisos já está suficientemente definida na literatura, contudo uma extensão destes operadores levando em conta a imprecisão também é considerado (Li, Liu, 1990). Dados dois números precisos x e y :

a) $x \cong y$ - Indica se um número preciso é aproximadamente igual ao outro. A relevância deste tipo de comparação é gerada por funções da diferença entre os valores, como por exemplo:

$$\mu_{\cong}(x, y) = e^{-C|x-y|}, \text{ onde } C \text{ é um parâmetro.}$$

b) $x \gg y$ e $x \ll y$ - Indicam comparações do tipo “Muito maior que” ou “Muito menor que”. Este tipo de comparação também é subjetiva e adequada a uma interpretação nebulosa.

$$\mu_{\gg}(x,y) = i \begin{cases} 0 & \text{se } x \leq y \\ [1 + C(x-y)^{-2}]^{-1} & \text{se } x > y \end{cases} \quad C > 0 \text{ é um parâmetro.}$$

A definição de $\mu_{\ll}(x,y)$ é simplesmente feita como sendo:

$$\mu_{\gg}(x,y) = \mu_{\ll}(y,x)$$

2) Comparações entre conjuntos nebulosos e números precisos

Os operadores de igualdade, desigualdade ou semelhança(=, \neq , \cong) entre conjuntos nebulosos segue o princípio da pertinência do valor preciso ao conjunto nebuloso em questão Zadeh(1977), Dubois Prade,1980 em Li,Liu(1990). O conceito do operador de semelhança(\cong) se confunde com o de igualdade nesse caso:

$$\mu_{\cong}(\mathbf{A},x) = \mu_{=}(\mathbf{A},x) = \mu_{\mathbf{A}}(x)$$

O operador de diferença corresponde ao complemento da igualdade:

$$\mu_{\neq}(\mathbf{A},x) = 1 - \mu_{\mathbf{A}}(x)$$

Os operadores de comparação (<, >, \geq e \leq .) são definidos em termos de integrais parciais Li,Liu(1990). Estes cálculos se inspiram no conceito de inclusão entre conjuntos nebulosos:

$$\mu_{\geq}(\mathbf{A},a) = \frac{\int_a^{\infty} \mu_{\mathbf{A}}(x) dx}{\int_{-\infty}^{\infty} \mu_{\mathbf{A}}(x) dx} \quad \text{e} \quad \mu_{>}(\mathbf{A},a) = \frac{\int_a^{\infty} \mu_{\mathbf{A}}(x) dx}{\int_{-\infty}^{\infty} \mu_{\mathbf{A}}(x) dx}$$

$$\mu_{\leq}(\mathbf{A},a) = \frac{\int_{-\infty}^a \mu_{\mathbf{A}}(x) dx}{\int_{-\infty}^{\infty} \mu_{\mathbf{A}}(x) dx} \quad \text{e} \quad \mu_{<}(\mathbf{A},a) = \frac{\int_{-\infty}^{a-1} \mu_{\mathbf{A}}(x) dx}{\int_{-\infty}^{\infty} \mu_{\mathbf{A}}(x) dx}$$

Os operadores \gg e \ll são definidos similarmente ao ítem acima, sendo que, \mathbf{A} é substituído pelo *valor esperado* $\mathbf{E}(\mathbf{A})$ do conjunto nebuloso.

3) Comparações entre conjuntos nebulosos

As comparações entre conjuntos nebulosos possuem diversas formas sugeridas e que podem ser baseadas nos métodos de cálculo de inclusão, similaridade, distância semântica ou compatibilidade.

Em Li,Liu(1990) sugere-se calcular a relevância dos operadores de igualdade, desigualdade ou semelhança(=, ≠, ≅) entre conjuntos nebulosos utilizando-se a distância semântica $S_d(\mathbf{A},\mathbf{B})$.

A igualdade(=) e a semelhança(≅) se confundem e correspondem ao complemento da distância semântica:

$$\mu_{=}(\mathbf{A},\mathbf{B}) = 1 - S_d(\mathbf{A},\mathbf{B})$$

O operador diferença é diretamente proporcional a distância semântica:

$$\mu_{\neq}(\mathbf{A},\mathbf{B}) = S_d(\mathbf{A},\mathbf{B})$$

Sugere-se na literatura que estes cálculos poderiam também ser realizados utilizando-se inclusão ou similaridade, como descrito acima.

As comparações de ordenação entre os conjuntos nebulosos possuem várias formas de cálculo sugeridas(Li,Liu,1990,Klir,Yuan,1995) uma delas se inspira em princípios de inclusão:

Dados dois conjuntos nebulosos \mathbf{A} e \mathbf{B} e uma condição de comparação $\mathbf{A} \theta \mathbf{B}$, onde $\theta = \{<, >, \leq, \geq\}$, a pertinência da comparação será dada por:

$$\mu_{\theta}(\mathbf{A},\mathbf{B}) = \frac{\sigma_1}{\sigma_1 + \sigma_2} \text{ onde,}$$

$$\mathbf{A} = \mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + \dots + \mu_A(x_n)/x_n \quad \mathbf{B} = \mu_B(y_1)/y_1 + \mu_B(y_2)/x_2 + \dots + \mu_B(y_n)/y_n$$

$$\sigma_1 = \begin{cases} \sum_{(x_i, y_j) \in P_1} \min(\mu(x_i), \mu(y_j)), & P_1 \neq \emptyset \\ 0, & P_1 = \emptyset \end{cases} \quad \sigma_2 = \begin{cases} \sum_{(x_i, y_j) \in P_2} \min(\mu(x_i), \mu(y_j)), & P_2 \neq \emptyset \\ 0, & P_2 = \emptyset \end{cases}$$

P_1 e P_2 são dois subconjuntos do produto cartesiano P de todos os pontos x e y tal que: $P = \{(x_i, y_j) | i=1, 2, \dots, k, j=1, 2, \dots, n\}$ eles representarão os grupos de elementos de cada conjunto nebuloso comparado dois a dois:

$$P_1 = \{(x_i, y_j) | x_i \theta y_j \}$$

$$P_2 = \{(x_i, y_j) | x_i \bar{\theta} y_j \}$$

- $\bar{\theta}$. representa o operador complementar do intervalo, como por exemplo, para $\theta =$ “>”, $\bar{\theta}$ será igual a “≤” .

Outro tipo de comparação(Li,Liu,1990,Klir,Yuan,1995) de igualdade é sugerida para ordenar conjuntos nebulosos, contudo é mais utilizado na manipulação de números nebulosos. Utiliza-se do princípio da extensão aplicado aos operadores de comparação:

$$\mu_{\theta}(\mathbf{A},\mathbf{B}) = \sup_{x \theta y} \min(\mu_{\mathbf{A}}(x),\mu_{\mathbf{B}}(y))$$

Klir e Yuan(1995) sugerem um princípio mais restrito onde um conjunto **A** somente seria considerado θ de **B**(**A**<**B** por exemplo) se o conjunto nebuloso cuja pertinência é $\mu_{\theta}(\mathbf{A},\mathbf{B})$ for igual a **A** para todo x, y. Em outras palavras, o princípio da extensão retorna um outro conjunto nebuloso que será o maior ou o menor do dois, de forma nebulosa. Este conjunto poderá ser uma combinação dos dois ou um dos dois, caso esta ultima condição ocorra dizemos que a comparação é rígida.

Li e Liu(1990) ainda propõe uma outra forma de comparação rígida entre conjuntos nebulosos. Ele compara o valor médio entre os valores de pertinência unitária de cada curva de pertinência dos dois conjuntos usando este valor como decisão de comparação:

$$\begin{aligned} x_{\min} &= \min(x_i \mid \mu_{\mathbf{A}}(x_i) = 1) & x_{\max} &= \max(x_i \mid \mu_{\mathbf{A}}(x_i) = 1) & x &= (x_{\min} + x_{\max}) / 2 \\ y_{\min} &= \min(y_j \mid \mu_{\mathbf{B}}(y_j) = 1) & y_{\max} &= \max(y_j \mid \mu_{\mathbf{B}}(y_j) = 1) & y &= (y_{\min} + y_{\max}) / 2 \end{aligned}$$

O comparação rígida entre **A** e **B** é dada por:

$$\mathbf{A} \parallel \theta \parallel \mathbf{B} \text{ se e somente se } x \theta y.$$

Os operadores de << e >> também são sugeridos para o cálculo entre conjuntos nebulosos, contudo não apresentam grandes diferenças entre o apresentado no item 1. Os valores esperados de cada conjunto são substituídos na função com já mostrado.

As formas de comparação são bastante diversas e sua aplicação está ligada a adequação do sistema a ser implementado. Na Tabela 2-1 apresentamos um resumo das

operações de comparação. A versatilidade de formas dentro de um mesmo sistema fornece a possibilidade de vários testes para determinação da melhor implementação.

Operador: θ $A = \mu_A(x)$ $B = \mu_B(y)$	Valor Preciso \times Valor Preciso	Valor Nebuloso \times Valor Preciso	Valor Nebuloso \times Valor Nebuloso
=	<i>Clássico</i>	$\mu_{=} (A, x) = \mu_A(x)$	<ul style="list-style-type: none"> Zadeh(1965): $A \subseteq B$ e $B \subseteq A$ Distância Semântica $\mu_{=} (A, B) = 1 - S_d(A, B)$
\cong	Função da diferença Ex: $\mu_{\cong}(x, y) = e^{- x-y }$	$\mu_{\cong} (A, x) = \mu_A(x)$	<ul style="list-style-type: none"> Similaridade $S(A, B) = \frac{\ A \cap B\ }{\ A \cup B\ }$ Inclusão $I(A, B) = \frac{\ A \cap B\ }{\ A\ }$
\neq	<i>Clássico</i>	$\mu_{\neq} (A, x) = 1 - \mu_A(x)$	<ul style="list-style-type: none"> Complemento
<		$\mu_{<} (A, a) = \frac{\int_{-\infty}^{a-1} \mu_A(x) dx}{\int_{-\infty}^{\infty} \mu_A(x) dx}$	<p><u>Li, Liu(1990):</u></p> $\mu_{\theta}(A, B) = \frac{\sigma_1}{\sigma_1 + \sigma_2}$ $\sigma_1 = \begin{cases} \sum_{(x_i, y_j) \in P_1} \min(\mu(x_i), \mu(y_j)), & P_1 \neq \emptyset \\ 0, & P_1 = \emptyset \end{cases}$ $\sigma_2 = \begin{cases} \sum_{(x_i, y_j) \in P_2} \min(\mu(x_i), \mu(y_j)), & P_2 \neq \emptyset \\ 0, & P_2 = \emptyset \end{cases}$ <p>$P = \{(x_i, y_j) i=1, 2, \dots, k, j=1, 2, \dots, n\}$</p> <p>$P_1 = \{(x_i, y_j) x_i \theta y_j\}$</p> <p>$P_2 = \{(x_i, y_j) x_i \bar{\theta} y_j\}$</p> <p><u>Li, Liu(1990), Klir, Yuan, (1995)</u></p> $\mu_{\theta}(A, B) = \sup_{x \in \theta} \min(\mu_A(x), \mu_B(y))$
\leq		$\mu_{\leq} (A, a) = \frac{\int_{-\infty}^a \mu_A(x) dx}{\int_{-\infty}^{\infty} \mu_A(x) dx}$	
>		$\mu_{>} (A, a) = \frac{\int_a^{\infty} \mu_A(x) dx}{\int_{-\infty}^{\infty} \mu_A(x) dx}$	
\geq		$\mu_{\geq} (A, a) = \frac{\int_a^{\infty} \mu_A(x) dx}{\int_{-\infty}^{\infty} \mu_A(x) dx}$	
<<	Função da diferença, se $x < y$	Função da diferença, se $x < y$. Utilizam o valor esperado $E(x)$.	
>>	Função da diferença, se $x > y$	Função da diferença, se $x < y$. Utilizam o valor esperado $E(y)$.	Função da diferença, se $x < y$. Utilizam o valor esperado $E(x)$ e $E(y)$.

Tabela 2-1 - Resumo dos operadores de comparação

2.3 Lógica Nebulosa

A lógica é o estudo dos métodos e princípios do raciocínio em todas as suas possíveis formas(Klir,1995). A lógica clássica é baseada no conceito de afirmações que podem ser verdadeiras ou falsas, servindo de base para construção de regras e modelos, os quais são utilizados em vários tipos de sistemas.

No mundo real as regras e afirmações nem sempre possuem grau de verdade absoluto, ou seja nem sempre são verdadeiras ou falsas, mas às vezes possuem “graus” de verdade. A aplicação da teoria nebulosa na lógica clássica(Zadeh,1988, Klir,1995a) deu origem a uma extensão chamada de *Lógica Nebulosa*, onde as proposições e operações lógicas podem ter sua incerteza modelada mais adequadamente. Ela tem sido aplicada em várias áreas, como engenharia, economia e indústria ou recuperação de dados e informação(Cox,1994a,1994b, Klir,1995b, Kosko,1993, Li,Liu,1990,Ross,1995).

2.3.1 Proposições Lógicas Nebulosas

Dada uma mistura de vinho e água, que podemos afirmar sobre o líquido? O líquido é vinho ou água? Uma das soluções consideradas poderia ser definir um limite, 50% por exemplo, para o qual definiríamos o líquido como água ou vinho contudo, se houvessem partes iguais teríamos um ponto de singularidade e um paradoxo.

A lógica nebulosa(Zadeh,1988) permite que uma proposição possua valores contínuos entre os limites dos valores verdade, desta forma o operador $T(\cdot)$ passa a ser definido por(Ross,1995):

$$T: \mathbf{P} \rightarrow [0,1] \text{ e } T(\mathbf{P}) = \mu_A(x) \text{ onde,}$$

$\mu_A(x)$ é a função de pertinência de um elemento x ao conjunto A que representa os valores verdade da proposição \mathbf{P} , definindo assim o grau de verdade da proposição.

Proposições e Conectivos lógicos

Os conectivos lógicos nebulosos são *isomórficos* aos apresentados na seção anterior, contudo, a sua função característica, agora definida pela função de pertinência $\mu(x)$, assume valores contínuos entre 0 e 1. Seja o universo de discurso X , Dadas duas

proposições $\mathbf{P}(x)$ e $\mathbf{Q}(y)$, onde dois conjuntos $\mathbf{A}, \mathbf{B} \subset X$ contém os elementos $x \in X$ que as satisfazem respectivamente, temos os conectivos:

- a) Disjunção(\vee) - $\mathbf{P}(x) \vee \mathbf{Q}(x)$: x é \mathbf{A} ou x é \mathbf{B} , $T(\mathbf{P} \vee \mathbf{Q}) = \max(\mu_{\mathbf{A}}(x), \mu_{\mathbf{B}}(x))$
- b) Conjunção(\wedge) - $\mathbf{P}(x) \wedge \mathbf{Q}(x)$: x é \mathbf{A} e x é \mathbf{B} , $T(\mathbf{P} \wedge \mathbf{Q}) = \min(\mu_{\mathbf{A}}(x), \mu_{\mathbf{B}}(x))$
- c) Negação($\bar{}$) - $\bar{\mathbf{P}}(x)$: $T(\bar{\mathbf{P}}) = 1 - T(\mathbf{P}) = 1 - \mu_{\mathbf{A}}(x)$
- d) Implicação(\rightarrow)(Zadeh, 1973) - $\mathbf{P}(x) \rightarrow \mathbf{Q}(x)$: x é \mathbf{A} então x é \mathbf{B} , $T(\mathbf{P} \rightarrow \mathbf{Q}) = T(\bar{\mathbf{P}} \vee \mathbf{Q}) = \max(1 - \mu_{\mathbf{A}}(x), \mu_{\mathbf{B}}(x))$

A implicação lógica pode ainda ser modelada como uma relação nebulosa entre o antecedente e o conseqüente. As proposições nebulosas criadas a partir de uma regra ou implicação são classificadas como condicionais(Klir, 1995) e as proposições simples como incondicionais. Cada uma delas pode ser ou não qualificada, ou seja um termo lingüístico nebuloso pode inferir o grau de verdade da proposição.

Proposições Incondicionais e não qualificadas

Seja uma variável lingüística(seção 2.2.4) V pertencente a um universo de discurso X , uma proposição \mathbf{P} é expressada como:

$$\mathbf{P}: V \text{ é } \mathbf{A},$$

onde \mathbf{A} é um conjunto nebuloso. Um elemento x representando um valor da variável V pertence a \mathbf{A} tem um grau de pertinência $\mu_{\mathbf{A}}(x)$. Suponhamos, por exemplo, que a V_i represente o valor do atributo “Idade” do banco de dados, na tupla i . Seja ainda \mathbf{A} um conjunto nebuloso representado pelo termo lingüístico “Jovem”. A proposição \mathbf{P} indicando que o elemento i é jovem seria:

$$\mathbf{P}: \text{Idade}(i) \text{ é Jovem, onde,}$$

- $\text{Idade}(i) = V_i$
- $\mathbf{A} = \text{“Jovem”}$

O grau de verdade da proposição seria expresso pela pertinência do elemento i ao conjunto “Jovem”:

$$\mu_A = \text{Jovem}(\text{Idade}(i))$$

Proposições Incondicionais e qualificadas

A qualificação de uma proposição mostra o grau de verdade do próprio grau de verdade da proposição, constituindo um conjunto nebuloso de segundo nível (Klir, 1995b). A proposição qualificada, pelos parâmetros do item 0 é da forma:

P: V é **A é **F**, onde,**

F é um conjunto nebuloso nível 2, ou seja, que faz o mapeamento dos valores de pertinência de (V é **A**) em $[0,1]$. Poderíamos dizer, por exemplo, que as chances do elemento i ser jovem é provável, a proposição ficaria:

P: Idade(i) é Jovem é Provável, onde

$$\text{“Provável”} = \mathbf{F} \text{ e } \mu_{\mathbf{F}} = \text{Provável}(\text{Jovem}(\text{Idade}(i)))$$

Proposições Condicionais e não qualificadas

Uma proposição condicional é uma abordagem da implicação lógica nebulosa descrita acima. Dadas duas variáveis X, Y pertencentes a universos de discurso X e Y , uma proposição condicional **P** terá a forma:

P: Se X é **A, então Y é **B**, onde,**

A e **B** são conjuntos nebulosos pertencentes a X e Y , respectivamente. A proposição pode ser igualmente vista como uma relação nebulosa, onde para cada valor x de X e y de Y definidos sobre o produto cartesiano $X \times Y$, temos a relação:

$$R(x,y) = \mathbf{I}[\mathbf{A}(x), \mathbf{B}(y)]$$

O operador **I** denota uma implicação nebulosa. A função de pertinência $\mu_R(x,y)$ da relação entre os dois valores x e y define o grau de verdade da implicação. Existem muitas funções de pertinência para a implicação, a mais comum é a de Zadeh (1973) descrita acima. Ross (1995) faz uma lista de possíveis funções a serem utilizadas:

- a) $\mu_R(x,y) = \max\{\min[\mu_A(x),\mu_B(y)], 1 - \mu_A(x)\}$. É a implicação clássica, como citada acima (Zadeh,1973)
- b) $\mu_R(x,y) = \max\{\mu_B(y), 1 - \mu_A(x)\}$. É equivalente a equação (a) para $\mu_B(y) \leq \mu_A(x)$. , também chamada de *correlação mínimo*.
- c) $\mu_R(x,y) = \min[\mu_A(x),\mu_B(y)]$. É chamada de *correlação mínimo* ou de *implicação Mamdani*(Mamdani,1976)
- d) $\mu_R(x,y) = \min\{1,[1 - \mu_A(x) + \mu_B(y)]\}$. Conhecida como *implicação Lukasiewicz* (Rescher,1969)
- e) $\mu_R(x,y) = \min\{1,[\mu_A(x) + \mu_B(y)]\}$. Conhecida como *Implicação de soma limitada*.
- f) $\mu_R(x,y) = \min\{1,[\text{Error!}]\}$, $\mu_A(x) > 0$, (Goguen,1969)
- g) $\mu_R(x,y) = \max\{\mu_A(x) \cdot \mu_B(y), [1 - \mu_A(x)]\}$. É uma forma de *Correlação produto*, sugerida por Vadiiee(1993).
- h) $\mu_R(x,y) = \mu_A(x) \cdot \mu_B(y)$. Outra forma de *Correlação produto*.

Proposições condicionais e qualificadas

As proposições deste tipo somente acrescentam a qualificação nebulosa as proposições condicionais descritas acima, da forma:

$$\mathbf{P: Se } X \text{ é } \mathbf{A}, \text{ então } Y \text{ é } \mathbf{B} \text{ é } \mathbf{F}$$

onde **F** é um conjunto nebuloso de segundo nível que faz o mapeamento dos valores de pertinência da relação ou implicação nos graus de pertinência ao conjunto **F**.

2.3.2 Regras e inferência nebulosa

Como verificamos na lógica clássica(seção **Error! Reference source not found.**), a inferência de regras pode ser feita através da composição de uma relação R representando uma regra de implicação e uma proposição qualquer **A'**:

$$\mathbf{B'} = \mathbf{A'} \circ \mathbf{R}$$

Esta forma é chamada de *regra de composição de inferência*(Klir,1995), ou regras de correlação(Cox,1994a,1994b). Ela pode utilizar vários tipos de regras de inferência, dependendo do fato que se tenha e a conclusão que se deseje produzir, como por exemplo, se temos uma regra ($\mathbf{P} \rightarrow \mathbf{Q}$) como, “Se as estruturas do prédio estão abaladas então o prédio cai” e a informação \bar{Q} : “O prédio não caiu”, deduzimos que as estruturas do prédio não estão abaladas.

- a) *modus ponens* - ($\mathbf{P} \wedge (\mathbf{P} \rightarrow \mathbf{Q})$) \rightarrow \mathbf{Q} - Quando temos um fato antecedente e desejamos inferir uma regra de forma a produzir o conseqüente.
- b) *modus tollens* - ($\bar{\mathbf{Q}} \wedge (\mathbf{P} \rightarrow \mathbf{Q})$) \rightarrow $\bar{\mathbf{P}}$ - Quando temos os conseqüentes e desejamos produzir o antecedente.
- c) *Silogismo hipotético* - ($(\mathbf{Q} \rightarrow \mathbf{S}) \wedge (\mathbf{P} \rightarrow \mathbf{Q})$) \rightarrow ($\mathbf{P} \rightarrow \mathbf{S}$) - Quando temos uma regra relacionando o conseqüente e desejamos produzir outra regra relacionada ao antecedente.

Para a regra de inferência *modus ponens*, Ross(1995) levanta algumas funções de pertinência conhecidas utilizadas para produzir a pertinência da *composição nebulosa* resultante:

- a) min-max - $\mu_B(y) = \max_{x \in X} \{ \min[\mu_A(x), \mu_R(x,y)] \}$
- b) max-produto - $\mu_B(y) = \max_{x \in X} \{ \min[\mu_A(x) \cdot \mu_R(x,y)] \}$
- c) min-max - $\mu_B(y) = \min_{x \in X} \{ \max[\mu_A(x), \mu_R(x,y)] \}$
- d) max-max - $\mu_B(y) = \max_{x \in X} \{ \max[\mu_A(x), \mu_R(x,y)] \}$
- e) min-min - $\mu_B(y) = \min_{x \in X} \{ \min[\mu_A(x), \mu_R(x,y)] \}$
- f) max-média - $\mu_B(y) = \frac{1}{2} \max_{x \in X} [\mu_A(x) + \mu_R(x,y)]$
- g) soma-produto - $\mu_B(y) = f \left\{ \sum_{x \in X} \mu_A(x) \cdot \mu_B(y) \right\}$

Todos estes recursos produzidos pela extensão da lógica clássica têm por objetivo nos permitir modelar de forma mais abrangente os conhecimentos de uma área de conhecimento. No estudo de bancos de dados eles nos permitirão modelar variáveis ou atributos através de regras.

2.4 Bancos de Dados

As imprecisões do mundo real trouxeram a busca por ferramentas capazes de manipular dados com imprecisões que nem sempre podem ser devidamente inferidas. Os modelos e fundamentos de bancos de dados estão baseados em lógica e teoria de conjuntos, desta forma, a lógica nebulosa e a teoria dos conjuntos nebulosos vêm a ser tornar uma base para construção de um modelo de banco de dados nebuloso. A adequação da teoria nebulosa ao tratamento de dados imprecisos permitirá que um banco de dados nebuloso possa armazenar, consultar e realizar operações com tais tipos de dados.

O modelo de banco de dados mais utilizado e mundialmente aceito é o modelo relacional(Codd,1970). A maioria dos autores propõe extensões deste modelo para lidar com dados nebulosos(Buckles,Petry,1982, Dubois,Prade,1990, Li,Liu,1990, Petry,1996, Shenoj,Melton,Fan,1990).

2.4.1 Modelo Relacional Nebuloso

O modelo relacional, introduzido por Codd,(1970) é definido por um conjunto de relações, regras de integridade e operadores sobre as relações. Dada uma coleção de domínios D_1, D_2, \dots, D_n e elementos $d_1 \in D_1, d_2 \in D_2, \dots, d_n \in D_n$, uma relação R é definida como:

$$R = (D_1 \times D_2 \times, \dots \times D_n).$$

A relação R define todas as possíveis combinações dos elementos dos domínios. Uma relação nítida representa a presença ou ausência de associação, interação ou interconectividade entre elementos de 2 ou mais conjuntos(Klir,1995). Uma relação nebulosa estende este conceito permitindo graus de interação entre os elementos. Ela é definida pelo produto cartesiano dos conjuntos de subconjuntos nebulosos A_i de cada universo de discurso X_i , chamados de conjuntos potência nebuloso, $\mathcal{F}(X_i)$, tal que,

$$R = (\mathcal{F}(X_1) \times \mathcal{F}(X_2) \times \dots \times \mathcal{F}(X_n)), \text{ onde,}$$

dados n conjuntos $A_i \subset X_i$, a pertinência da relação de um grupo de elementos $x_i \in A_i$ é dada por

$$\mu_R(x_1, x_2, \dots, x_n) = \min(\mu_{A_1}(x_1), \mu_{A_2}(x_2), \dots, \mu_{A_n}(x_n))$$

As relações nebulosas são conjuntos nebulosos onde cada elemento representa a relação entre elementos dos universos de discurso, X_1, X_2, \dots, X_n . A pertinência de cada elemento à relação nebulosa R expressa o grau de relação entre eles. Como um conjunto nebuloso qualquer, as operações básicas se aplicam a elas (Ross, 1995):

- a) União: $\mu_{R \cup S}(x_1, x_2, \dots, x_n) = \max(\mu_R(x_1, x_2, \dots, x_n), \mu_S(x_1, x_2, \dots, x_n))$
- b) Interseção: $\mu_{R \cap S}(x_1, x_2, \dots, x_n) = \min(\mu_R(x_1, x_2, \dots, x_n), \mu_S(x_1, x_2, \dots, x_n))$
- c) Complemento: $\mu_{\bar{R}}(x_1, x_2, \dots, x_n) = 1 - \mu_R(x_1, x_2, \dots, x_n)$

Assim como as relações entre conjuntos nítidos, as propriedades de *comutatividade*, *associatividade*, *distributividade*, *involução* e *idempotência* se mantêm, contudo, as leis de exclusão do centro não são necessariamente mantidas:

$$R \cup \bar{R} \neq E$$

$$R \cap \bar{R} \neq O$$

onde, E representa o conjunto nebuloso completo ($\mu_E(x) = 1, \forall x$) e O representa o conjunto nulo ($\mu_O(x) = 0, \forall x$). Isto é verificado pelo fato de dois elementos sempre poderem estar relacionados num certo grau ao mesmo tempo que não estão relacionados em outro grau. O chamado modelo relacional nebuloso (Li, Liu, 1990, Petry, 1996, Buckles, Petry, 1982) está baseado numa abordagem de uma relação nebulosa. Aqui a relação nebulosa modela o relacionamento entre os domínios dos atributos das tabelas. Dada uma coleção de domínios D_1, D_2, \dots, D_n , representando universos de discurso, e n conjuntos $A_i \subset D_i$, uma relação nebulosa R é descrita como (Li, Liu, 1990):

$$R = (D_f(A_1) \times D_f(A_2) \times \dots \times D_f(A_n)), \text{ onde}$$

$D_f(A_i) = F(D_i) \cup \{\emptyset\}$ representa o conjunto dos subconjuntos nebulosos de D_i mais o valor nulo. A função de pertinência é similarmente calculada.

Normalização e regras de integridade

Uma das questões que surgem no modelo relacional nebuloso se relaciona a normalização. Em muitos casos e modelos utilizados(Li,Liu,1990,Petry,1996) um atributo pode ter um conjunto de valores nebulosos, como por exemplo, uma pessoa pode ser Jovem(30%), Adulta(80%) e Velha(40%). Para lidar com este tipo de abordagem, são citados vários autores(Petry,1996) que relaxam a primeira forma normal permitindo que um atributo assumir valores de conjunto. Ela é chamada de não primeira forma normal , ou NF².

Outro ponto importante a ser tratado, ligado às regras de integridade referencial, é com referência aos valores nulos. No modelo nebuloso a incerteza de um atributo pode ganhar uma dimensão a mais. Três tipos de valores nulos são usualmente citados(Li,Liu,1990,Petry,1996):

- Indecisos(Li,Liu,1990) ou desconhecidos(Petry,1996) - Este tipo de valor nebuloso significa que sabemos que existe um valor mas não temos idéia qual é.
- Indefinidos(Liu,Liu,1990) ou inaplicáveis(Petry,1996) - Este tipo indica que não existe valor possível, como perguntar a data do parto de um ser do sexo masculino.
- Nulo(Petry,1996) ou Total Ignorância(Petry,1996) - Indica total desconhecimento do atributo, ou seja, ele pode existir e não sabemos qual é ou não ser aplicável.

A inclusão da teoria nebulosa no modelo relacional nebuloso tem duas importantes abordagens encontradas na literatura, a primeira relacionada a similaridade entre valores(Li,Liu,1990, Petry,1996, Buckles,Petry, 1982,Shenoi,Melton,1990) e a segunda relativa a distribuição de possibilidade para as tuplas(Petry,1996, Buckles,Petry,1982).

Modelo de Similaridade

A principal importância do modelo de similaridade(Petry,1996) está relacionada a identificação das tuplas de uma relação. No modelo relacional convencional os valores dos atributos são nítidos e podemos identificar diretamente se dois valores são iguais ou não. No modelo nebuloso, dois valores podem ser semelhantes, mas não iguais e nem diferentes. Tomemos como exemplo a Tabela 2-2, João e Mario são

ambos Jovens, porem diferenciados por um grau de relevância. João e Renato, contudo, tem a mesma idade mas com avaliações lingüísticas diferentes.

Pessoa	Idade	Valor
João	Jovem(60%)	30
Mario	Jovem(80%)	25
Renato	Velho(40%)	30

Tabela 2-2 - Relação Idade

Dado um conjunto de subconjuntos nebulosos $\mathcal{F}(D_i) = (A_1, A_2, \dots, A_n)$ uma relação de similaridade $s(A_k, A_j)$ entre dois conjuntos nebulosos A_k e A_j é escrita por:

$$s: A_k \times A_j \rightarrow [0,1], \text{ onde } A_k, A_j \subset \mathcal{F}(D_i)$$

A relação de similaridade deve atender a 3 propriedades básicas:

- a) Reflexividade: $s(A_k, A_k) = 1$
- b) Simetria: $s(A_j, A_k) = s(A_k, A_j)$
- c) Transitividade: $s(A_k, A_l) \geq \max_j \{ \min [s(A_k, A_j), s(A_j, A_l)] \}$ ou
 $s(A_k, A_l) \geq \max_j \{ s(A_k, A_j) * s(A_j, A_l) \}$

A Tabela 2-3 mostra a matriz de similaridade entre os dois conjuntos. A matriz de similaridade deve ser de alguma forma levantada para cada domínio da relação R. Isto é feito adequadamente dependendo de cada tipo de aplicação.

	A_1	A_2	...	A_n
A_1	1.0	$s(A_1, A_2)$	$s(A_1, \dots)$	$s(A_1, A_n)$
A_2		1.0	$s(A_2, \dots)$	$s(A_2, A_n)$
...			1.0	...
A_n				1.0

Tabela 2-3 - Relações de similaridade

A relação de similaridade pode conter cortes alfa, analogamente aos conjuntos nebulosos. Uma relação de similaridade S_α , chamada de α similar é dada por:

$$S_\alpha = \begin{cases} s(A_j, A_k), & \text{se } s < \alpha \\ 0, & \text{se } s \geq \alpha \end{cases}$$

A similaridade mínima ou limite de similaridade de uma relação é definida como a menor similaridade entre todos os domínios D_i . Dadas duas tuplas $t_i = \langle d_{i1}, d_{i2}, \dots, d_{ij} \rangle$ e $t_k = \langle d_{k1}, d_{k2}, \dots, d_{kj} \rangle$, elas são similares se,

$$\text{Limite}(D_j) \leq \min_j s(d_{ij}, d_{kj}), \text{ onde}$$

$\text{Limite}(D_j)$ indica o valor de similaridade mínima para se considerar a tupla como redundante. Alguns autores sugerem a variação deste limite para criação de partições do domínio (Shenoi, Melton, 1990). Este modelo é usado em bancos de dados cujos atributos possuem valores nebulosos não atômicos (Petry, 1996).

A similaridade entre os valores nebulosos pode ser calculada utilizando os cálculos de igualdade mostrados na seção 2.2.9.

Modelo Possibilístico

A abordagem possibilística, citada por Petry (1996) considera o valor de pertinência do atributo, neste caso com valores atômicos, para compor a pertinência de cada tupla. Este valor de pertinência pode ter várias interpretações distintas:

1. Representando o grau de adequação da relação, ou seja o quanto os valores satisfazem a relação. Podemos ter por exemplo, a relação $\langle \text{Pessoa}, \text{Jovem} \rangle$, onde a pertinência de cada tupla vai indicar o quanto cada pessoa atende ao atributo "Jovem".
2. Representando a certeza da informação armazenada. Podemos ter como exemplo a relação $\langle \text{Animal}, \text{Localização} \rangle$. A pertinência da tupla irá indicar a certeza da localização registrada de cada animal.
3. Representando a possibilidade da informação armazenada. É semelhante à anterior, com a diferença de que a relação torna-se mais fraca quanto menor a sua possibilidade.

2.4.2 Processamento de Consultas

Arquiteturas

As arquiteturas propostas para o processamento de consultas envolvendo incertezas possuem 3 abordagens básicas(Bosc e Pivert,1997a,1997b). O diagrama de cada uma pode ser visto na Figura 2-13.

A primeira abordagem se constitui de uma divisão da consulta em duas partes. A primeira é constituída de sentenças booleanas convencionais e é submetida diretamente ao SGBD. A segunda parte, contendo sentenças imprecisas ou nebulosas é utilizada a posteriori para executar um “ranking” nas tuplas recuperadas na primeira parte da consulta. As duas partes são na verdade conectadas por uma conjunção AND. A parte imprecisa pode ser constituída por proposições nebulosas do tipo “Em torno de 30” e combinadas pelos operadores E e OU nebulosos usualmente definidos pelos operadores min e max. Um dos sistemas citados(Bosc e Pivert,1997a) é o PREFERENCES(Lacroix e Lavency,1987). O problema principal deste tipo de arquitetura está no fato de que a consulta imprecisa não faz parte efetivamente da consulta, apenas é usada para “ranking”.

A segunda abordagem permite a utilização direta de sentenças imprecisas. As sentenças são traduzidas transformando os valores imprecisos em intervalos. Em alguns sistemas é utilizado o modelo de similaridade onde o usuário deve definir os limites de confiança(ARES, Motro,1988) ou através de funções normalizadas de similaridade dentro do domínio dos atributos(VAGUE, Ichikawa e Hirakawa,1986). A vantagem deste modelo, bem como o anterior é a sua simplicidade de implementação, Bosc e Pivert(1997b) inclusive sugerem a derivação da parte booleana da consulta para um SGBD convencional, resultando assim num aproveitamento das vantagens já existentes do sistema.

O problema básico desta abordagem é que ao traduzir os termos imprecisos em intervalos os limites dos mesmos apresentam a mesma descontinuidade dos valores nítidos.

A terceira abordagem, inicialmente pesquisada por Tahani(1976,1977) e outros, é baseada na teoria dos conjuntos nebulosos(Zadeh,1965) e utiliza o conceito de relação

nebulosa. As operações entre as relações são as extensões das operações básicas (união, interseção, etc.) para as relações nebulosas. As sentenças envolvem diretamente consultas precisas e imprecisas, utilizando conjunções e disjunções entre ambos os tipos de consultas. Os atributos dos predicados envolvem variáveis lingüísticas, termos nebulosos e modificadores. Esta abordagem ainda propõe a introdução de quantificadores nebulosos às consultas. Ela é a mais complexa de implementação, sendo sugerida a inclusão de algoritmos especializados dentro do SGBD permitindo a extensão do uso de suas capacidades. No modelo criado por Li e Liu (1990) para um banco de dados nebuloso baseado em PROLOG, a arquitetura do sistema (Figura 2-14) proposto se divide em nível Interno, Conceitual e Externo, correspondendo respectivamente às visões física, lógica e conceitual do banco de dados.

No nível do usuário é estabelecida a linguagem de definição de dados nebulosa e de consulta. A característica especial da linguagem é a definição de atributos nulos ou incompletos, a qual é feita como na seção 2.4.1.

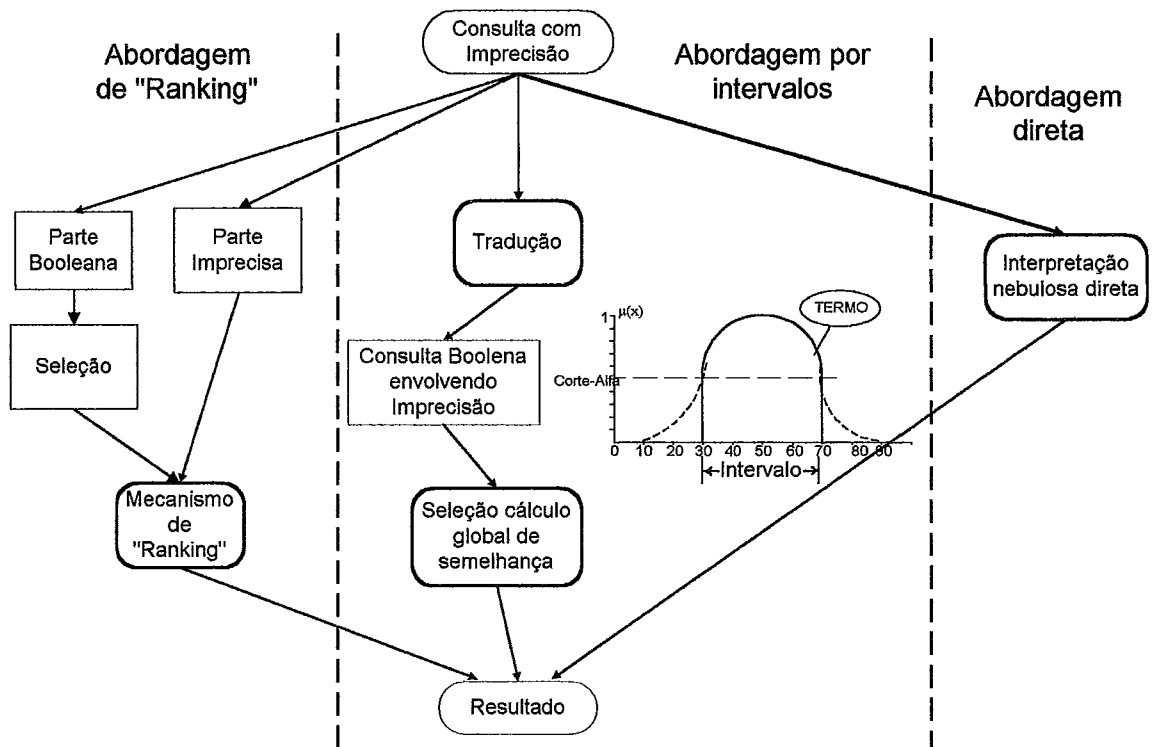


Figura 2-13 - Abordagens de arquitetura nebulosa

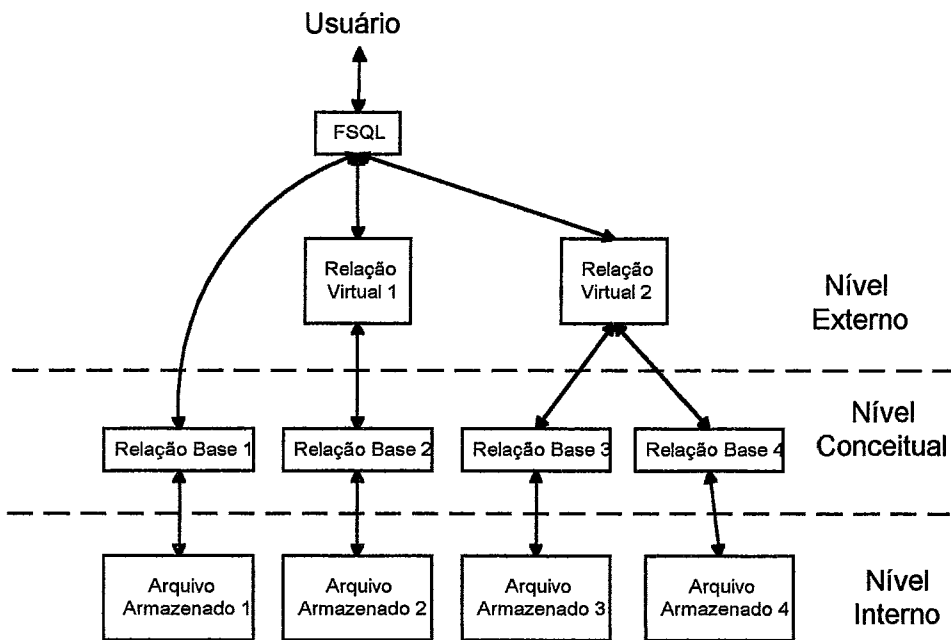


Figura 2-14 - Arquitetura em visões

Linguagens de Consulta

Todas as linguagens de consulta incorporando a teoria nebulosa utilizam-se o SQL como base. De um modo geral as linguagens tem o mesmo formato proposto para o SQL. O que difere está nas sentenças de criação das tabelas e nas proposições utilizadas nas condições da consulta.

Os comandos de criação das tabelas podem possuir palavras-chave especiais para designar os atributos que serão nebulosos (Petry, 1996) ou simplesmente assumem que qualquer variável pode admitir valores nebulosos (Li e Liu, 1990).

Em alguns casos a linguagem dispõe de um termo extra no SQL padrão para designar o grau de confiança nebuloso das tuplas do resultado (Li, Liu, 1990):

```

SELECT    <Atributos>
FROM      <Relações>
WHERE     <Condições>
WITH      <Grau de pertinência ou termo nebuloso de segunda ordem>

```

Suponhamos, por exemplo, que queiramos recuperar todos os funcionários jovens de uma empresa com certeza maior que 80%. A sentença nebulosa ficaria:

```

SELECT    IDADE, FUNCIONARIO
FROM      EMPRESA
WHERE     IDADE = Jovem
WITH      0.8

```

O termo seguinte à cláusula WITH pode também ser um termo nebuloso mapeando o grau encontrado na tupla numa outra possibilidade, como por exemplo, “Possível”:

```

SELECT    IDADE, FUNCIONARIO
FROM      EMPRESA
WHERE     IDADE = Jovem
WITH      Possível

```

O resultado desta consulta seria uma lista de funcionários e a pertinência de sua tupla em relação ao termo “Possível”.

Os operadores de seleção, projeção, união e etc. tem o mesmo formato do SQL padrão, contudo, seu método de cálculo difere, desta forma uma álgebra nebulosa é em geral proposta sobre múltiplas formas.

Álgebra relacional Nebulosa

As propostas de extensão nebulosa da álgebra relacional são feitas sobre a extensão das operações entre as relações nebulosas(Li,Liu,1990, Petry,1996, Shenoj et al,1990). As operações da álgebra estão divididas em dois grupos, o primeiro se refere as operações básicas entre conjuntos, o segundo contém as operações de seleção, projeção e junção.

Dadas duas relações nebulosas compatíveis R e S com funções de pertinência $\mu_R(t)$ e $\mu_S(t)$, onde t indexa as tuplas de R e S , as operações básicas de conjunto são(Li,Liu,1990):

- **União**($R \cup S$): As tuplas resultantes da união de R e S podem não aparecer simultaneamente em ambas as relações, desta forma, cada tupla t que aparece nas duas relações terá seu valor de pertinência calculado pelo máximo das duas tuplas:

$$\mu_{R \cup S}(t) = \max\{\mu_R(t), \mu_S(t)\}$$

As tuplas de uma relação que não aparecem na outra levam o seu próprio valor de pertinência. As duas relações devem ser compatíveis, ou seja, o número de atributos e os domínios de cada atributo devem ser os mesmos. Esta restrição se aplica as operações de interseção e diferença.

- **Interseção**($R \cap S$): Cada tupla t que está em R e S tem sua pertinência calculada por:

$$\mu_{R \cap S}(t) = \min\{\mu_R(t), \mu_S(t)\}$$

- **Diferença**($R - S$): A diferença entre duas relações pode também ser vista como a interseção de R com a negação de S, cuja função de pertinência é dada por:

$$\mu_{R-S}(t) = \min\{\mu_R(t), (1 - \mu_S(t))\}$$

- **Produto Cartesiano**($R_1 \times R_2 \times \dots \times R_n$): O produto cartesiano pode ser feito por relações não compatíveis. O conjunto de tuplas final $t=(t_1, t_2, \dots, t_n)$ é composto pelo produto cartesiano das tuplas t_i de cada relação R_i . A pertinência do produto cartesiano é dada por:

$$\mu_{R_1 \times R_2 \times \dots \times R_n}(t) = \min\{\mu_{R_1}(t_1), \mu_{R_2}(t_2), \dots, \mu_{R_n}(t_n)\}$$

- **Seleção** - A avaliação de um predicado ϕ sobre uma relação R envolvendo um conjunto de atributos A_1, A_2, \dots, A_n , um conjunto de elementos do domínio de cada atributo $D_f(A_1), D_f(A_2), \dots, D_f(A_n)$ e um conjunto de operadores nebulosos op_1, op_2, \dots, op_n é feita substituindo-se cada elemento $d_i \in D_f(A_i)$ de cada tupla nos atributos A_i envolvidos. A pertinência $\mu_{\sigma_\phi}(R)$ de cada tupla da seleção $\sigma_\phi(R)$ é feita através do cálculo entre os operadores nebulosos, que podem ser lógicos(AND,OR,NOT) aritméticos(+,-, *, / , =, < , > , ≤ , ≥) ou operadores nebulosos especiais.

- **Projeção** - A projeção de uma relação nebulosa $R = (D_f(A_1) \times D_f(A_2) \times \dots \times D_f(A_n))$ sobre um grupo de atributos A_1, A_2, \dots, A_k , $k \leq n$ pode gerar um conjunto de tuplas redundantes. As tuplas t de menor pertinência são eliminadas, ou seja,

$$\mu_R(d_{i_1}, d_{i_2}, \dots, d_{i_k}) = \max_{d_j \in D_f(A_j), j \in \{1, 2, \dots, i_k\}} \{\mu_R(d_1, d_2, \dots, d_n)\}$$

Nome	Idade	μ
João	Jovem	0.5
Maria	Adulto	0.7

Tabela 2-4 - (R1)Relação projetada

- **Junção** - A operação de junção consiste de um produto cartesiano seguido de uma seleção. Em Li,Liu,(1990) a junção é calculada sobre as extensões cilíndricas(Zadeh,1965) de cada relação em relação a todos os atributos do universo, desta forma, a comparação entre duas tuplas da junção pode ser sempre feita entre tuplas compatíveis. A função de pertinência final da junção é dada por:

$$\mu_{\triangleright \triangleleft_{i=1}^s R_i}(a_1, a_2, \dots, a_n) = \min\{\mu_{R_1}(a_1, a_2, \dots, a_n), \dots, \mu_{R_s}(a_1, a_2, \dots, a_n)\}$$

Neste processo leva-se em conta somente o grau de relevância das extensões cilíndricas. Petry(1996) coloca ainda dentro do cálculo a pertinência do predicado de junção:

$$\mu_{R \triangleright \triangleleft_{A \theta B} S}(x, y) = \min\{\mu_R(x), \mu_S(y), \mu_{\theta}(x, A, y, B)\}$$

De todo modo, não fica claro como dois valores nebulosos de 2 tuplas podem ser comparados, visto que, no caso de conjuntos nebulosos que não possuam igualdade mais estrita(Zadeh,1965), ainda pode haver similaridade.

Cálculo Relacional

O cálculo nebuloso de predicados tem a mesma forma do cálculo de predicados de primeira ordem(Li,Liu,1990,Petry,1996). A diferença básica está em que a interpretação das fórmulas não assume somente verdadeiro ou falso mas graus de pertinência.

Dada uma relação $R_{(\lambda)} = \{t \mid \mu_R(t) \geq \lambda\}$, onde λ é o nível mínimo de pertinência da tupla t variando em R , uma expressão é definida recursivamente pelas suas fórmulas atômicas Ψ que somente podem ser:

- $\mu_R(t)$
- $t[i] \vartheta C$ ou $C \vartheta t[i]$ - Sendo que $t[i]$ é o i -ésimo atributo da tupla t e ϑ pode ser qualquer operador de comparação nebuloso ou não. C é uma constante que pode ser um valor nebuloso ou não.

- $t[i] \wp u[j]$, onde t e u são atributos de duas tuplas como definido acima.
- $\neg\Psi$ - A Operação nebulosa NOT aplicada a fórmula.
- $\Psi_1 \wedge \Psi_2$ - A conjunção de duas fórmulas através do operador nebuloso AND.
- $\Psi_1 \vee \Psi_2$ - A disjunção de duas fórmulas através do operador nebuloso OR.

O cálculo de predicados é basicamente utilizado para as operações de seleção de tuplas. A avaliação das fórmulas resulta num grau de pertinência que classifica ou rejeita as tuplas selecionadas numa consulta. Isto pode ocasionar em perda de informação, visto que algumas implementações da operações de AND e OR nebulosas podem dar resultados iguais para vários valores de pertinência diferentes, como por exemplo, uma consulta do tipo:

```

SELECT    PACIENTE, PRESSÃO, TEMPERATURA
FROM      PACIENTES
WHERE     PRESSÃO = Alta AND TEMPERATURA = Alta

```

Na Tabela 2-5 podemos ver a que as duas tuplas recuperadas apresentariam o mesmo grau de pertinência, apesar de que a ordem de importância do paciente P1, com maior TEMPERATURA parece mais visível.

PACIENTE	PRESSAO	μ	TEMPERATURA	μ	ACUCAR	μ
P1	Alta	0.6	Alta	0.9	Baixo	0.2
P2	Alta	0.6	Alta	0.7	Baixo	0.7

Tabela 2-5 - Exemplo de classificação

Berthier e Muntz(1995) propõe que as tuplas deveriam ser comparadas levando-se em conta todos os operandos do predicado da consulta. A comparação não seria na mesma ordem dos atributos da relação mas teria a ordem alterada através de operadores sincronizados com as operações de conjunção e disjunção. Um predicado da forma:

$\Psi = (\text{PRESSÃO} = \text{Alta} \text{ AND } \text{TEMPERATURA} = \text{Alta}) \text{ OR } (\text{AÇÚCAR} = \text{Baixo})$

seria escrito convencionalmente por:

$$\mu_{\Psi}(t) = \max(\min(\mu_{\text{Pressão}}(t), \mu_{\text{Temperatura}}(t)), \mu_{\text{Açúcar}}(t)),$$

onde t é uma tupla qualquer pertencente a R . Berthier e Muntz(1995) propõe dois operadores, chamados de F e G , os quais substituem as operações de mínimo e máximo. O operador F ordena os seus operandos de forma crescente e G de forma decrescente. A lista de graus de pertinência devidamente ordenada de cada tupla é então comparada e classificada. A consulta Ψ seria escrita como:

$$\Psi = G(F(\text{Pressão}, \text{Temperatura}), \text{Açúcar}),$$

A classificação obtida é dada na Tabela 2-6, onde os graus de pertinência são ordenados da esquerda para a direita.

PACIENTE	μ	μ	μ
P1	0.6	0.9	0.2
P2	0.7	0.6	0.7

Tabela 2-6 - Classificação pela álgebra F-G

Todas as arquiteturas vistas e propostas aqui visam estender as operações lógicas e a teoria dos conjuntos para adaptar o modelo e o cálculo relacional, respectivamente. Existem outras propostas, utilizando por exemplo o banco de dados orientado por objetos ou o modelo rede(Petry,1996). A maioria destas propostas utiliza um banco de dados nebuloso que deve ser construído ou convertido de um banco de dados convencional. Isto traz uma série de questões relativas a modelagem nebulosa de dados precisos, ponto que não é muito discutido nos trabalhos de bancos de dados nebulosos, apesar das arquiteturas propostas(seção 2.4.2), sendo mais discutido dentro da área de controle de processos(Cox,1994a,1994b).

A filosofia de bancos de dados nebulosos tem sido proposta em vários tipos de sistemas onde se requer tratamento de imprecisão. Os Sistemas de Informação Geográfica(SIG) se utilizam dela em duas abordagens, uma relativa aos dados convencionais com que tratam, outra relativa ao tratamento de dados espaciais.

Capítulo III

3. Sistemas de Informação Geográfica

Sistemas de Informação Geográfica(SIGs) são definidos das mais variadas formas, dependendo por exemplo, de sua área de atuação, utilização ou funcionalidades. A falta de formalização deste tipo de sistema proporciona um múltiplo número de definições pelos mais diversos autores(Maguire,1991).

A construção de um SIG está fundamentada em três aspectos básicos(Maguire,1991): o mapa, a base de dados e as análises espaciais.

O aspecto referente aos mapas relacionam-se a todas as operações de manipulação e representação cartográfica. Esta abordagem levanta todas as questões referentes a armazenamento, estruturas de dados espaciais, escala, visualização de dados, etc. Na seção 3.3 trataremos mais detalhadamente das questões que envolvem este aspecto dos SIGs.

Dentro da visão da base de dados, tratamos da questão da modelagem da realidade em modelos de dados compatíveis com a manipulação que se deseja fazer. Devemos tratar das questões que envolvem os tipos de dados não convencionais dos SIGs. Na seção 3.2 trataremos deste assunto mais especificamente.

O maior potencial de um SIG está na capacidade das suas ferramentas de análise. A análise de dados espaciais é bastante complexa e não convencional, devido a natureza de seus dados. Existem muitos tipos de análises que são extremamente aperfeiçoadas através da utilização de um SIG, como por exemplo, a busca de rotas mais eficientes para transporte de carga. Além disso, existem análises que somente poderiam ser realizadas com sistemas deste tipo.

Na seção seguinte(3.1) apresentaremos as funcionalidades básicas requeridas por um SIG e mostraremos como podemos posicioná-las dentro destes três enfoques básicos.

3.1 Funcionalidades

A correta definição das funcionalidades necessárias a um SIG nos auxilia a classificar e especificar o tipo de SIG que desejamos implantar(Maguire, Raper,1992). As funcionalidades necessárias a um SIG são divididas no seu aspecto espacial e não

espacial. Esta divisão deve-se ao fato das arquiteturas mais convencionais de SIGs serem compostas de um gerenciamento espacial e não espacial separados. As arquiteturas mais recentes, utilizando a orientação para objetos(seção 3.2.2), já fazem um tratamento mais integrado de suas funções. Um resumo dos componentes funcionais(Maguire, Raper, 1992, Maguire, Dangermond, 1991) divididos em espaciais e não espaciais pode ser visto na Tabela 3-1.

Grupos funcionais	Espacial	Não Espacial
Captura	<ul style="list-style-type: none"> • Imagens - <i>Scanners</i>, foto satélite, etc. • Digitalização de dados, GPS, etc. • Conversão de formatos. • Edição e validação 	<ul style="list-style-type: none"> • Teclado • Reconhecimento de voz e texto.
Estruturação	<ul style="list-style-type: none"> • Estruturas espaciais: <ul style="list-style-type: none"> - Tesselações - Mapas poligonais - Híbridas • Topologia 	<ul style="list-style-type: none"> • Bancos de dados <ul style="list-style-type: none"> - Orientados para Objeto - Relacionais - Outros tipos • Arquivos e listas
Manipulação	<ul style="list-style-type: none"> • Restruturação <ul style="list-style-type: none"> - Mudança das estrutruras de dados: agregação de objetos. • Generalização: <ul style="list-style-type: none"> - Alisamento de contornos, Eliminação de ruído. • Transformação <ul style="list-style-type: none"> - Rotação, mudança de escala, projeções, etc. • Consulta <ul style="list-style-type: none"> - Busca, sobreposição de objetos, etc. 	<ul style="list-style-type: none"> • Restruturação <ul style="list-style-type: none"> - Mudança das estrutruras de dados • Generalização: <ul style="list-style-type: none"> - Resumos estatísticos • Transformação <ul style="list-style-type: none"> - Linearização • Consulta <ul style="list-style-type: none"> - Busca, recuperação de dados.
Análise	<ul style="list-style-type: none"> • Adjacência, sobreposição de mapas, <i>buffers</i>, 	<ul style="list-style-type: none"> • Estatísticas • Multivariadas
Representação	<ul style="list-style-type: none"> • Mapas cromáticos • Mapas texturizados • Mapas tri-dimensionais 	<ul style="list-style-type: none"> • Tabelas • Relatórios • Gráficos

Tabela 3-1 - Aspectos funcionais de um GIS

As análises feitas em SIGs são amplamente variadas dependendo do tipo de aplicação. Os tipos de análises consistem em geral de consultas sobre a topologia da base espacial, como na Tabela 3-3(Maguire, 1991), ou então, sobre análises estatísticas espaciais(ex. Nobre, Carvalho, 1996, Dunn, 1987). Openshaw(1991) propõe uma lista básica de análises a serem feitas em SIGs:

1. Espalhamento de padrões
2. Investigadores de relacionamentos
3. Simplificação de dados
4. Detetores de contornos
5. Modelagem espacial automatizada
6. Análise de padrões nebulosos
7. Visualização avançada
8. Análise espacial visual

3.1.1 Representação

Os métodos de representação em SIGs envolvem a representação de elementos espaciais ou gráficos e os dados convencionais representados sobre a base espacial. Existem muitas formas de representação tradicionais de dados em mapas, as mais comuns são mapas cromáticos, padrões hachurados e símbolos. Outras formas mais modernas podem ser vistas, utilizando animação ou representação em três dimensões.

3.2 Modelos de Dados

O objetivo da modelagem de dados consiste em criar esquemas capazes de definir a forma e o comportamento dos dados do mundo real. A modelagem de dados, em bancos de dados convencionais, é bastante conhecida e possui componentes já bem estabelecidos. A modelagem de dados espaciais, por outro lado, possui complexidades além das convencionais, desta forma existem muitas versões de como seria a modelagem apropriada para um SIG.

3.2.1 Modelos

Apesar da grande ambigüidade de termos referente à questão da modelagem de dados, a maioria dos autores (Burrough, 1992b, Maguire e Dangermond, 1991, Frank, 1992, Egenhofer e Herring, 1991) tenta definir três níveis básicos de modelagem:

1. **Modelos da realidade espacial** - feições espaciais.
2. **Modelagem geométrica dos dados** - forma de se abordar as feições espaciais na geometria do espaço.
3. **Modelagem lógica** - Estruturas de dados espaciais (ver seção 3.3).

Modelagem da realidade

A modelagem da realidade, às vezes chamada de modelagem conceitual, consiste em descrever objetos que irão representar as entidades do mundo real. Estes objetos são associados com os dados não espaciais e então chamados de feições geográficas (Maguire e Dangermond, 1991). As feições geográficas mais comuns são baseadas em estruturas de geometria euclidiana:

- **Ponto** - Utilizado para representar elementos geográficos concentrados em uma única coordenada, como cidades, fazendas, ocorrências de eventos, etc.
- **Linha** - Utilizado para representar elementos que se estendem ao longo do espaço através de uma trajetória contínua, com rios, estradas, rotas, fronteiras, etc.
- **Área** - Utilizado para representar extensões de terra ou atributos que se estendam continuamente sobre ela, como lagos, colheitas, estados, etc.
- **Superfície** - Utilizado para representar superfícies no espaço de 3 dimensões, como por exemplo morros.

A associação de cada tipo de atributo da base espacial com as estruturas geométricas formam as feições geográficas (Tabela 3-2). Os atributos não espaciais podem ser divididos em três categorias: nominal, ordinal e intervalo ou taxa de variação. Os atributos nominais não possuem um ordem necessária e apenas indicam a classe ou especificação do valor do atributo, como por exemplo, nome, tipo sanguíneo, etc. Os atributos ordinais diferenciam-se dos nominais apenas no sentido de existir uma ordem dos valores em classes específicas. Podemos ver como exemplos a classificação de índices de qualidade de uma colheita. Os atributos de intervalo ou taxa de variação representam faixas de valores escalados continuamente ou não, como por exemplo faixas de temperatura ou densidade de população. A Tabela 3-2 mostra mais detalhadamente como é a modelagem de cada feição espacial após a combinação de cada tipo de atributo com cada tipo de estrutura geométrica.

Atributo	Ponto	Linha	Área	Superfície
Nominal	Rótulo	Rede	Classes de cores	Cores livres
Ordinal	Símbolos ordenados	Rede Ordenada	Cores ordenadas	Cores ordenadas
Intervalo/taxa de variação	Símbolos Graduados	Linha de Fluxo	espalhamento colorido	Contornos

Tabela 3-2 - Classificação de dados geográficos

Tipo de Consulta	Consulta
Localização	O que existe em ... ?
Condição	Onde está ... ?
Modificação	O que foi alterado em ... ?
Roteamento	Qual o melhor caminho para... ?
Padrão	Qual o padrão ... ?
Modelagem	O que acontece se... ?

Tabela 3-3 - Alguns tipos de consultas feitas em SIGs

3.2.1.1 Modelagem Geométrica

A modelagem geométrica(Frank,1992), às vezes chamada de modelagem lógica(Maguire e Dangermond,1991), às vezes de modelo gráfico(Burrough,1992b), consiste na escolha de como iremos definir o espaço para descrever nossas entidades espaciais. As formas de modelagem estão divididas basicamente em duas classes: Vetorial e Tesselação espacial, também chamadas de formatos, e às vezes modelos(Frank,1992), *Vector* e *Raster*, respectivamente.

A tesselação espacial(Maguire e Raper,1992) está baseada num modelagem contínua do espaço em células(Figura 3-1). Cada célula assume um valor que pode ser relacionado a um atributo da base não espacial, como a indicação da densidade populacional naquele elemento do espaço, ou uma característica própria do espaço, como a vegetação. Este formato é mais eficiente e flexível para sobreposição de

camadas de mapas e em alguns casos, análise de contiguidade, contudo não é tão adequado quando desejamos identificar objetos individuais, como rios, lagos, cidades, etc.

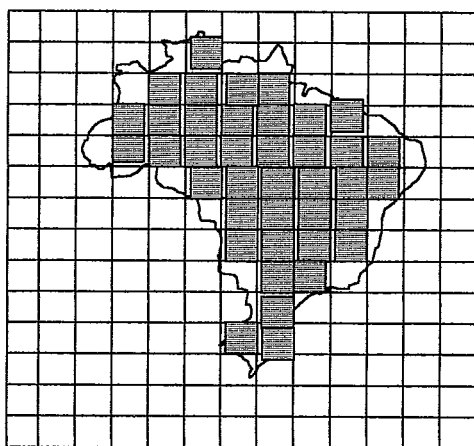


Figura 3-1 - Exemplo de tesselação

No formato vetorial, as entidades geométricas da base espacial são descritas diretamente através das coordenadas de todos os pontos que as compõe. Um segmento de reta, por exemplo, passa a ser representado pelas duas coordenadas que delimitam seus extremos. O formato vetorial é mais usado para modelagem propriamente dita de feições espaciais, pois permite que cada objeto ou feição seja descrita individualmente em sua forma e extensão.

3.2.2 Paradigma Relacional

A questão da modelagem de dados está relacionada com os sistemas utilizados para implementação. Os bancos de dados baseados no modelo relacional(Codd,1970) já estão amplamente difundidos e estabelecidos, cuidando da segurança, integridade, recuperação e tratamento de concorrência(Herring,1992, Batty,1992, Worboys et al.,1994). Em vista disso, tornam-se vantajoso em relação a sistemas *ad-hoc*.

Dada a estrutura complexa dos dados espaciais, o processo de armazenamento e modelagem utilizando o modelo relacional traz uma série de dificuldades. A solução destes problemas faz-se de muitas formas. Uma delas é criar sistemas híbridos, onde o gerenciador espacial beneficia-se das capacidades do banco de dados relacional, armazenando cada entidade num registro, definindo campos para especificação de

coordenadas e extensão dos objetos, como o ARC/INFO(Morehouse,1992) e o GeoManager da IBM(Batty,1992).

Apesar da grande utilização do modelo relacional nos SIGs, as características dos dados espaciais, como a extensão ou a multidimensionalidade, trazem muitos problemas de armazenagem e manipulação, gerando inclusive uma distorção na modelagem dos dados(Yearsley et al.1994, Worboys,1994, Davis,1994). Uma das soluções para essas questões é a utilização do modelo relacional estendido para suportar algumas capacidades do modelo orientado para objetos(Herring,1992).

3.2.3 Paradigma Orientado para Objetos

Os problemas encontrados na tentativa de modelagem dos bancos de dados espaciais através do modelo relacional conduzem os autores na direção dos bancos de dados orientados para objeto(Maguire,1994, Yearsley et al.,1994, Worboys,1994, Milne et al,1993, Davis,1994, Egenhofer,1996, Times,Salgado,Strauch,Mattoso,1994).

O conceito dos bancos de dados orientados para objetos está centrado nas classes. Um objeto de uma classe é uma entidade definida por um grupo de atributos, ou mesmo outros objetos, e uma série de procedimentos próprios a ele. Os procedimentos, chamados de métodos, agem como uma interface que isola a construção dos atributos internos do objeto em relação ao usuário do objeto. Estas características dão adequação aos objetos para modelagem de entidades espaciais.

Quatro características importantes, entre outras, que definem a modelagem orientada por objetos e trazem maior flexibilidade para modelagem de SIGs são(Yearsley et al.,1994, Egenhofer,1996, Milne et al,1993): Encapsulamento, Herança, Polimorfismo e Relacionamento de classe.

3.3 Estruturas de dados

Dados espaciais possuem várias características particulares, como a multidimensionalidade, que trazem a necessidade de se utilizar estruturas de armazenamento e manipulação mais complexas que as estruturas convencionais, como listas, árvores binárias, árvores B, tabelas Hash, etc. Não existe um tipo de estrutura espacial única para uma base espacial ou mesmo para um tipo de entidade espacial(pontos, linhas,etc.). Cada estrutura espacial tem propriedades específicas para

atender um grupo de problemas específicos. A escolha de uma estrutura específica para os dados espaciais em um SIG é dependente da aplicação e influenciada pelo modelo de dados adotado.

As estruturas de dados utilizadas para implementar modelos espaciais está baseada nas duas modelagens geométricas principais(Maguire e Raper,1992, Goodchild,1992): vetoriais e tesselação espacial.

3.3.1 Tesselação Espacial.

Os métodos de tesselação dividem-se basicamente em: regulares e irregulares. No formato regular o espaço é dividido em células de mesmo tamanho. As estruturas espaciais mais comuns para trabalhar com tesselação utilizam a divisão hierárquica do espaço. As mais usadas baseiam-se nas *Quad-Trees* (Samet,1990) e suas variações.

No formato irregular, o espaço pode ser dividido em células de tamanhos diferentes. Temos como exemplos a rede de triângulos irregulares(TIN) e os diagramas de voronoi(Franklin,1991).

3.3.2 Formato Vetorial

As estruturas de dados utilizadas com este formato de dados são classificadas em: não estruturadas, hierárquicas e topológicas(Maguire, Raper, 1992).

Os métodos não estruturados são muito comuns em SIGs de menor porte. As estruturas mais comuns são chamadas de modelo “espaguete”. Este processo consiste em armazenar todos os pontos e linhas de uma base espacial num grande lista de coordenadas. Algumas versões incluem pontos especiais de ligação entre as linhas que se cruzam. As estruturas hierárquicas são baseadas em árvores como as *Quad-trees* (Samet,1990)

Outras estruturas similares são utilizadas para linhas e polígonos. A mais conhecida delas utiliza-se de um princípio análogo ao das Árvores-B, denominadas de R-Trees (Guttman,1984). Outras estruturas adaptadas também são sugeridas(Mediano MR, Casanova MA, and Dreux,1995, Beckmann N, Kriegel H, Schneider R, and Seeger,1990).

3.4 **Processamento de Consultas**

Os tipos de consulta geralmente feitas em SIGs(Tabela 3-3) são bastante complexas. Existem várias propostas de linguagens de consulta ou dialetos

(Egenhofer,1992, Portier,1996, Herring,1992). Cada dialeto pode ser caracterizado por alguns aspectos básicos:

1. **Tipo espacial de dado** - Alguns dialetos fornecem tipos de dados universais para o tratamento de entidades espaciais. Outros tipos fornecem classes pré-definidas, como pontos, linhas, áreas, etc, além de fornecer tipos para as propriedades espaciais, como comprimento ou volume.
2. **Relacionamentos espaciais** - Cada implementação tenta oferecer capacidades para lidar com predicados espaciais de forma similar à convencional, contudo, a complexidade das relações espaciais torna linguagens similares ao SQL pobres para tais manipulações, como proximidade ou contiguidade.
3. **Visualização gráfica** - A representação de dados convencionais é feita basicamente através de tabelas. A representação de dados espaciais é bem mais complexa, pois envolve múltiplas formas de representação. Linguagens como o SQL não possuem sentenças específicas em sua linguagem para definir tais formas de representação.
4. **Seleção visual** - Outro aspecto importante é que em muitos casos o usuário deseja indicar objetos de seleção através do mapa sendo visualizado.

Todas estas questões tentam ser resolvidas dentro das propostas de linguagem existentes, contudo não há ainda uma formalização no assunto. Existe consenso na utilização da extensão orientada para objetos do SQL, a OQL(Herring,1992). Além disso, uma linguagem de consulta espacial deve possuir muitos recursos, inclusive não declarativos, como são as linguagens citadas, como o acesso visual da base. Os autores sugerem ainda acrescentarmos capacidades mais sofisticadas, como consulta de metadados, qualitativas e de recuperação de conhecimento(Egenhofer,1992).

3.5 Observações

Em vista de todas essas questões, a maioria dos autores concorda que os Sistemas de Informação Geográfica possuem várias características não convencionais que são bastante compatíveis com a modelagem orientada para objetos. Esta tendência leva a progredir a tecnologia de implementação dos SIGs utilizando-se bancos de dados orientados para objetos.

A manipulação de dados necessita ainda da implementação de vários algoritmos especiais para manipulação e armazenamento das estruturas espaciais complexas. Os bancos de dados orientados para objetos, fornecem uma base adequada para implementação desses algoritmos.

3.6 Ferramentas de Manipulação Nebulosa

Os sistemas do mundo real, em particular os Sistemas de Informação Geográfica(SIG), contêm um grande número de informações imprecisas. Os sistemas de bancos de dados convencionais não são capazes de lidar com essa imprecisão de forma trivial. A teoria dos conjuntos nebulosos de Zadeh (1965) veio trazer uma nova ferramenta para manipulação e modelagem de imprecisões. Existem muitas referências discutindo que o grande número de incertezas nas informações de origem cartográfica não podem ser somente tratadas utilizando-se os métodos de classificação e análise convencionais, como citado em trabalhos de Burrough (1989, 1992a, 1992b, 1996), Wang (1990a, 1990b, 1994, 1996), Robinson (1988a, 1988b, 1990) e Molenaar(1994,1995,1996), entre outros. A maioria deles propõe a utilização da lógica nebulosa como alternativa para manipular com os diversos tipos de informação com incerteza. Altman (1994) faz alusão aos vários autores que propõe uma ou outra forma de aplicação da teoria nebulosa em SIGs.

Nos SIGs, a abordagem nebulosa divide-se em basicamente em duas categorias: a primeira delas (Wang, Hall, Subaryono,1990, Wang,1994, Kolia, Voliotis,1991, Banai, 1993, Petry,1996) emprega o modelo relacional nebuloso somente aos dados não gráficos². Nessa abordagem, os dados são somente coletados ou representados a partir da base espacial, porém nenhuma análise ou consulta é feita levando-se em conta as características espaciais entre os dados. Poderíamos exemplificar isto numa consulta do tipo:

Selecione todas as cidades com população alta e baixa taxa de mortalidade.

Como podemos observar, este tipo de sistema têm somente que verificar na lista de cidades àquelas que atendem à consulta, a seguir, representar, de alguma forma, num mapa as cidades e o nível de pertinência do resultado da consulta.

² Os dados ditos não gráficos são aqueles cuja existência não depende de nenhuma característica espacial

A outra abordagem (Molenaar, 1996, Dijkmeijer, Hoop, 1996, Wang, Hall, 1996, Altman, 1994, Bernardes, Wu, D'Ottaviano, 1995) é sobre as incertezas de localização espacial dos dados, bem como as relações entre eles. Nesta abordagem, as características espaciais como distância, posição, etc., são modeladas e manipuladas de forma nebulosa. Um exemplo poderia ser,

Selecione todas as cidades próximas de Itatiba

Na maioria dos casos pesquisados existe mais ênfase a uma ou outra abordagem. Cada uma será destinada aos problemas específicos a serem resolvidos. A combinação das duas facilidades traria recursos de grande utilidade, contudo, a maioria dos autores prefere focalizar sua abordagem dentro do problema específico que está tratando.

3.6.1 Incertezas em Bancos de Dados Espaciais

Os SIGs são em grande parte utilizados na área de cartografia. A maioria dos autores concorda com o fato de que há muita incerteza em torno dos dados colhidos e manipulados nesta área. Há vários enfoques de utilização onde surgem incertezas nas bases de dados espaciais, alguns relativos ao processo de classificação de solos, como a adequação para plantação; outros, relativos a imprecisão do processo de aquisição de dados. Há ainda as questões relativas aos processos de manipulação e análise de dados.

As ferramentas de classificação cartográfica são a base para as análises feitas sobre as bases espaciais. Os métodos convencionais dividem os dados de acordo com faixas de valores de atributos. A perda de informação originada por este processo é causada pelo fato que, na maioria das vezes, a transição de uma faixa para outra não se faz de forma abrupta. Nesta pesquisa, um grande número de referências para a utilização da modelagem nebulosa como método de classificação (Wang, 1990a, Hall, Wang, Subaryono, 1991, Fisher, Pathirana, 1989, Burrough, MacMillan, Van Deursen, 1992). Davidson, Theocharopoulos, Blosma, (1994) faz uma comparação entre os dois métodos mostrando a clara adequação provida pela técnica nebulosa. No trabalho de Banai (1993), um processo de classificação hierárquica (Saaty, 1987) é utilizado com lógica nebulosa gerando uma hierarquia ponderada por níveis de relevância.

Ainda dentro dos processos de classificação, uma das questões de maior interesse em cartografia está na adequação do uso do solo. Deseja-se saber qual a adequação natural, ou seja aquela para a qual a terra já está pronta para ser utilizada. Além disso, temos a adequação potencial(Wang et al. 1990), àquela na qual a terra poderá receber fatores para adequá-la. Uma avaliação da terra, fatores climáticos, tipo de vegetação, entre muitos outros fatores geram um aumento de produtividade e eficiência de utilização de recursos.

A ineficiência dos métodos convencionais de tratamento de uso da terra deve-se ao fato do processo de combinação dos vários fatores não ser preciso, assim como os dados processados. Em vista disso, vários autores levantam a questão da incerteza e propõe a utilização da teoria nebulosa como alternativa de processamento e classificação de dados (Wang,Hall,Subaryono,1990, Koliass,Voliotis,1991, Banai,1993, Davidson et. al,1994, Chang,Burrough,1987, Burrough,1989, Fisher,Pathirana,1989, Burrough,MacMillan, Van Deursen,1992, Hall,Wang,Subaryono,1991)

3.6.2 Manipulação de Dados

Dentro do material pesquisado, as abordagens nebulosas de dados para SIGs utilizam o modelo relacional de banco de dados(Codd, 1970). Esta abordagem leva geralmente em conta que os dados não gráficos estão depositados em algum sistema de banco de dados relacional. Esses dados seriam devidamente associados com a base de dados espacial através de algum código ou referência.

Os artigos em geral dividem-se em três questões: a modelagem em si, métodos de processamento e análise e linguagens de consulta. Alguns artigos abordam diretamente só um dos temas e, às vezes, introduzem um termo em outro.

Modelagem

A modelagem nebulosa de dados não gráficos em SIGs é basicamente uma reformulação do modelo de dados relacional nebuloso. Wang (1990) faz esta modelagem usando o *software* de GIS ARC/INFO. No tipo de SIG, onde este tipo de abordagem é feita, as informações estão divididas em dados não gráficos e gráficos ou dados espaciais. Os dados espaciais são caracterizados por entidades espaciais, que podem ser pontos, linhas ou áreas. Os dados não gráficos irão descrever os valores cada atributo da

base de dados pertencentes a cada entidade espacial, como por exemplo, indicar a população de uma área.

O conceito fundamental desta abordagem consiste em considerar o grupo de atributos associado a cada entidade espacial, como os componentes de um vetor. Este vetor, definido sobre um espaço n-dimensional, caracteriza a combinação dos atributos para cada entidade espacial. Esta combinação define uma feição.

No modelo relacional convencional(Codd 1970), uma relação R é definida pelo seu conjunto de atributos $d_i \in D_i$, tal que $R \leftarrow D_0 \times D_1 \times \dots \times D_n$, onde D_i é o domínio do atributo i da relação R. Na extensão proposta $R(S_0, S_1, \dots, S_n)$ é tal que $R \leftarrow S_0 \times S_1 \times \dots \times S_n$ e S_i é uma coleção de objetos identificados por uma condição de pertinência, ou seja cada valor $s_i = (x_i, \mu(x_i))$ onde x_i é o valor do atributo i e $\mu(x_i)$ é o grau de relevância do atributo. O vetor de relevâncias, que define os dados de cada entidade espacial, pode ser definido por $\vec{s} = \langle \mu_1, \mu_2, \dots, \mu_n \rangle$.

Os dados relativos a adequação da terra são modelados tendo em cada registro um campo de associação com cada área específica da base espacial. Wang et al.(1990) usa como exemplo uma Relação $R = (Area\#, p_1, p_2, \dots, p_n)$, onde $Area\#$ representa o código que correlaciona a tupla com a área da base espacial e (p_1, p_2, \dots, p_n) representam os n atributos com as propriedades a serem processadas.

As entidades que representam áreas também podem ser particionadas numa grade de células(Petry, 1996, Jiang, Kainz, 1996). Neste caso, cada célula da grade é considerada como uma nova área vinculada ao mesmo registro de dados.

Análise

Os dois tipos de análise mais comuns com este tipo de abordagem são a classificação de entidades espaciais e a recuperação de informação para alguma adequação. No primeiro tipo de análise(Davidson et. al,1994) deseja-se organizar a base de dados para a melhor compreendê-la. Este processo traz a necessidade de um processo de classificação fiel e sem perda de informação e generalidade. Os dados passam a ser classificados não mais em faixas mas através de funções de pertinência contínuas.

No segundo tipo de análise (Banai, 1993, Wang, 1994, Koliai, Voliotis, 1991) tem-se uma massa de dados que se deseja filtrar para obter entidades espaciais que atendam às características de adequação específicas. Um dos exemplos mais citados é o da adequação de solos para as diversas finalidades agrícolas.

Em Wang et al (1990) a caracterização de cada área j é dada pelo vetor de pertinências $\vec{s}_j = \langle \mu_1, \mu_2, \dots, \mu_n \rangle$. O processo de classificação nítido é baseado no cálculo da distância ou dessemelhança $d_E(\vec{s}, \vec{s}_c)$ entre uma área qualquer \vec{s}_j e uma classe de adequação específica \vec{s}_c onde,

$$d_E(\vec{s}, \vec{s}_c) = \sqrt{\sum_{j=1}^n (s_j - s_{jc})^2} = (\vec{s} - \vec{s}_c)^T (\vec{s} - \vec{s}_c)$$

Uma área \vec{s}_j qualquer pertence à uma classe c se $d_E(\vec{s}, \vec{s}_c) < d_E(\vec{s}, \vec{s}_i)$, para todo $i \neq c$, $i = 1, 2, \dots, m$, sendo m o número total de classes. O processo de classificação nítida sofre a perda da informação referente à adequação parcial de uma área a uma classe. A aplicação da teoria nebulosa nos permite levantar o grau de pertinência f_c de cada área a uma classe específica:

$$f_c(\vec{s}) = \frac{1}{\sum_{i=1}^m \frac{1}{d_E(\vec{s}, \vec{s}_{ic})}}$$

Outro tipo de análise fundamentalmente utilizada é o *Overlay* ou sobreposição de camadas. O trabalho de Jiang e Kainz (1997) fornece um modelo para sobreposição e camadas em modelo *Raster* utilizando métodos de agregação nebulosos para a composição de cada célula.

Linguagem de Consulta

Os modelos de linguagem de consulta utilizados em SIGs, no que se refere a manipulação de dados, não se diferencia muito das linguagens de consulta a bancos de dados nebulosos, citados no capítulo 2. As linguagens, de um modo geral, são similares ao SQL. Elas tomam formas e estruturas relacionadas aos sistemas onde são implementados. Em alguns casos são definidas relações para lidar com as

características espaciais. Em Wang(1994) uma relação “DISTÂNCIA_A_AREA_DE_CAMPING” é criada para lidar com as distâncias entre os pontos da base espacial. Os termos nebulosos são definidos de alguma forma a permitir descrever a distribuição de possibilidade dos atributos. O trabalho de Kolilas e Voliotis(1991) traz uma linguagem com uma gramática bem completa para utilização em análise de solos. Ela permite a definição de dados, recuperação e ordenação nebulosas.

3.6.3 Mapeamento espacial e geométrico

As principais falhas na manipulação de dados em SIG deve-se às características peculiares dos dados espaciais(Peuquet,1984). Uma delas é o fato dos contornos de objetos tenderem a ser bastante irregulares. Dutton(1992) discute os erros introduzidos nas bases durante a sua digitação, descreve os limites de confiança e sua implicação com o uso e a qualidade das bases de dados. Um modelo de discussão sugere a especificação de graus de certeza para o acesso aos objetos, levando ao ajuste da generalização das entidades espaciais.

Uma questão importante é o fato de não permitirem uma definição matemática precisa e mais compacta(Peuquet, 1984). Alesheikh e Li,(1996) não utilizam diretamente a abordagem nebulosa, contudo, descrevem modelos analíticos para as primitivas geométricas em SIGs. Através do modelo para as primitivas geométricas deriva-se o erro dos pontos para as linhas, subseqüentemente estabelecendo as incertezas dos polígonos e contornos.

A teoria nebulosa é utilizada para se construir uma modelagem genérica e abrangente de incertezas espaciais. É importante que primeiro caracterizemos as entidades espaciais. A abordagem sintática de Molenaar(1994,1995,1996) distingue três definições com respeito a existência de um objeto espacial:

- Definição *Existencial* - Assegurando que existem condições espaciais e temáticas.
- Definição *Extensional* - Identificando os elementos geométricos que descrevem a extensão espacial dos objetos.
- Definição *Geométrica* - Identificação da forma, tamanho e posição no sentido métrico.

O trabalho de Molenaar(1996) está mais voltado para a manipulação com dados espaciais em formato *Raster*, contudo em trabalhos anteriores (Molenaar,1994,1995) é mostrado que a geometria *Vector* e *Raster* tem poder de expressão similar. A modelagem utilizada aborda a questão de contornos e topologia nebulosa, como será vista na próxima seção.

As definições *extensional* e *geométrica* são melhor caracterizadas por Duta,(1989,1991) que descreve como podemos classificar as várias características espaciais, como a mobilidade, tamanho, posição, etc. Ele divide as entidades espaciais em duas classificações básicas as quais definem restrições existenciais das entidades:

- I. Posição - Caracteriza a posição no espaço das entidades espaciais. Podemos dividir esta classificação em 4 outras.
 - A. Proposicional - Define descrições lingüísticas quantitativas, da forma X é F .
Exemplo: *Itatiba é próxima de Washington.*
 - B. Métrica - Descreve quantitativamente as posições dos objetos. Exemplo: *Itatiba fica a 10.000km de Washington.*
 - C. Faixa - É uma variação da especificação métrica. Exemplo: *Itatiba fica entre 1.000 e 10.000km de Washington*
 - D. Visual - Esta variante indica quando podemos descrever graficamente áreas nebulosas em torno de objetos.
- II. Movimento - Caracteriza as modificações dos objetos no tempo. Esta classificação leva as mesmas sub-classificações acima descritas. Devido ao objetivo desta pesquisa não abranger a modelagem temporal, achamos desnecessário maior descrição.

As classificações acima não foram exaustivamente exploradas(Duta,1989), podemos notar que existe também uma classificação proposicional de faixa. Estas restrições servem de base para a construção de um modelo nebuloso geral cujas proposições envolverão características nebulosas ou precisas.

Consideremos um proposição p da forma:

$$p = X \text{ é } F,$$

onde X representa uma coleção de variáveis que irão caracterizar a entidade espacial de acordo com as restrições propostas acima. F representa um termo lingüístico ou conjunto nebuloso dentro do universo de discurso e pertencente a X , como por exemplo:

$$p = \text{Itatiba é Grande}$$

Esta proposição seria traduzida por:

$$\Pi_{\text{Tamanho}(\text{Itatiba})} = \text{Grande, onde,}$$

- Π - Poderia ser interpretado pela operação que retorna um valor nebuloso do tipo $(x, \mu(x))$
- Tamanho é uma variável X_I
- Itatiba é um valor do domínio de X_I
- Grande é o conjunto nebuloso pertencente à X_I

O operador $\Pi_{(X_1, X_2, \dots, X_n, Y_1, Y_2, \dots, Y_n)}$ pode representar qualquer restrição de posição ou movimento entre dois objetos, como por exemplo:

Itatiba está em torno de 5.000km

a noroeste de são paulo

$$\Pi_{(\text{Posição}(\text{Itatiba}), \text{Posição}(\text{São Paulo}))} = (5.000\text{km}, \text{Noroeste})$$

Os conjuntos nebulosos “*Em torno de*” e “*noroeste*” seriam modelados como na Figura 3-2 . O artigo propõe um operador “ \circ ” que representaria a relação entre as entidades espaciais, neste caso, modeladas por uma operação de min-máx, desta forma, a proposição acima seria representada por

$$\Pi(A, B) = F,$$

onde A e B são as posições de Itatiba e São Paulo em distância e orientação e F os conjuntos nebulosos representando a distância e orientação entre as duas cidades:

$$A = (\text{distânciaA}, \text{orientaçãoA})$$

$$B = (\text{distânciaB}, \text{orientaçãoB})$$

$$\Pi = \max(\min(\text{distânciaA}, \text{orientaçãoA}), \min(\text{distânciaB}, \text{orientaçãoB}))$$

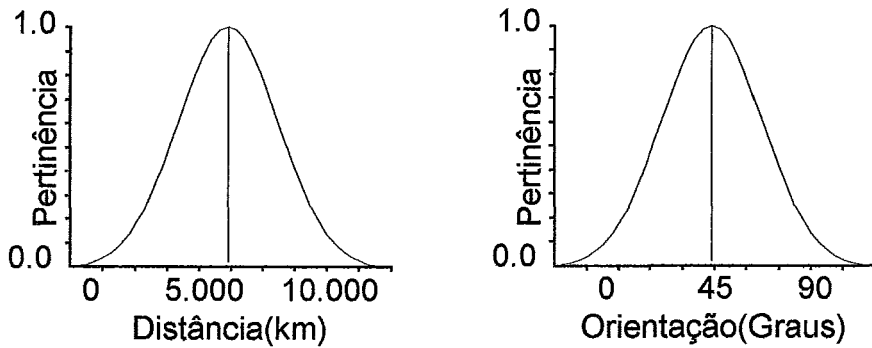


Figura 3-2 - Números nebulosos "Em torno de" e Noroeste

Esta abordagem permite-nos iniciar uma formalização para a relação genérica entre objetos espaciais nebulosos.

Utilizando o modelo *Raster*, Altman (1994) faz um mapeamento nebuloso do espaço. Ele mostra como podemos inferir a distância e a direção nebulosas, como acima citadas, a partir dos objetos extensos, trazendo para estes predicados um conceito mais geral. O seu objetivo é definir o conceito de *região nebulosa* e as relações entre elas. As referências para Leung(1987) definem uma região caracterizada por proposições lingüísticas mas cujos contornos são definidos por gradientes ao longo do espaço.

Uma região nebulosa é caracterizada por pontos com graus de pertinência com duas interpretações possíveis. A primeira delas é o grau que o ponto está dentro ou é parte de alguma entidade ou feição espacial com possíveis contornos definidos(Altman,1994,Jiang e Kainz,1996). Na segunda abordagem a pertinência ou relevância representa a concentração de algum atributo relacionado à feição ou região considerada, como referenciado em (Fisher,Pathirana,1989,1990 e Robinson, 1988a,1988b).

Para qualquer uma das duas abordagens, neste artigo, uma região nebulosa é definida por $R = \{\mu_R(x_1, x_2, \dots, x_n) / x_1, x_2, \dots, x_n\}$, onde x_1, x_2, \dots, x_n representam as coordenadas de um ponto no espaço n-dimensional e μ_R é a pertinência de cada ponto no espaço. A extensão das regiões no espaço não permite a medida de direção e distância entre elas de forma convencional. Elas são definidas entre pontos num espaço n-dimensional onde, dados dois pontos no espaço q_1 e q_2 especificados pelos seus

vetores $\vec{v}_{q_1} = (x_{11}, x_{12}, \dots, x_{1n})$ e $\vec{v}_{q_2} = (x_{21}, x_{22}, \dots, x_{2n})$ temos que a distância entre q_1 e q_2 será dada pelo módulo da diferença entre eles:

$$d(q_1, q_2) = \left| \vec{v}_{p_1} - \vec{v}_{p_2} \right|$$

Através da métrica L_p de Minkowski, teríamos como medida genérica:

$$d_p(q_1, q_2) = \left| \vec{v}_{p_1} - \vec{v}_{p_2} \right| = \left(\sum_{j=1}^n |x_{1j} - x_{2j}|^p \right)^{1/p}, \text{ onde}$$

A distância Euclidiana seria considerada uma métrica L_2 e a métrica *manhattan* seria uma métrica L_1 . A direção de q_1 em relação a q_2 seria dada igualmente por:

$$dir(p_1, p_2) = dir\left(\vec{v}_{p_1} - \vec{v}_{p_2}\right)$$

Estes métodos têm em comum a característica de produzir números escalares e considerar apenas pontos e não regiões como um todo. Geralmente se calcula um *centróide* das regiões. Para expressarmos sem perda de informação as relações entre regiões extensas é necessário que também tenhamos um resultado “extenso”, ou seja, expresso por um conjunto de valores e não por um único valor escalar. Através da teoria dos conjuntos nebulosos podemos expressar estas relações por conjuntos nebulosos, onde temos um conjunto de valores para uma dada relação associados à sua relevância.

Uma região nebulosa bi-dimensional é proposta por Altman(1994) como sendo $R = \{\mu_R(x,y)/(x,y)\}$. Neste exemplo em R^2 a distância nebulosa entre duas regiões nebulosas **A** e **B** é definida como sendo:

$$dist(\mathbf{A}, \mathbf{B}) = \bigcup_{(a,b) \in (\mathbf{A} \times \mathbf{B})} \left[\min(\mu_A(a), \mu_B(b)) / d_2(a,b) \right] \text{ onde,}$$

- $a = (x_a, y_a)$ e $b = (x_b, y_b)$ são dois pontos do espaço pertencentes a **A** e **B** respectivamente e
- $(a, b) \in \mathbf{A} \times \mathbf{B}$ é um par pertencente ao produto cartesiano de todos os pontos das regiões **A** e **B**.
- $d_2(a, b)$ é a distância entre pontos considerando-se um matriz L_2 .

- $\mu(a)$ e $\mu(b)$ representam a relevância dos pontos a e b na região R .

Verificamos até agora como os vários autores modelam o espaço e suas relações de forma nebulosa. Primeiramente temos de definir o espaço, as entidades espaciais e depois mostrar como podemos modelar as relações nebulosas. Indicamos também algumas relações básicas que têm sido mostradas. As relações espaciais nebulosas não têm sido amplamente exploradas, algumas relações não são definidas para o modelo vetorial ou então as entidades espaciais não são exaustivamente caracterizadas. Verificaremos, a seguir, mais um enfoque onde se exploram a definição de contornos e as relações topológicas nebulosas.

Contornos nebulosos

Em geral, em mapas poligonais, as linhas ou contornos representam linhas nítidas que delineiam áreas com características diferentes (Burrough, 1986). Este método é acurado quando se manipulam com dados que possuam característica discreta, com contornos bem definidos, como censo, limites administrativos, etc. Porém, existem propriedades, como feições de terreno que têm uma característica de mudança ao longo do espaço. A representação cartográfica padrão torna-se desapropriada (Burrough, 1986, Chrisman, 1989, Coleman, 1980, Fisher, 1989, Leung, 1984, Robinson, 1980). Mostramos igualmente na seção 3.0 deste capítulo que o trabalho prévio de Wang (1990) já mostrava a séria perda de informação ao se tratar com contornos precisos. Neste trabalho ele propõe um modelo para tratamento nebuloso de contornos.

A idéia básica do trabalho é estabelecer a "nitidez" de um contorno de forma nebulosa. Esta nitidez é representada por um grau de pertinência dado à variação de alguma propriedade, como por exemplo, tipo de solo, cruzando um contorno.

Utiliza-se a derivada de primeira ordem para expressar à variação da propriedade ao longo do espaço. Dado um mapa temático, com uma propriedade a ser representada por $g(l)$, onde l é um ponto no espaço perpendicular ao contorno, a pertinência de um ponto do contorno $l(b)$ é expressa por:

$$\mu_B(b) = f\left(\left|\frac{dg}{dl}(l(b))\right|\right)$$

Como podemos analisar, a pertinência será tanto maior quanto maior a taxa de variação da propriedade, desta forma, quanto mais a propriedade varia, mais nítido é o contorno. A função f vai relacionar os diversos tipos de dados geográficos e suas derivadas.

Uma região pode possuir várias propriedades, como por exemplo, diversos tipos de solos para uma certa área. Cada propriedade está associada a um *poligon fuzzy set*, ou seja, um mesmo polígono no espaço pode ter várias "cópias sobrepostas do polígono" em relação aos atributos nele representados. Desta forma, dado um conjunto de propriedades \mathbf{m} , um vetor \mathbf{g} , representando os n atributos no espaço \mathbf{l} , poderia ser escrito por: $\vec{g}(l) = [m_1(l), m_2(l), \dots, m_n(l)]$, e a taxa de variação destes atributos em \mathbf{l} seria:

$$\frac{d\vec{g}(l)}{dl} = \left[\frac{dm_1(l)}{dl}, \frac{dm_2(l)}{dl}, \dots, \frac{dm_n(l)}{dl} \right]$$

Para calcularmos então a pertinência de um contorno \mathbf{b} , em função de sua nitidez, teríamos,

$$\mu_{(b)} = f \left[\sum_{i=1}^n \left| \frac{dm_i(l(b))}{dl} \right| \right].$$

Poderíamos entender este resultado como sendo a soma das variações dos atributos ao longo da distancia Δl da largura do contorno \mathbf{b} . Para fins numéricos, o artigo desenvolve uma fórmula aproximando a taxa de variação pela própria diferença entre os atributos. Isto vem a resultar na fórmula:

$$\mu_{(b)} = \frac{1}{2} \left[\sum_{i=1}^n |m_i(l + \Delta l) - m_i(l - \Delta l)| \right]$$

Para o caso de propriedades ordinais, como por exemplo, caracterizações de uma população como "Velha", Jovem, etc, o artigo propõe o cálculo individual de cada conjunto nebuloso("Velha", Jovem, etc), independentemente, usando o mesmo método acima, contudo sem usar o somatório.

Podemos igualmente aplicar este modelo em casos de sobreposição de camadas. Várias formas podem ser utilizadas para a agregação. Uma forma comum seria utilizar

a média, dado que a variação em apenas uma camada é uma mudança ponderada parcial, outra forma seria usar operação AND e achar a variação mínima.

A outra utilização para o modelo é na generalização de polígonos espúrios surgidos pela sobreposição de camadas. Quando camadas são sobrepostas, às vezes, surgem pequenos polígonos que não pertencem a área nenhuma, dessa forma é necessário incorporá-los aos polígonos maiores.

Existem várias referências sobre a questão dos contornos imprecisos (Burrough, Frank, 1996). Leung (1987) propõe a definição dos contornos como zonas de distribuição contínua de propriedades e dividindo em regiões de "núcleo" e contorno onde, respectivamente, a propriedade é precisa e nebulosa.

Capítulo IV

4. Um Sistema de Banco de Dados Nebuloso

O sistema de manipulação de dados nebulosos propõe-se a lidar, de forma nebulosa, com um banco de dados convencional, utilizando o modelo relacional. O módulo projetado trabalha como uma interface, entre a consulta e a modelagem nebulosa de um banco de dados convencional. Ele permitirá a definição do esquema nebuloso espelhado no esquema do banco de dados, possuindo as ferramentas necessárias para definição do modelo relacional nebuloso(Li,Liu,1990, Nola,1995, Berthier,1995). Ele possui operadores básicos da álgebra relacional de forma a possibilitar tanto as operações precisas como nebulosas.

A aplicação do módulo descrito dar-se-á no sistema de consulta de um SIG desenvolvido para a área da saúde(Braga et. al,1994), onde a consulta e a representação dão-se através da base espacial, mas o processamento e os predicados da consulta são feitos sobre a base de dados não gráfica.

A modelagem nebulosa de dados faz-se através de um mapeamento dos atributos das relações do banco de dados em variáveis nebulosas(Figura 4-1). As variáveis são criadas indicando-se o atributo e a relação que ela mapeia.

Após o mapeamento, podemos criar novas variáveis, chamadas *Variáveis Solução*, que são construídas a partir de regras inferidas sobre as variáveis e valores do esquema nebuloso. As regras são construídas utilizando-se proposições e operadores nebulosos(Klir,1995a, Cox,1994a,1994b, Kosko,1993).

O sistema de consulta básico gerará como resultado novas tabelas no banco de dados, fornecendo valores nebulosos, suas pertinências, bem como os valores das *Variáveis Solução*. A saída e representação de dados nebulosos, na base espacial(Jiang,1997a,1997b, Wang,1990, Davidson et. al,1997), são feitas através de visualização dos elementos selecionados, podendo ser, mapas temáticos ou qualquer outra forma de representação que permita mostrar os valores resultantes.

4.1 Arquitetura

O sistema é composto de um módulo de controle que contém um módulo de consulta e um módulo que armazena o esquema nebuloso(Figura 4-2). O módulo de consulta é composto de um *parser* e o sistema de processamento. O *parser* traduz a

linguagem de definição de dados nebulosos e a extensão do SQL, para consulta nebulosa num grupo de comandos básicos do sistema de controle.

O sistema de processamento possui um grupo de comandos para produzir os operadores básicos de projeção, seleção e junção. Ele manipula com os elementos armazenados no esquema nebuloso e envia o resultado das consultas novamente para o SGBD.

O módulo de esquema nebuloso armazena os elementos correspondentes às variáveis nebulosas e as proposições lógicas nebulosas sobre as variáveis nebulosas. A linguagem de consulta utiliza comandos específicos desse módulo, para a criação do esquema. Ela irá definir o modelo nebuloso que será aplicado ao esquema do banco de dados convencional.

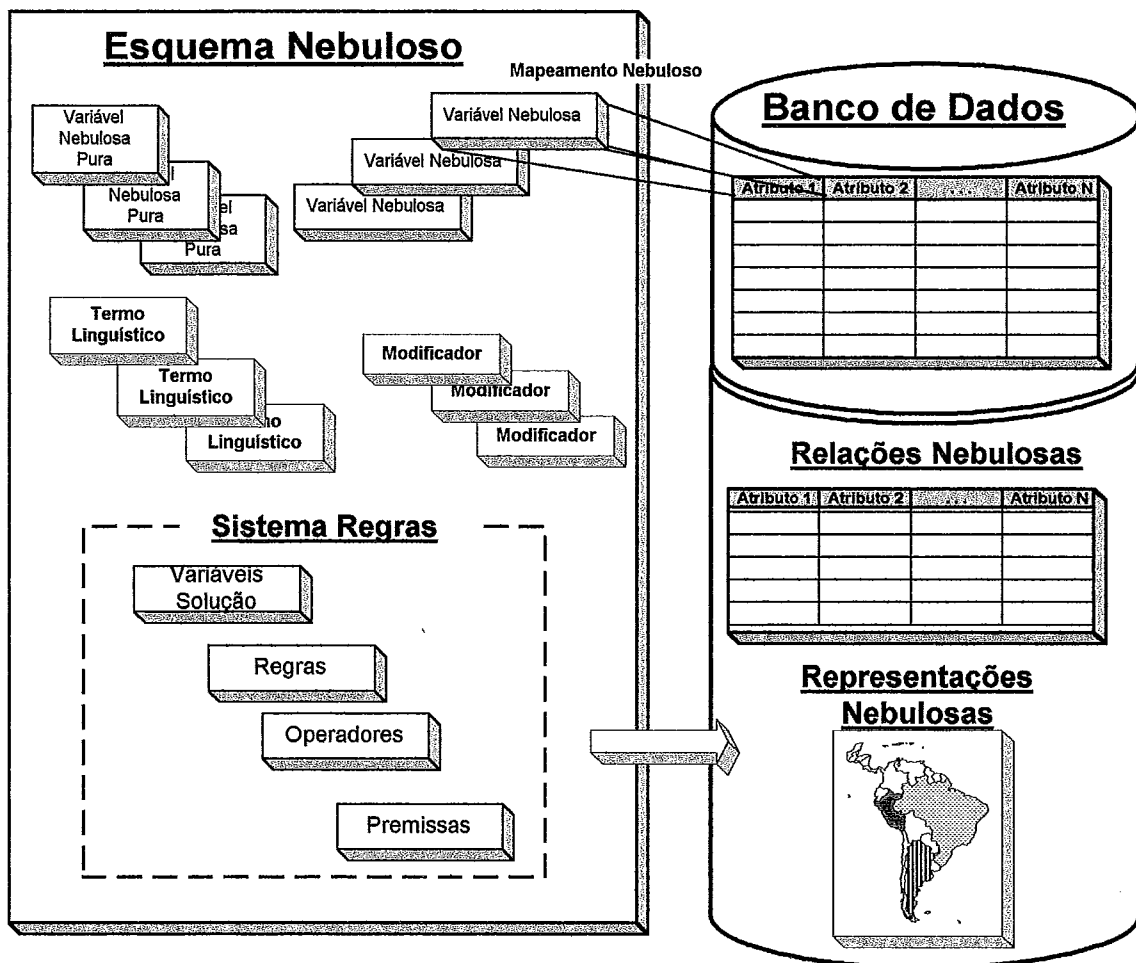


Figura 4-1 - Descrição geral do mapeamento de dados nebuloso

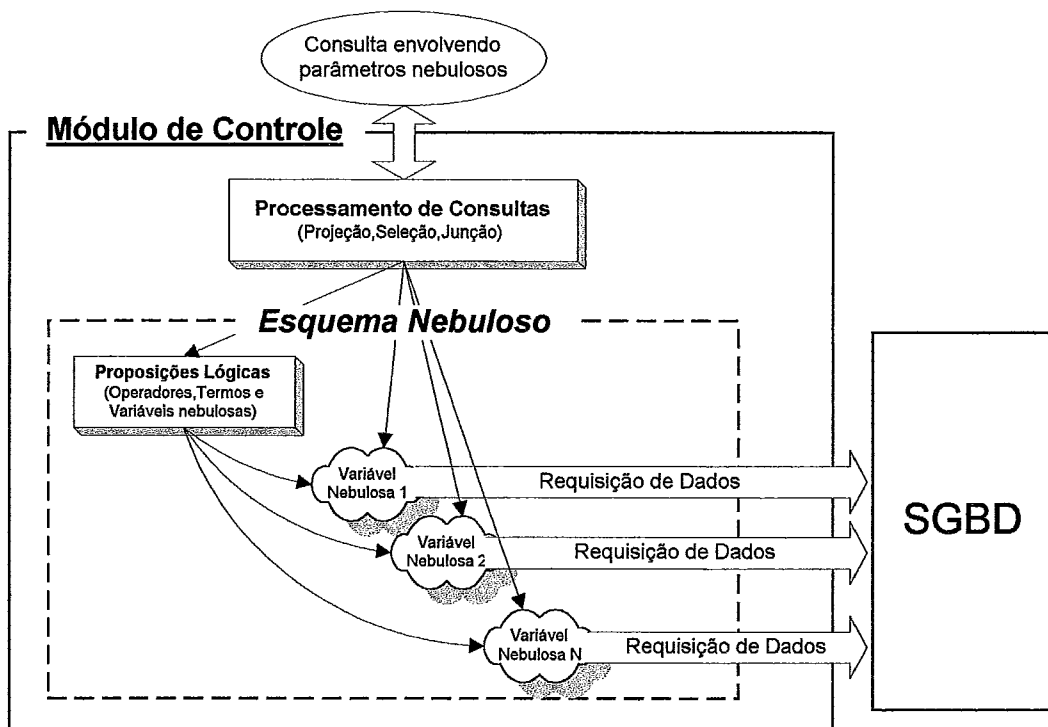


Figura 4-2 - Arquitetura geral do sistema

Os elementos correspondentes às variáveis nebulosas requisitam os dados diretamente do SGBD. Os dados são convertidos, ou *fuzzificados* de acordo com o modelo nebuloso(Figura 4-3).

As variáveis nebulosas são tratadas de forma genérica pelo sistema, desta forma, elas podem ter várias implementações diferentes de acordo com os vários tipos de manipulação desejada para o sistema.

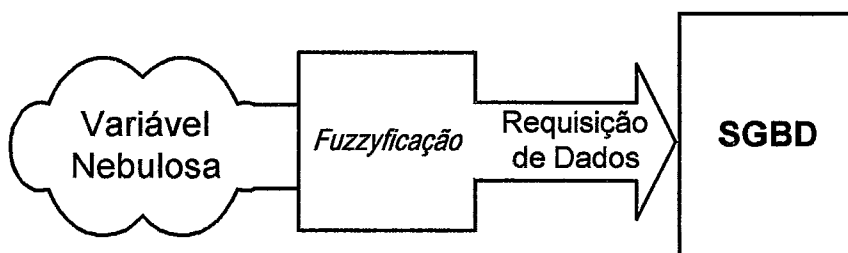


Figura 4-3 - Variável nebulosa

Além da variável nebulosa básica, existem dois tipos de variáveis nebulosas implementados. A primeira(Figura 4-4) é construída através de regras sobre as variáveis mapeadas diretamente sobre o banco. Ela é chamadas de *variável-solução*(Cox,1994a,1994b). A regra é definida por uma proposição nebulosa e um termo inferido à *variável-solução*. Os elementos correspondentes às proposições lógicas são

compostos de operadores nebulosos que são por sua vez compostos das variáveis nebulosas. Eles requisitam os dados para os operadores que recursivamente requisitam os dados para as variáveis nebulosas.

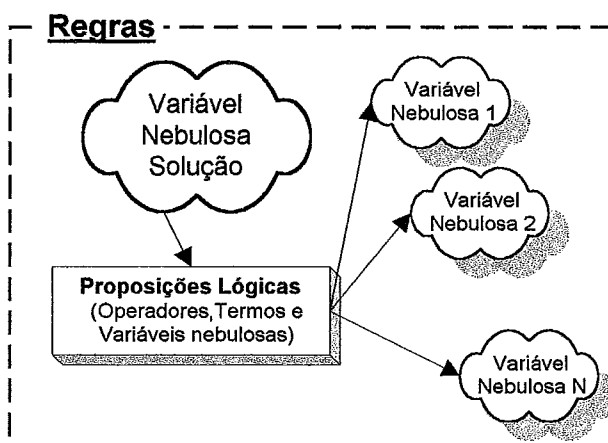


Figura 4-4 - Variável solução

Outro tipo de implementação da variável nebulosa são as variáveis nebulosas puras (Figura 4-5). Este tipo de variável serve para trabalharmos com bancos de dados nebulosos reais. O presente sistema aplicar-se-á em bancos de dados com valores precisos contudo, caso tenhamos um banco de dados com valores nebulosos, este tipo de implementação não fará o processo de *fuzzificação* e colherá os dados nebulosos diretamente do banco.

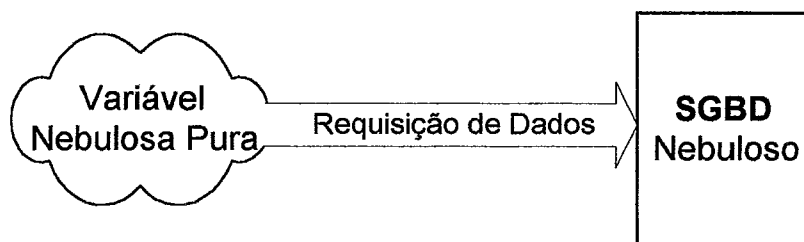


Figura 4-5 - Variável nebulosa pura

A arquitetura do sistema é baseada numa variação das arquiteturas descritas em (Bosc e Pivert, 1997a, 1997b). As primeiras duas arquiteturas propostas sugerem basicamente um processo de projeção sobre o banco de dados baseado nas variáveis envolvidas na consulta nebulosa. No segundo processo é feita uma seleção baseada em intervalos de confiança da consulta. Os dois métodos produzem um conjunto de dados que serão convertidos de acordo com o modelo nebuloso pré-definido.

Na arquitetura proposta, a projeção é feita internamente ao definirmos as variáveis da consulta. A seleção é feita através do cálculo registro a registro das proposições lógicas nebulosas. Desta forma, apesar de não utilizar os recursos do SGBD, temos um controle mais forte sobre os algoritmos e cálculos utilizados. Os algoritmos de cálculo de junção, por exemplo, não podem ser convertidos em consultas por intervalos de confiança.

4.2 Esquema Nebuloso

A implementação e modelo do sistema utilizam o paradigma da orientação por objetos. Cada entidade que irá modelar o esquema do banco de dados convencional de forma nebulosa é composta de uma ou mais classes de objetos(Figura 4-6).

O conjunto de objetos que modela um determinado esquema é armazenado na classe de objetos **Schema**. Os objetos desta classe contém sete tipos de coleções representando as sete classes de objetos que compõe o esquema nebuloso:

- *lingterms* - Coleção de termos lingüísticos utilizados ou a disposição no esquema.
- *interfaces* - Coleção de interfaces de dados utilizadas no esquema.
- *lingvars* - Coleção de variáveis lingüísticas utilizadas no esquema.
- *operators* - Coleção de operadores utilizados ou a disposição do esquema.
- *propositions* - Coleção de proposições utilizadas no esquema.
- *rules* - Coleção de todas as regras utilizadas no esquema.
- *hedges* - Coleção de todos os modificadores utilizados ou a disposição no esquema.

Cada um desses tipos de objetos criados é inserido na coleção correspondente do esquema. Cada elemento possui um nome específico, deste modo, ao ser criado um novo elemento, o sistema verifica se ele já existe. Em caso afirmativo, seus dados podem ser redefinidos mas o objeto não é recriado. Desta forma, podemos reavaliar um esquema modificando os parâmetros de apenas uma entidade dentro do esquema.

Os tipos de objetos: *lingterms*, *operators* e *hedges* podem ser definidos de forma genérica e independente do esquema em si. Em vista disso, uma pequena biblioteca de elementos pode ser posta previamente a disposição do usuário.

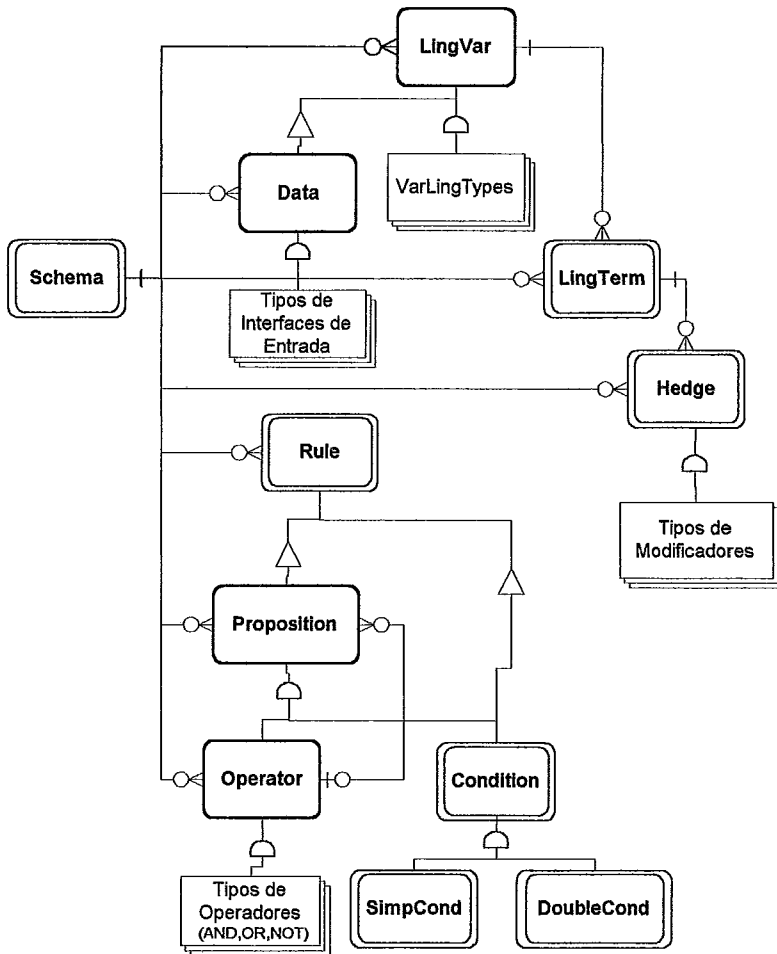


Figura 4-6 - Diagrama de objetos do esquema nebuloso

4.2.1 Elementos Básicos

Dados

Os objetos de dados (Figura 4-7) têm por finalidade acessar os dados dos bancos convencionais que estão sendo modelados pelo esquema nebuloso. Eles estão disponíveis no esquema para todas as variáveis nebulosas.

Eles são compostos de uma classe virtual **Data**. Esta classe age como uma interface para os múltiplos tipos de bancos de dados. Cada tipo de banco diferente possuirá a implementação da classe **Data** de acordo com os processos de acesso e manipulação dos dados. Cada objeto implementado da classe controla um grupo de

atributos ou tabela do banco. O valor de cada atributo individual é requisitado indicando-se o nome do atributo e o registro que está se desejando buscar. A operação de requisição e busca de dados, nos objetos da classe, consiste em selecionar, primeiramente, o registro de acesso e em seguida chamar o método correspondente que retorne o tipo do dado(inteiro, ponto flutuante, data, caracteres, etc.).

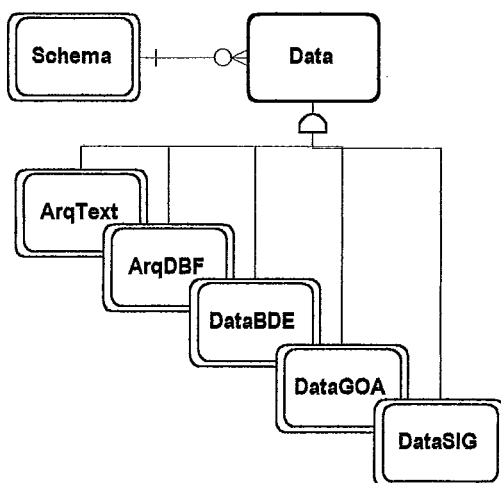


Figura 4-7 - Objetos de interface de dados

A classe virtual básica possui dois atributos:

- *variables* - Consiste de uma coleção das variáveis nebulosas servidas pelo objeto.
- *RecordNb* - Indica o número do registro atualmente sendo acessado pelo objeto.

Existe um grupo de métodos virtuais que são devidamente implementados de acordo com cada tipo de banco acessado em questão:

- **RegNb** - Retorna o registro atualmente em acesso.
- **DataNb** - Retorna o número de registros do banco para o grupo de variáveis sendo controladas em questão.
- **SelectData (i)** - Seleciona o registro *i* do banco como o registro atual.
- **IntData (VarName)** - Retorna como inteiro, se possível, o valor do atributo *VarName* do banco.
- **RealData (VarName)** - Retorna como ponto flutuante, se possível, o valor do atributo *VarName* do banco.

- **StringData (VarName)** - Retorna como cadeia de caracteres, se possível, o valor do atributo **VarName** do banco.
- **RangeInt (VarName, Min, Max)** - Retorna em Min e Max, se possível, o domínio da variável **VarName**. Variáveis inteiras.
- **RangeReal (VarName, Min, Max)** - Retorna em Min e Max, se possível, o domínio da variável **VarName**. Variáveis reais.

Os objetos de interface de dados podem ser implementados de múltiplas formas para atender aos múltiplos tipos de bancos de dados. O presente sistema possui cinco implementações básicas:

- **ArqText** - Uma interface para arquivos-texto simples. Este tipo de arquivo deve conter no seu início uma breve descrição das variáveis da forma:

<Número de variáveis> <Número de registros>

<Tipo da variável 1> <Nome1>

<Tipo da variável 2> <Nome2>

<Tipo da variável n> <Nome n>

<Tipo da Variável> = INTEIRA | REAL | STRING

- **ArqDBF** - Uma interface para arquivos do tipo DBASE.
- **DataBDE** - Uma interface para o BORLAND Database Engine da BORLAND International Inc.
- **DataGoa** - Um interface para o Gerente de Objetos Armazenáveis GOA.
- **DataSIG** - Um interface para o Sistema de Informação Geográfica SIGEPI(Braga et. al,1994).

Variáveis Lingüísticas

A classe de variáveis lingüísticas(Figura 4-8) compõe-se de uma classe virtual **LingVar**. Ela representa as variáveis nebulosas do esquema. Os objetos desta classe

têm por finalidade mapear os atributos do banco de dados convencional em variáveis nebulosas. Esta classe age como uma interface para os diversos tipos de variáveis a serem mapeadas. A implementação de cada tipo de variável está relacionada com seu tipo numérico e o seu comportamento.

Cada variável nebulosa tem uma referência para um objeto do tipo **Data** para acesso ao atributo mapeado. A variável lingüística mapeia os valores convencionais de dados em termos ou valores lingüísticos. Ela possui uma lista dos termos lingüísticos, que a variável nebulosa pode assumir. Para as variáveis que estão sendo mapeadas do seu valor convencional para um valor nebuloso, a pertinência para cada termo lingüístico t_i é calculada pelo valor normalizado *RelativeCrispValue* aplicado à função de cálculo do termo lingüístico ($t_i.MShip(.)$). A coleção de termos lingüísticos $lingterms = \{t_i | i=1, 2..n\}$ pode ser calculada como sendo:

$$\{(t_i, \mu(t_i)) | i=1, 2..n, \mu(t_i) = t_i.MShip(RelativeCrispValue)\}$$

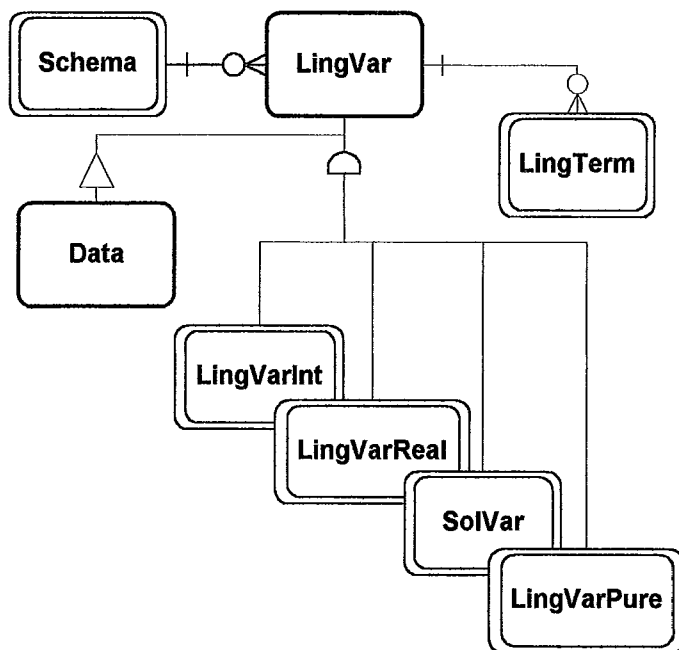


Figura 4-8 - Objetos de variáveis lingüísticas

O sistema implementa três tipos de valores nebulosos(Li,Liu,1990) a serem utilizados em certos casos de desconhecimento dos dados a serem manipulados:

- UNDECIDED ou desconhecidos(Petry,1996) - Este tipo de valor indica que não podemos precisar qual valor nebuloso a ser retornado por uma avaliação.
- UNDEFINED ou inaplicáveis(Petry,1996) - Este tipo indica que não existe valor possível a ser retornado.
- NULL ou Total Ignorância(Petry,1996) - Indica total desconhecimento do atributo, ou seja, ele pode existir e não sabemos qual é ou não ser aplicável.

A classe virtual básica possui os seguintes atributos:

- *DataSource* - Referência para o objeto de dados.
- *DataMember* - Nome da variável dentro da interface de dados.
- *RelativeCrispValue* - Valor preciso da variável em valores relativos: [0,1]
- *lingterms* - Coleção de termos lingüísticos aplicáveis à variável.

Os seguintes métodos virtuais são re-implementados dependendo de cada tipo variável nebulosa:

- **AddTerm(add)** - Acrescenta mais um termo a coleção de termos lingüísticos.
- **SetupData** - Atualiza o valor de *RelativeCrispValue* buscando o dado em *DataSource* e normalizando-o.
- **MembershipTerm(TermName)** - Retorna a pertinência do termo **TermName** dentro da coleção de termos.
- **type** - Retorna o tipo da variável.
- **MaxFuzzyValue(Membership)** - Retorna o termo e a pertinência de maior valor de acordo com o dado atual. Caso tenhamos dois ou mais termos com os maiores valores de pertinência, este método retornará um termo do tipo UNDECIDED ou indeciso, com pertinência de 100%. Caso o dado coletado por **SetupData** esteja fora dos limites do domínio especificado a variável nebulosa assumirá o valor nebuloso UNDEFINED ou indefinido, com valor de pertinência de 100%. Os valores nulos NULL serão retornados quando não houver nenhum termo com pertinência positiva a ser retornado.

As variáveis lingüísticas são implementadas de três formas. A primeira envolve o tipo do atributo a ser mapeado:

- **VarLingInt** - Implementada para mapear números inteiros.
- **VarLingReal** - Implementada para mapear números de ponto flutuante.

O segundo tipo são os objetos da classe **VarSolucao** que envolvem a definição de uma variável cujos dados não são gerados diretamente a partir dos dados, mas através de regras aplicadas às variáveis convencionais. Os termos aplicados a esta variável vêm através das regras a ela aplicadas. O método **SetupData** é reescrito para ativar todas as regras associadas.

Os atributos acrescentados a esta implementação são:

- *DefuzzMethod* - Indica o tipo de método de *defuzzificação* que será utilizado para converter os dados nebulosos em dados precisos.
- *RuleList* - Coleção de regras associadas à variável.

A classe implementa um método especial para *defuzzificação* da variável. Existem dois métodos básicos utilizados: pelo valor de máxima pertinência e pelo centro de massa dos termos nebulosos aplicados à variável(Cox,1994a1994b).

Por último, temos um tipo de variável utilizada para comunicação com banco de dados nebulosos **VarLingPure**. Os objetos desta classe buscam os dados nebulosos diretamente do banco de dados. É assumido que o valor do atributo no banco de dados constará de uma lista de termos lingüísticos com suas respectivas pertinências. Este tipo de objeto não foi utilizado nos testes do protótipo devido à dificuldade de se encontrar bancos de dados nebulosos.

Termos Lingüísticos

Cada termo lingüístico(Figura 4-9) constitui um conjunto nebuloso dentro de um universo normalizado. Ele estabelece uma curva de distribuição de pertinência ao longo de um intervalo dentro da faixa de [0,1]. Esta forma de implementação permite que o termo possa ser usado no domínio de qualquer variável lingüística.

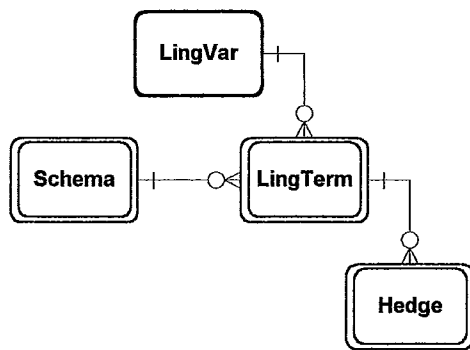


Figura 4-9 - Objetos de termos lingüísticos

Cada termo pode estar associado a um ou mais modificadores através de objetos da classe **Hedge**. Uma vez associado a um modificador o valor de pertinência retornado pelo termo lingüístico é internamente alterado sem ser vista pela variável lingüística que a utilize. O nome do termo lingüístico é acrescentado ao valor do modificador, desta maneira, por exemplo, um termo “Alto” acrescentado de um modificador “Muito” altera o nome do termo para o nome “Muito Alto”.

Os atributos básicos de cada termo lingüístico definem a forma e o tipo da curva utilizada:

- *CurveType* - Indicam o tipo da curva. As curvas implementadas são: triangular, trapezoidal e gaussiana.
- *BegCurve* - Indicam o valor normalizado do início da curva.
- *MidCurve1* - Indicam o valor normalizado do meio da curva gaussiana ou triangular ou o início da base superior da curva trapezoidal.
- *MidCurve2* - Indicam o valor normalizado do meio da curva gaussiana ou triangular ou o fim da base superior da curva trapezoidal.
- *EndCurve* - Indicam o valor normalizado do fim da curva gaussiana, triangular ou trapezoidal.
- *AlfaCut* - Indica o valor de pertinência para o corte alfa do termo em questão. É iniciado com zero.
- *VectorValues* - Contém uma lista que divide igualmente o intervalo normalizado nos valores de pertinência. É utilizado para definir curvas diretamente.
- *hedges* - Lista de modificadores associados ao termo.

Os métodos implementados no termo lingüístico são:

- **name** - Retorna o nome do termo acrescentando o nome dos modificadores no início do mesmo.
- **AddHedge (hedge)** - Adiciona mais um modificador na lista.
- **MShip (RelativeValue)** - Retorna o valor de pertinência correspondente ao valor normalizado **RelativeValue** e aplicando a lista de modificadores e cortes-alfa.

Algoritmo:

MShip (RelativeValue)

RelativeValue = <Função da curva>(RelativeValue)

Para cada modificador h_i faça,

RelativeValue = h_i .**Filter** (**RelativeValue**)

Se **RelativeValue** < *AlfaCut*

RelativeValue = *AlfaCut*

retornar **RelativeValue**

- **MaxMShip** - Retorna o valor de pertinência máxima do termo.
- **Defuzzify (Type)** - Retorna o valor do termo *defuzzificado* em valor normalizado. **Type** indica qual o tipo de *defuzzificação* utilizado, que pode ser pelo valor de máxima pertinência ou pelo centro de massa da curva.

Modificadores(hedges)

Os modificadores(Figura 4-10) têm por finalidade alterar os termos lingüísticos criando termos mais estritos ou mais genéricos, estreitando ou ampliando a distribuição de pertinência do termo lingüístico a eles associados.

Cada modificador possui uma função genérica **filter** a qual entra com um valor de pertinência e retorna o valor modificado. O atributo *Factor* indica o grau de modificação do tipo de modificador.

O sistema implementa os três tipos básicos de modificadores(Cox,1994a,1994b):

- **Dilatador:** O cálculo de filtragem é dado pela fórmula:

$$\mathbf{Filter}(\mathbf{Value}) = \mathbf{Factor} \sqrt{\mathbf{Value}}$$

- **Concentrador:** cálculo de filtragem é dado pela fórmula:

$$\mathbf{Filter(Value)} = \mathbf{Value}^{\mathbf{Factor}}$$

- **Intensificador:**

se $\mathbf{Value} \leq 0.5$ $\mathbf{Filter(Value)} = 2 \times \mathbf{Value}^{\mathbf{Factor}}$

senão $\mathbf{Filter(Value)} = 1.0 - 2 \times (1.0 - \mathbf{Value})^{\mathbf{Factor}}$

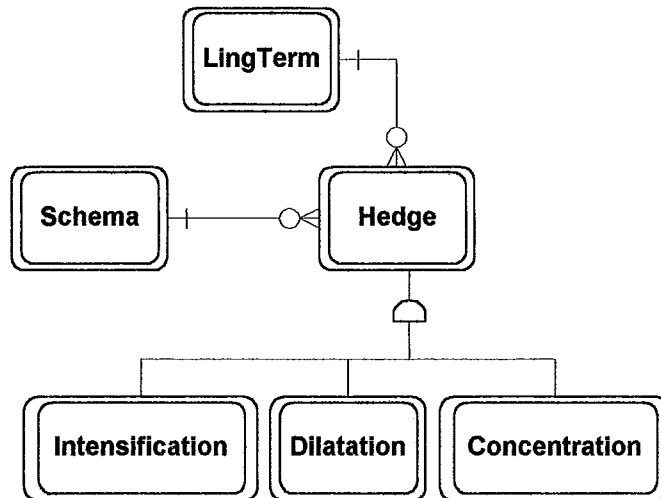


Figura 4-10 - Objetos de modificadores

4.2.2 Regras

O conjunto de classes que forma o sistema de regras do esquema tem por finalidade construir variáveis através de uma inferência de conhecimento. Uma regra é associada a uma variável nebulosa do tipo Variável Solução(Cox,1994a,1994b). Uma regra é composta de uma proposição que pode ser um ou mais conjuntos de condições associadas através de operadores.

Proposições

Classe virtual **Proposition** pode ser implementada como uma condição simples ou conjuntos delas associadas por um ou mais operadores(Figura 4-11). Este processo recursivo flexibiliza a construção de qualquer tipo de proposição(Seção 2.3.3).

A classe não possui atributos, somente métodos virtuais que serão devidamente implementados nas condições ou operadores. Os métodos virtuais implementados são:

- **Active** - Retorna verdadeiro ou falso caso a regra tenha valor de pertinência ≥ 0 . Este processo poderia ser testado diretamente pelo valor de pertinência da premissa, contudo a ativação da proposição pode depender de outros fatores além do valor de pertinência.
- **MShip** - Retorna o valor de pertinência da proposição.
- **SetupData** - Atualiza os dados utilizados na proposição. Em última instância, atualiza os dados das variáveis constituintes das condições.

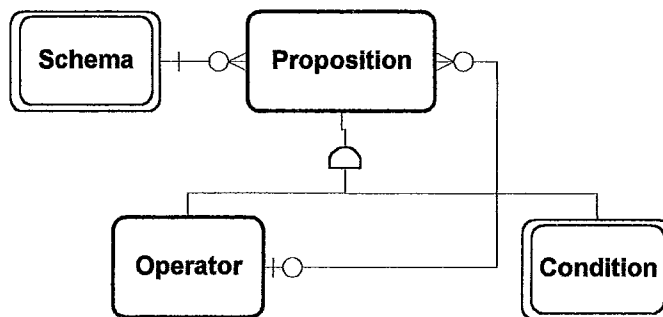


Figura 4-11 - Objetos de proposições

Condições

As condições são objetos especializados de uma classe **Proposition** que relacionam uma variável e um termo lingüístico ou duas variáveis, como por exemplo:

- Temperatura = Alto
- Índice Mortalidade = Índice Natalidade
- Etc.

As condições implementadas no sistema(Figura 4-13) consideram somente a igualdade nebulosa, as inequidades podem ser igualmente implementadas. As condições que relacionam uma variável e um termo são chamadas de condições simples(**SimpCond**) e as que relacionam duas variáveis são chamadas de condições duplas(**DoubleCond**). A igualdade nebulosa simples é feita (Tabela 2.1 da seção 2.2.9) entre o valor preciso coletado para a variável e o termo nebuloso, resultando portanto na pertinência do valor preciso ao termo nebuloso. A igualdade entre duas variáveis será feita através de uma tabela de similaridade entre os termos nebulosos. Esta tabela será levantada usando distâncias semânticas(seção 2.2.9).

Os métodos virtuais da classe **Proposition** são implementadas como se segue:

- **Active** - Para condições simples, retorna verdadeiro caso a pertinência do termo dentro da variável seja ≥ 0 . Para condições duplas, retorna verdadeiro para a pertinência agregada resultante da igualdade $\text{for} \geq 0$.
- **MShip** - Para condições simples, retorna o valor de pertinência do termo na variável. Para condições duplas, retorna o valor agregado da igualdade.
- **SetupData** - Chama os respectivos métodos de atualização de dados das variáveis envolvidas nas condições.

Operadores

Os operadores(Figura 4-12) têm por finalidade realizar operações lógicas nebulosas entre proposições. A classe virtual **Operator** serve de base para implementação de qualquer tipo de operador. Na literatura, muitos tipos de operadores são propostos. Os implementados aqui são os propostos por Zadeh(1965) e indicados na seção 2.3.2. A diferença na implementação de cada operador está no processo de cálculo do valor de pertinência, a partir do método **MShip**.

Um operador associa proposições que podem ser outros operadores ou condições formando uma estrutura em árvore como na Figura 4-14.

Regras

A classe de tratamento de regras(Figura 4-15) visa inferir um conhecimento sobre uma variável em função de uma associação lógica nebulosa entre outras variáveis. Os objetos desta classe representam proposições condicionais não qualificadas(seção 2.3.2) e são associadas a variáveis solução(Cox,1994a,1994b) através das regras de correlação mostradas na seção 2.3.3.

Uma variável solução pode conter várias regras. Cada regra infere um termo lingüístico à variável solução. A pertinência do termo na variável, para um certo registro de dados, é feita através da regra de correlação, onde cada regra possui uma indicação do tipo de correlação que será usada. O sistema permite flexibilidade na implementação de vários tipos de correlação. A correlação implementada no protótipo foi a min-max:

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

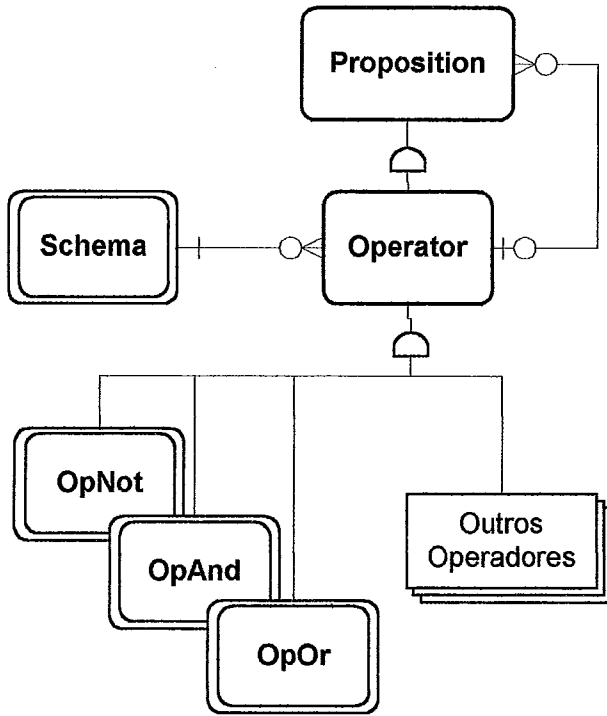


Figura 4-12 - Objetos de operadores

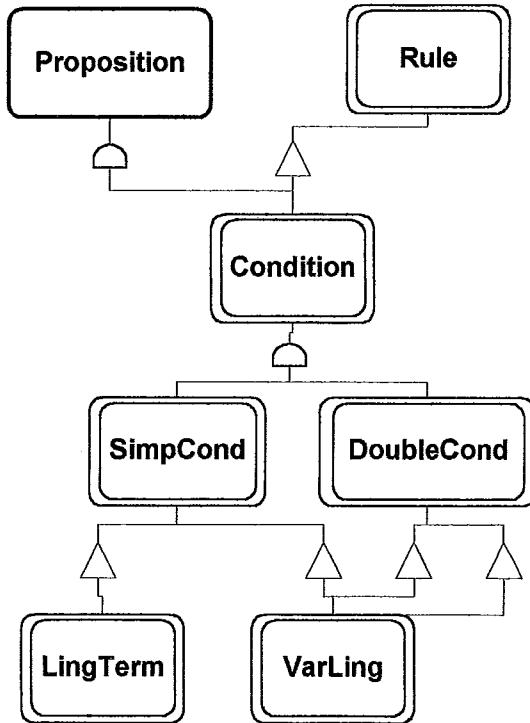


Figura 4-13 - Objetos de condições

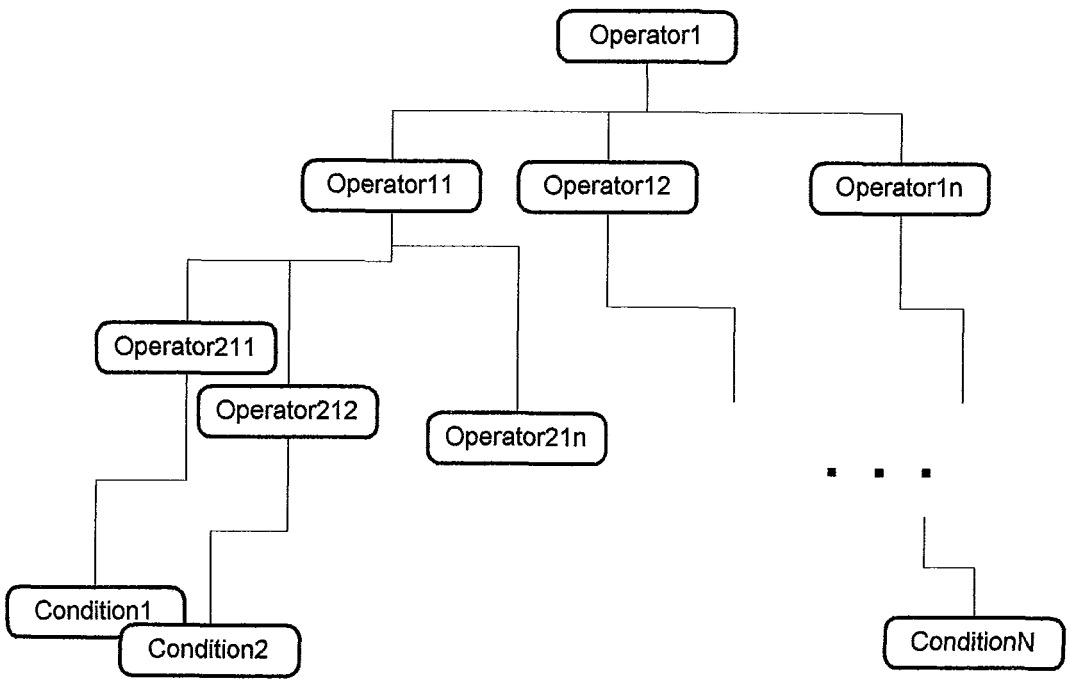


Figura 4-14 - Estrutura genérica de uma proposição

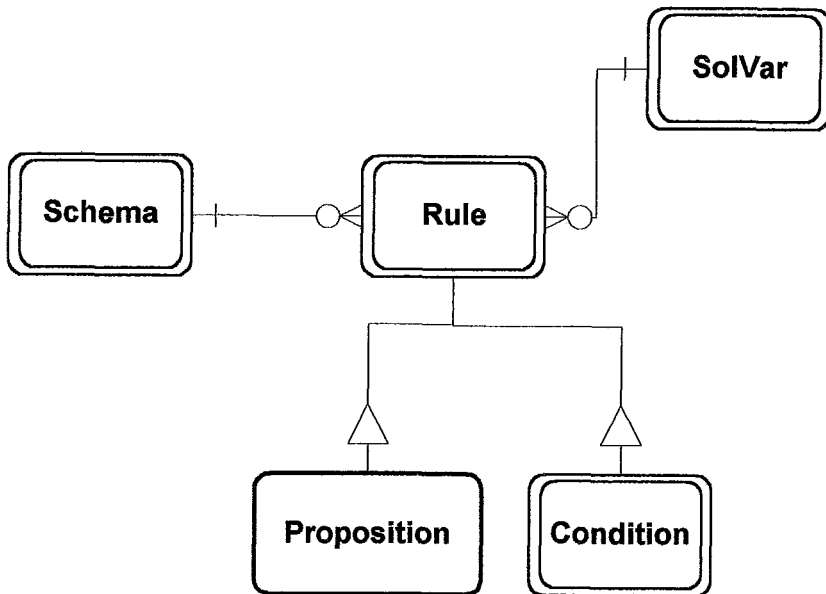


Figura 4-15 - Objetos de regras

4.3 Sistema de Controle

O módulo de controle é o responsável pela execução das consultas ou banco de dados e a geração dos resultados de saída. A classe de controle é especializada da classe

Schema(Figura 4-16). Ela possui dois atributos básicos: a coleção de dados de saída e a coleção de comandos.

Os comandos são agregados numa classe **FuzzyMacro**. Cada grupo de comandos criado é armazenada na coleção de comandos para posterior execução. Os comandos permitem a definição do modelo e a realização de consultas.

Um processador de linguagem externo pode gerar o arquivo de comandos permitindo assim que um usuário externo construa uma linguagem própria e adequada as suas necessidades. Além disso, vários processos de otimização podem ser implementados e testados.

O sistema contém ainda um *Parser* interno, utilizando uma extensão nebulosa do SQL, que interpreta uma linguagem de definição de dados nebulosos e execução de consultas com variáveis nebulosas.

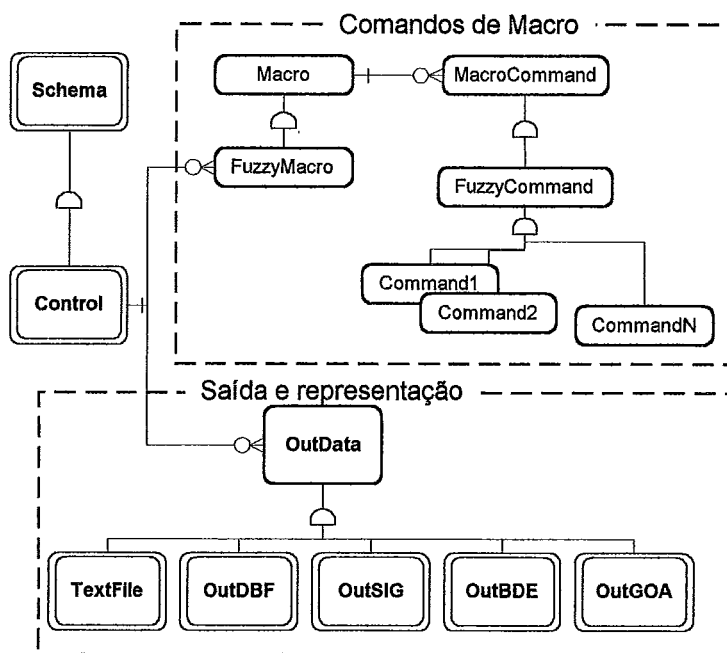


Figura 4-16 - Sistema de controle

4.3.1 Saída e representação

O objeto de saída **OutData**(Figura 4-16) é uma classe virtual que contém métodos genéricos de saída, que serão implementados diferentemente em cada tipo de banco de dados:

- **send(.)** - Vários métodos isomórficos a este permitem ao objeto enviar todos os tipos de informações manipuladas pelo sistema.

- **Commit** - Confirma o envio de um ou mais blocos de dados.

Foram implementados 3 tipos de saídas:

- **TextFile** - Os objetos desta classe são implementados somente para saída em arquivo texto. O formato é indicado na seção 4.2.1.
- **OutDBF** - Os objetos desta classe são implementados somente para saída em arquivo DBASE.
- **OutBDE** - Este objeto de saída é utilizado para conexão com o BORLAND Database Engine da BORLAND International Inc.
- **OutSIG** - Os objetos desta classe são utilizados para executar saídas para o Sistema de Informação Geográfica SIGEPI(Braga et. al,1994).
- **OutGOA** - Os objetos desta classe são utilizados para executar saídas para o Gerente de Objetos Armazenados em desenvolvimento no programa de engenharia de sistemas e computação(PESC) da COPPE/UFRJ.

4.3.2 Processamento de Consultas

O processamento de consultas do sistema faz-se primeiramente definindo-se o modelo nebuloso sobre o banco de dados convencional. Em segundo lugar, as consultas propriamente ditas podem ser feitas e executadas. Este procedimento pode ser feito através de um arquivo de comandos, através de chamadas diretas de métodos da classe **Schema** e **Control** ou utilizando a linguagem de definição e consulta nebulosas desenvolvida(Capítulo 6). Os comandos de projeção, seleção e junção descritos aqui constituem uma primeira versão utilizada no protótipo. Outras implementações são possíveis utilizando melhor os recursos do banco de dados.

Comandos de definição do modelo

Os comandos a seguir, métodos da classe **Schema**, podem ser invocados em qualquer ordem contudo retornarão uma condição de erro caso alguma inconsistência seja encontrada. Apesar do usuário não necessitar, todos os componentes criados podem possuir um nome próprio, o qual será usado para referenciá-lo dentro da sua coleção específica. Cada componente criado é automaticamente posto em sua coleção, dentro do seu esquema. Esta lista de comandos é expansível. Todos estes comandos podem ser igualmente chamados a partir de comandos de um arquivo texto:

- **CreateVarInt** (*name, min, max, data, DataVarName*) - Cria uma variável lingüística para mapear números inteiros. *min* e *max* representam os valores mínimo e máximo do domínio, respectivamente. O nome *data* deve ser de um objeto de interface de dados já criado e *DataVarName* é o nome da variável dentro da interface de dados.
- **CreateVarReal** (*name, min, max, data, DataVarName*) - Cria uma variável lingüística para mapear números de ponto flutuante. *min* e *max* representam os valores mínimo e máximo do domínio, respectivamente. O nome *data* deve ser de um objeto de interface de dados já criado e *DataVarName* é o nome da variável dentro da interface de dados.
- **CreateSolInt** (*name, min, max, data, DataVarName*) - Cria uma variável solução para mapear números inteiros. *min* e *max* representam os valores mínimo e máximo do domínio, respectivamente. O objeto de interface de dados e o nome da variável são opcionais e podem ser usados para trocar a interface de dados original utilizada nas variáveis das regras.
- **CreateSolReal** (*name, min, max, data, DataVarName*) - Cria uma variável solução para mapear números de ponto flutuante inteiros. *min* e *max* representam os valores mínimo e máximo do domínio, respectivamente. O objeto de interface de dados e o nome da variável são opcionais e podem ser usados para trocar a interface de dados original utilizada nas variáveis das regras.
- **CreateTerm** (*name, Init, Mid, End, type*) - Cria um termo lingüístico com curva definida por *type*. Os valores do início, meio e fim da curva são dados em forma normalizada.
- **CreateTerm2** (*name, Init, Mid1, Mid2, End, type*) - Cria um termo lingüístico com curva definida por *type*. Esta é uma forma alternativa para poder criar curvas trapezoidais onde *Mid1* e *Mid2* representam início e fim da base superior do trapézio.
- **CreateTermHedge** (*hedge, Term*) - Cria um novo termo lingüístico que é composto da associação do modificador ao termo.
- **CreateDataText** (*name, file*) - Cria uma interface de dados do tipo arquivo texto associada ao arquivo *file*.

- **CreateCondition** (*name, lingvar, term*) - Cria uma condição simples do tipo $\langle \text{lingvar} \rangle = \langle \text{term} \rangle$.
- **CreateOpBin** (*name, type, Prop1, Prop2*) - Método utilizado para criar operadores entre duas condições ou proposições previamente criadas. O nome *type* indica o tipo de operador e *Prop1* e *Prop2* indicam o nome de duas proposições previamente criadas. Ao final armazena como **Proposition** na coleção de proposições.
- **CreateOpUna** (*name, type, Prop1*) - O mesmo que o método anterior para operadores unários, como por exemplo, “NOT”.
- **CreateRule** (*name, prop, cond, corrtype*) - Cria uma regra associando à proposição *prop* e a condição *cond* pelo método de correlação *corrtype*.
- **CreateHedge** (*name, type, factor*) - Cria um modificador do tipo *type* e fator *factor*.
- **UseTerm** (*Term, Var*) - Associa o termo *Term* à variável *Var*. A variável e o termo devem existir previamente.
- **UseHedge** (*hedge, Term*) - Associa o modificador *hedge* ao termo *Term*. O termo e o modificador devem ter sido criados anteriormente.
- **UseRule** (*rule, Var*) - Associa a regra *rule* à variável *Var*. Ambas devem ter sido criadas previamente.

Projeção

A operação de projeção é feita pelos métodos abaixo. Cada interface de dados possui a lista das variáveis a que atende. O algoritmo ativa ou desativa as variáveis para executar a projeção, em seguida, envia as variáveis para a saída. A operação de determinação das variáveis antes de enviar para a saída é utilizada em todas as operações de consulta, uma vez que, a operação de projeção é implícita em todas as consultas.

- **Validate** (*dataname, VarName*) - Valida a variável *VarName* em *dataname*.

- ***Invalidate (dataname, VarName)*** - Invalida a variável ***VarName*** em ***dataname***.
- ***Projec (dataname, OutputTbl)*** - Faz a operação de projeção enviando a saída para a interface de saída ***OutputTbl***.

Algoritmo:

Projeção (dataname, x_1, x_2, \dots, x_k , OutputTbl)

Para $i = 1$ até n , <i>Invalidate (dataname, v_i)</i>	Determinação das variáveis envolvidas
Para $i = 1$ até k , <i>Validate (dataname, x_i)</i>	

Para $i = 1$ até <i>dataname.DataNb</i> , <i>dataname.SelectData (i)</i>	Número de registros.
---	----------------------

Para $j = 1$ até <i>dataname.VarNb</i> , Se <i>dataname.Var (j)</i> válida <i>OutputTbl.send (dataname.Var (j))</i>	Número de variáveis.
--	----------------------

<i>OutputTbl.Commit</i>	Confirma envio de 1 registro.
--------------------------------	-------------------------------

Seleção

A determinação das variáveis envolvidas ou a projeção implícita é incluída na operação. O processo consiste em ativar o método ***.SetupData*** da proposição. Ele ativa recursivamente os métodos virtuais dos operadores e condições, como da árvore da Figura 4-14, até ativar o método correspondente das variáveis que são as folhas da árvore onde as mesmas buscam os dados na interface de dados. Os métodos ***.Active*** e ***.MShip*** podem então ser requisitados. Os dois chamam os métodos recursivos da árvore calculando a pertinência final da proposição.

A operação de seleção é feita pelo método abaixo. A projeção deve ter sido feita anteriormente. O método entra com os nomes, dentro das coleções, da interface de dados, da proposição e da interface de saída.

- *Select (dataname, Prop, OutputTbl)*

Algoritmo:

Seleção(*dataname*, x_1, x_2, \dots, x_k , *Prop*, *OutputTbl*)

Para $i = 1$ até n , <i>Invalidate (dataname, v_i)</i> Para $i = 1$ até k , <i>Validate (dataname, x_i)</i>	Determinação das variáveis envolvidas
Para $i = 1$ até <i>dataname.DataNb</i> , <i>dataname. SelectData (i)</i>	Número de registros
<i>Prop. SetupData</i>	Atualiza os dados de todas as variáveis envolvidas na proposição
Se <i>Prop.Active</i>	Verifica se a pertinência calculada final da proposição atende à pertinência mínima
<i>OutputTbl. send (Prop.MShip)</i>	Envia a pertinência da proposição
Para $j = 1$ até <i>dataname.VarNb</i> Se <i>dataname.Var (j)</i> válida <i>OutputTbl. send (dataname.Var (j))</i>	Número de variáveis.
<i>OutputTbl. Commit</i>	Confirma envio de 1 registro.

Junção

O algoritmo de junção está baseado num produto cartesiano e uma projeção. A igualdade entre os atributos de junção é feita de forma nebulosa (Petry, 1996). O algoritmo proposto faz a projeção da mesma forma que nos métodos anteriores. Dentro do produto cartesiano entre as duas relações ou tabelas, o sistema avalia uma condição dupla (seção 4.2.2) entre os atributos de junção. Após a avaliação, o sistema envia a tabela de junção para a interface de saída. Diferente da junção convencional, os valores

dos atributos de junção podem não ter os mesmos valores precisos e nem nebulosos, desta forma o sistema envia o valor dos dois atributos para a saída.

- *Join (dataname1, dataname2, x₁, x₂, OutputTbl)*

Algoritmo:

Para i = 1 até n, <i>Invalidate (dataname, v_i)</i> <i>Validate (dataname, x₁)</i> <i>Validate (dataname, x₂)</i>	Determinação das variáveis envolvidas
<i>Prop</i> ← <i>x₁ = x₂</i>	Constrói a proposição de junção.
Para i = 1 até <i>dataname1.DataNb</i> , <i>dataname1.SelectData (i)</i>	Número de registros da relação 1.
Para j = 1 até <i>dataname2.DataNb</i> , <i>dataname2.SelectData (j)</i>	Número de registros da relação 2.
<i>Prop.SetupData</i>	Atualiza os dados de todas as variáveis envolvidas na proposição.
Se <i>Prop.Active</i>	Verifica se a pertinência calculada final da proposição atende à pertinência mínima.
<i>OutputTbl.send (Prop.MShip)</i>	Envia a pertinência da proposição.
Para k = 1 até <i>dataname1.VarNb</i> Se <i>dataname1.Var (j)</i> válida <i>OutputTbl.send (dataname1.Var (k))</i>	Número de variáveis.
Para k = 1 até <i>dataname2.VarNb</i> Se <i>dataname2.Var (j)</i> válida <i>OutputTbl.send (dataname2.Var (k))</i>	Número de variáveis.
<i>OutputTbl.Commit</i>	Confirma envio de 1 registro.

4.4 Utilização e expansões

O sistema descrito pode ser utilizando de duas formas. A primeira é independentemente como um processador nebuloso de modelagem e consulta de bancos de dados convencionais. A segunda forma é incorporada internamente a algum sistema

ou *software* de banco de dados. Nesta forma, ele pode funcionar como uma biblioteca ou como um processador de consultas. Em ambas, o sistema fornece uma linguagem de consulta e definição para a sua utilização, ela será descrita no capítulo 6.

O método de conversão de dados convencionais implementado pode ser estendido para outros tipos de dados, como datas, possibilitando a implementação de manipulações temporais ou texto, possibilitando avaliar a semelhança de nomes. A utilização do sistema em bancos de dados espaciais poderá ser efetuada implementando-se objetos para a conversão de dados espaciais convencionais em nebulosos. Um modelo para esta tarefa será proposto no próximo capítulo.

Capítulo V

5. Manipulação Espacial Nebulosa

A manipulação de incertezas em Sistemas de Informação Geográfica é interessante e necessária, não somente relativo a bancos de dados(Seção 2.4) mas também no que se refere a incerteza da posição dos dados no espaço(seção 3.6.3). Existem muitos casos onde desejamos por exemplo, “as áreas de alta ocorrência de malária próximo a rios ou córregos”.

Para possibilitar consultas deste porte devemos ter um sistema capaz de modelar e manipular as incertezas do banco de dados bem como as da base de dados espaciais. É necessária a definição de um modelo para os objetos espaciais e um sistema capaz de lidar com as consultas espaciais levando em conta as incertezas. Construiremos o modelo de incerteza agregando às propriedades das unidades espaciais básicas. Propomos a construção do sistema de modelagem e consulta numa extensão do descrito no capítulo 4.

Na seção 5.1 descreveremos o modelo proposto e na seção 5.2 mostraremos a aplicação do modelo para as primitivas espaciais básicas.

5.1 Modelagem Espacial

As bases de dados espaciais são construídas sobre o conceito de entidades ou feições espaciais(seção 3.2.1). No modelo orientado por objetos(seção 3.2.3) usamos o conceito de objeto espacial. Ele tem três caracterizações básicas: existencial ou temática, extensional e geométrica(Molenaar,1996). As características extensionais e geométricas irão nos mostrar a descrição do objeto no espaço, enquanto que a característica existencial ou temática nos dará a semântica do objeto.

5.1.1 Modelagem geométrica e extensional

Os Sistemas de Informação Geográfica constroem feições espaciais em dois tipos de formatos básicos, Raster e o Vetorial. No modelo Raster, todas as feições ou objetos são construídos a partir de entidades atômicas que são as células. No modelo Vetorial, eles são construídos a partir de primitivas espaciais, como pontos e segmentos de reta.

Definimos aqui um objeto espacial genérico a partir de elementos atômicos, representados por células no modelo Raster, ou pontos no modelo Vetorial. Cada elemento atômico é representado por um ponto $p = \{ (x_1, x_2, \dots, x_n) \mid p \in \mathfrak{R}^n \}$ onde x_1, x_2, \dots, x_n representam as n componentes do vetor que definem p no espaço \mathfrak{R}^n . Por simplificação nos propósitos deste trabalho, assumiremos o espaço bi-dimensional, onde $p = (x, y)$. Utilizaremos esta entidade espacial atômica para definir a menor unidade de incerteza de um objeto espacial.

Um conjunto de objetos espaciais \mathbf{M} no espaço \mathfrak{R}^n é definido por:

$$\mathbf{M} = \{ m_j \mid m_j \subseteq \mathbf{P}, j = 1, 2, 3, \dots \} \text{ onde,}$$

- $\mathbf{P} \subseteq \mathfrak{R}^n$ é um domínio espacial representando uma porção do espaço, como por exemplo um mapa,
- $m_j = \{ p_i \mid p_i \in \mathbf{P}, i = 1, 2, 3, \dots \}$ é um objeto espacial composto de pontos p_i do domínio \mathbf{P} , cuja função característica $\mu_{m_j}(p_i) = 1$, se $p_i \in m_j$ e $\mu_{m_j}(p_i) = 0$, se $p_i \notin m_j$ e
- $p_i = \{ (x_1, x_2, \dots, x_n) \mid p \in \mathbf{P} \}$ é um elemento do domínio pertencente ou não a algum objeto m_j .

O conjunto \mathbf{M} é o conjunto de subconjuntos m_j contidos no domínio espacial \mathbf{P} .

Este conjunto é chamado de conjunto potência de \mathbf{P} , escrito como:

$$\mathbf{M} = \mathcal{P}(\mathbf{P}) = \{ m_j \mid m_j \in \mathbf{M}, i = 1, 2, 3, \dots, n \}$$

O conjunto de pontos $p_i \in m_j$ irá caracterizar a geometria do objeto m_j , a função característica $\mu_{m_j}(p_i)$ irá definir as características extensional e existencial. Se tivermos, por exemplo, um objeto espacial m num domínio $\mathbf{P} \subseteq \mathfrak{R}^n$, representado por um ponto P_0 no espaço:

$$m = \{ p \mid p \in \mathbf{P}, \mu_m(p) = 1 \text{ se } p = P_0 \text{ e } \mu_m(p) = 0, \text{ caso contrário} \}$$

O ponto P_0 aqui caracteriza a geometria do objeto, a função característica indica sua existência. No caso de um círculo no espaço, por exemplo, a função característica

indicará a extensão do círculo. Supondo um círculo centrado em P_0 e de raio R , o objeto m seria:

$$m = \{ p \mid p \in \mathbf{P}, \mu_m(p) = 1 \text{ se } \text{distância}(P_0, R) \leq R \text{ e } \mu_m(p) = 0, \text{ caso contrário} \}$$

5.1.2 Modelagem Semântica

A característica existencial ou temática define a semântica do objeto. Ela será determinada de acordo com as variáveis não gráficas associadas a ele (seção 3.2.1).

Dado um banco de dados qualquer DB e um elemento x_1 do banco associado a um objeto $m_j \in \mathbf{M}$, onde \mathbf{M} é um conjunto de objetos espaciais, podemos definir uma função h tal que:

$$h: \mathbf{M} \times \text{DB}$$

$$x_1 = h(m_j): m_j \rightarrow x_1$$

O elemento x_1 caracteriza uma propriedade do objeto m_j . A função h retorna as propriedades que definirão, entre outras coisas, a semântica do objeto.

Exemplo:

Suponhamos um conjunto de objetos \mathbf{M} (Figura 5-1) de um domínio \mathbf{P} representando os pontos de um mapa. Cada objeto corresponde a um ponto numa região do espaço. Uma relação $R = \langle j, \text{Tipo} \rangle$ define qual o tipo de ponto está sendo representado para cada objeto m_j . Suponhamos que se deseje representar os pontos cuja propriedade $h(m_j). \text{Tipo} = \text{“Fazendas”}$.

Relação R

j	Tipo
1	Fazenda
2	Sítio
3	Cidade

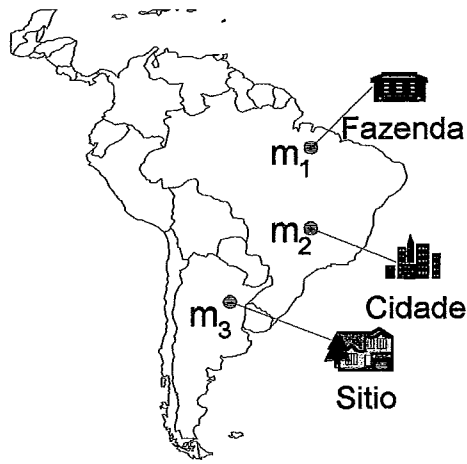


Figura 5-1 - Domínio Espacial P

A restrição definirá um subconjunto $\mathbf{M}' \subseteq \mathbf{M}$ tal que,

$$\mathbf{M}' = \{ m_j \mid m_j \in \mathbf{M}' \text{ e } h(m_j). \text{Tipo} = \text{“Fazendas, } j = 1,2,3\dots \}$$

A definição do objeto por sua função característica fica:

$$m_j = \{ p \mid p \in \mathbf{P}, \mu_{m_j}(p) = 1 \text{ se } p = P_j \text{ e } h(m_j). \text{Tipo} = \text{“Fazendas,} \\ \mu_{m_j}(p) = 0, \text{ caso contrário} \}$$

Para o exemplo em questão temos que:

$$\mathbf{M}' = \{ m_1 \}$$

Vários tipos de propriedades podem ser associadas aos objetos espaciais. Eles podem indicar valores, faixas ou identificação nominal (seção 3.2.1 - Tabela 3.3). Indicações de valores ou identificação nominal em geral surgem como formas de representação dos objetos espaciais, como mapas cromáticos ou de padrões.

5.1.3 Objetos Espaciais Nebulosos

A posição e características geométricas das feições ou objetos espaciais nem sempre são precisas e levam em conta vários tipos de incertezas. O modelo de objetos espaciais pode ser estendido pelo conceito de conjuntos nebulosos. Neste modelo a existência do objeto é vista de forma nebulosa, ou seja, sua extensão ou existência possui graus de relevância ao longo do espaço, como proposto por (seção 3.6.3).

Os graus de relevância serão determinados pelas incertezas associadas aos objetos. As incertezas podem ser de ordem posicional referindo-se às imprecisões geradas na obtenção ou definição da forma tamanho e extensão dos objetos. As incertezas oriundas dos dados não gráficos, associados a cada objeto espacial, são aquelas que inferem sua imprecisão através dos valores de seus atributos, como por exemplo, valores de população com imprecisão associados a localidades de um mapa.

No modelo básico, os pontos ou unidades atômicas do espaço eram descritos em termos da sua pertinência aos objetos espaciais. Na extensão nebulosa, a função característica μ passa a ter valores intermediários entre 0 e 1, indicando desta forma pontos que podem pertencer parcialmente ao objeto. Os valores da função característica podem indicar a pertinência do ponto ao objeto ou a concentração de algum atributo do banco de dados, similarmente ao trabalho de Altman(1994) relativo a feições utilizando o modelo Raster.

Vamos considerar um objeto nebuloso O composto de pontos $p_i \in \mathbf{P}$ do espaço, onde,

- $\mathbf{P} \subseteq \mathfrak{R}^n$ é um domínio espacial,
- $O_j = \{(p_i, \mu_{O_j}(p_i)) \mid O_j \in \mathbf{S}, p_i \in \mathbf{P}, \mu_{O_j}(p_i) > 0 \text{ e } i = 1,2,\dots,n\}$ é um objeto espacial nebuloso composto de pontos p_i do domínio \mathbf{P} cuja pertinência $\mu_{O_j}(p_i) \geq 0$, se $p_i \in O_j$ e $\mu_{O_j}(p_i) < 0$, caso contrario e
- $\mathbf{S} = \{ O_j \mid O_j \subseteq \mathbf{P}, j = 1,2,3,\dots,n \}$ é o conjunto de subconjuntos O_j contidos de forma nebulosa no domínio espacial \mathbf{P} . Este conjunto é chamado de conjunto potência nebulosa de \mathbf{P} , escrito como:

$$\mathbf{S} = \mathcal{F}(\mathbf{P}) = \{ O_j \mid O_j \in \mathbf{S}, O_j \subseteq \mathbf{P} \text{ e } j = 1,2,3,\dots,n \}$$

A função característica $\mu_{O_j}(p_i)$ devolve os valores de pertinência de cada ponto em relação ao objeto. Um objeto espacial nebuloso é um conjunto nebuloso de pontos definido por pares $(p_i, \mu_{O_j}(p_i))$ que associam um ponto e um valor pertinência ao objeto. A função μ_{O_j} pode ser discreta, como por exemplo no modelo Raster, indicando a pertinência de cada célula, ou uma função definida em \mathfrak{R}^n (Dutta, 1989).

A existência do objeto pode estar vinculada a um nível de precisão qualquer α , onde definimos o valor de pertinência mínima. A notação O^α , usada de forma análoga para cortes-alfa (seção 2.2.2) é definida por:

$$O^\alpha = \{(p_i, \mu_d(p_i)) \mid \mu_d(p_i) > \alpha, i = 1, 2, \dots, n\}$$

O valor de pertinência mínima especifica o índice de confiança da extensão do objeto. O valor α também pode ser usado para especificar níveis de detalhe de uma camada ou grupo de objetos, onde para atributos específicos, não nos interessam visualizar pontos do espaço com concentração menor que determinado valor.

Exemplo:

Podemos ver na Figura 5-2 o exemplo de um objeto nebuloso O . A função de pertinência μ , neste caso, é uma função da distância de um ponto qualquer p ao ponto de referência p_0 . O objeto seria definido por:

$$O = \{(p, \mu_d(p)) \mid \mu_d(p) \geq 0\} \text{ onde,}$$

- $\mu_d(p) = 1 - d(p_0, p)/K$, se $d(p_0, p) < K$ e $\mu(p) = 0$, caso contrário.
- $d(p_0, p)$ - é a distância de um ponto qualquer p a p_0
- O fator K indica a distribuição da pertinência de O ao longo do espaço.

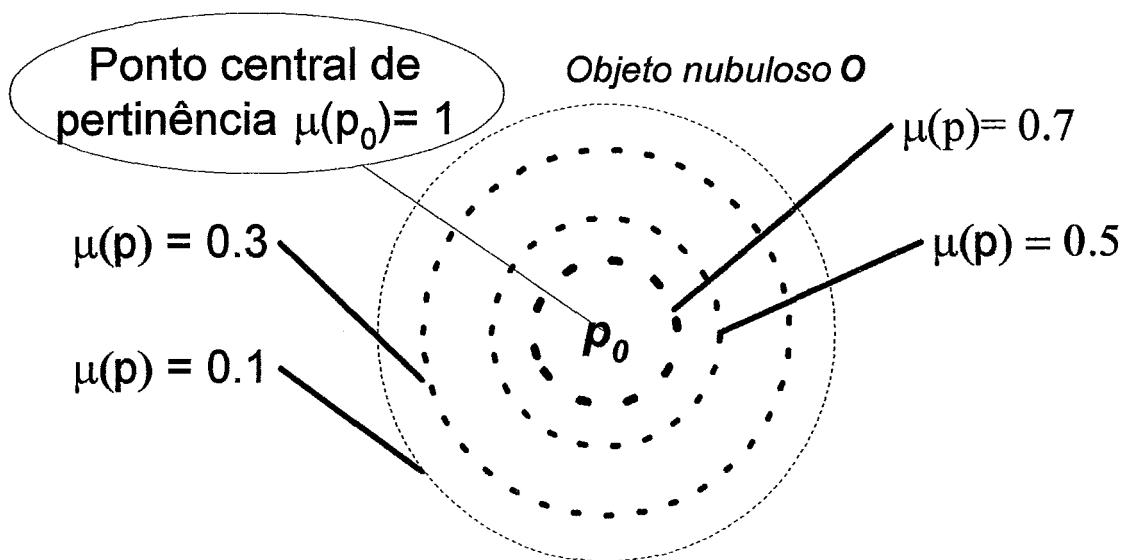


Figura 5-2 - Ponto nebuloso

A distribuição dos valores de pertinência ao longo do espaço pode variar dependendo da forma e extensão do objeto. Objetos complexos como linhas ou áreas podem não ter uma função de distribuição única para todas as partes do objeto. No caso do formato Raster, por exemplo, a função deverá mapear individualmente cada célula no seu valor de pertinência. Desta forma, no modelo proposto aqui, um objeto nebuloso O é composto da agregação de outros objetos a_k tais que,

$$O = \bigcup a_k, k = 1, 2, \dots, n \text{ onde,}$$

$$a_k = \{(p_i, \mu_{a_k}(p_i)) \mid \mu_{a_k}(p_i) > 0, i = 1, 2, \dots, n\}$$

Utilizando o operador de união (Zadeh, 1965) podemos calcular,

$$O = \{(p_i, \mu_O(p_i)) \mid \mu_{a_k} = \max_k (\mu_{a_k}(p_i)), k = 1, 2, \dots, n, i = 1, 2, \dots, n\}$$

O método de agregação de objetos permite que possamos construir objetos mais complexos utilizando outros mais simples, similarmente ao trabalho de Usery (1996), para a construção de uma área nebulosa. Na modelagem apresentada aqui, podemos construir uma área nebulosa da seguinte forma:

Dado um domínio espacial $\mathbf{P} \subseteq \mathfrak{R}^n$ e um conjunto de objetos nebulosos $\mathbf{S} = \mathcal{F}(\mathbf{P})$, a área nebulosa O pode ser dividida em dois objetos nebulosos a_1 e a_2 onde,

- $a_1 = \{(p_i, \mu_{a_1}(p_i)) \mid \mu_{a_1}(p_i) > 0, i = 1, 2, \dots, n\}$ é o objeto nebuloso que representa a linha de contorno da área nebulosa,
- $a_2 = \{(p_i, \mu_{a_2}(p_i)) \mid \mu_{a_2}(p_i) = \sup \mu_{a_1}(p_i), \text{ se } p_i \text{ interior ao contorno, } i = 1, 2, \dots, n\}$ é o objeto nebuloso que representa a área interior ou núcleo do objeto e
- $O = a_1 \cup a_2$ - é construído pela agregação do contorno e o núcleo da área, desta forma a área nebulosa seria definida como:

$$O = \{(p_i, \mu_O(p_i)) \mid \mu_O = \max [\mu_{a_1}(p_i), \mu_{a_2}(p_i)], i = 1, 2, \dots, n\}$$

5.1.4 Modelagem Semântica Nebulosa

A inferência dos dados não gráficos faz-se de forma análoga à seção 5.1.2. No modelo preciso, a função característica somente pode estabelecer as condições de existência ou não do objeto, desta forma, as análises ficam restritas a limites rígidos dos valores dos dados. Num sistema convencional, pode-se selecionar cores diferentes para diversos pontos com várias populações porem, esta modelagem torna-se ineficiente para mostrar ao longo do espaço a variação de concentração de um atributo, como por exemplo, a densidade de vegetação ao longo de uma certa área.

No modelo nebuloso de objetos, podemos estabelecer índices de relevância para pontos do espaço em função de atributos do banco de dados não gráfico. Cada objeto espacial nebuloso Q_j pode ser composto da união de objetos básicos q_{jk} , desta forma podemos ter elementos do banco de dados associados a cada um destes.

Dado um banco de dados qualquer DB e um elemento x_1 do banco associado a um objeto $q_{jk} \in \mathbf{S}$, onde $\mathbf{S} = \mathcal{F}(\mathbf{P})$ é um conjunto de objetos espaciais nebulosos. Podemos redefinir a função h tal que:

$$h: \mathbf{S} \times \text{DB}$$

$$x_1 = h(q_{jk}): q_{jk} \rightarrow x_1$$

Exemplo:

Consideremos um conjunto espacial nebuloso \mathbf{S} de um domínio \mathbf{P} representando os pontos de um mapa e uma coleção de objetos Q_j da classe “fazendas”, cuja população é dada na relação R:

<u>j</u>	<u>Tipo</u>	<u>População</u>
1	Fazenda	10.000
2	Fazenda	5.000
3	Sítio	5.000
4	Cidade	30.000

A população de cada ponto está distribuída em 2.000 pessoas/km ao longo do espaço a partir de um ponto P_0 . Desejamos representar no espaço os objetos da classe fazenda, levando em conta a relevância da sua população.

Cada objeto Q_j será composto de apenas um elemento q_k , e associado a uma tupla x_1 logo por simplicidade, Q_j será:

$$Q_j = \{(p, \mu_{q_j}(p)) \mid \mu_{q_j}(p) > 0\}, \quad p = \{p_i \in \mathbf{P} \mid i=1,2,\dots,n\}$$

A função $\mu_{s_k}(p)$ será mapeada de forma a distribuir a população a partir do centro p_0 onde consideramos a população indicada com certeza de 100%:

- $\mu_{q_j}(p_i) = 1 - 2.000 \times d(P_j, p) / h(q_j).Pop$, se $\{d(P_j, p) < (h(q_j).Pop/2.000)\}$ e $h(q_j).Tipo = \text{“Fazenda”}$ e $\mu_{q_j}(p_i) = 0$, caso contrário.
- $d(P_j, p)$ - é a distância de um ponto qualquer p a p_0

Na Figura 5-3 podemos ver, por exemplo, a representação das duas fazendas com a distribuição populacional em níveis de cinza. Os valores de pertinência indicam a existência do objeto na região bem como a concentração do atributo do banco de dados relacionado ao objeto.

Através da modelagem nebulosa em SIGs, os dados podem apresentar-se de duas formas: convencional ou nebulosa. A modelagem de dados nebulosa em SIGs(seção 3.6.2), mencionada no capítulo anterior, nos traz uma dimensão a mais nos dados associados aos objetos. Podemos ter um banco de dados relacional nebuloso, onde os valores dos dados são também nebulosos. O quadro do exemplo anterior poderia ser escrito como:

k	Tipo	População	Pertinência
1	Fazenda	Média	80%
2	Fazenda	Baixa	75%
3	Sítio	Baixa	75%
4	Cidade	Alta	95%

A função que define a pertinência dos pontos no espaço do objeto depende dos valores do atributo de inferência. Vários métodos podem ser estabelecidos para compor o objeto nebuloso com os dados. Um dos métodos triviais é defuzzificar os valores nebulosos, outras formas possíveis podem ser baseadas em métodos de agregação.

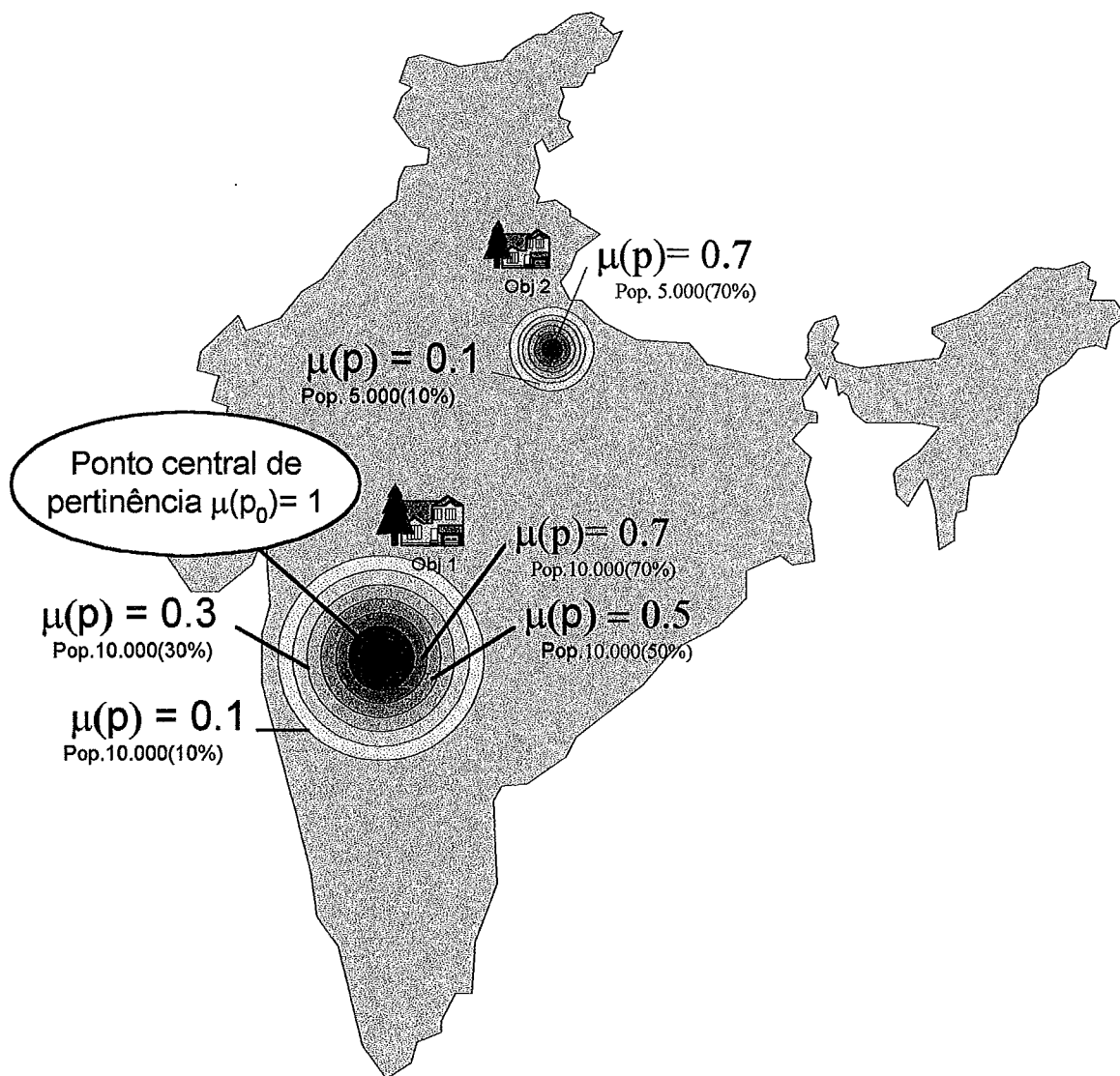


Figura 5-3 - Representação de objetos nebulosos

5.2 Visão nebulosa de objetos espaciais precisos

A descrição nebulosa dos objetos ao longo do espaço parte da obtenção de coordenadas espaciais consideradas precisas. Inferimos a elas a imprecisão a partir das informações e dados conhecidos. Este método, na verdade, consiste em aplicarmos um

método de fuzzyficação espacial aos dados espaciais, supostamente precisos. Devemos mostrar como estabelecemos a visão nebulosa de uma entidade espacial mínima, agregando esta de forma a obter a fuzzyficação dos objetos mais complexos.

O modelo apresentado está baseado na composição de elementos espaciais, com diferentes relevâncias nos vários pontos do espaço. Cada ponto p é descrito por um par $(p, \mu(p))$. O ponto ou célula $p = \{ (x_1, x_2, \dots, x_n) \mid p \in \mathfrak{R}^n \}$ pode ser visto como uma extensão de um número nebuloso (seção 2.2.7). Chamaremos essa extensão de coordenada nebulosa. Ela é formada de n coordenadas que são na verdade números nebulosos.

Um ponto p no espaço é representado por um vetor de coordenadas cujo domínio é $(X_1 \times X_2 \times \dots \times X_n)$. Uma coordenada nebulosa p será analogamente representada pelo produto cartesiano de números nebulosos $X_n \in X_n$, onde X_n é uma variável nebulosa:

$$P(X_1, X_2, \dots, X_n) = \{ (X_1 \in X_1) \times (X_2 \in X_2) \times \dots \times (X_n \in X_n) \}$$

O cálculo de pertinência final pode ser feito utilizando-se o princípio da extensão (seção 2.2.6). Um ponto no espaço pode ser visto como uma função de suas coordenadas:

$$p = f(x_1, x_2, \dots, x_n) = x_1 \mathbf{x}_1 + x_2 \mathbf{x}_2 + \dots + x_n \mathbf{x}_n, \text{ onde,}$$

- \mathbf{x}_n representa o vetor unitário na direção n ,
- $X_n = (x_n, \mu_{X_n}(x))$ é um número nebuloso que fuzzifica a variável x_n .

Pelo princípio da extensão, uma coordenada nebulosa $P(X_1, X_2, \dots, X_n)$ será:

$$P(X_1, X_2, \dots, X_n) = \{ (p, \mu(p)) \mid \mu(p) = \sup_{p=f(x_1, x_2, \dots, x_n)} \min (\mu_{X_1}(x_1), \mu_{X_2}(x_2), \dots, \mu_{X_n}(x_n)) \}$$

As coordenadas nebulosas produzem então os objetos nebulosos, que podem ser analogamente vistos como “números nebulosos complexos”. A sua manipulação é dependente das funções de distribuição ao longo do espaço. Tomando como exemplo uma função triangular simples baseada na distribuição por distância, vamos mostrar o

processo de fuzzyficação de algumas entidades espaciais básicas: ponto, linha poligonal e área.

5.2.1 Pontos Espaciais Nebulosos

Um ponto espacial nebuloso foi implicitamente definido na seção anterior, considerando que a representação de um ponto é feita por uma única coordenada nebulosa. Outra forma de *fuzzificar* um ponto no espaço, baseado na coordenada provável do mesmo, é distribuir a relevância ao longo do espaço uniformemente. A distribuição mais comum (Dutta, 1989) é feita atribuindo-se 100% de relevância ao ponto considerado provável e calcular uma taxa de decrescimento dessa em função da distância. Propomos aqui uma modelagem similar, incluindo porém fatores semânticos e distribuições não lineares.

Um objeto nebuloso \mathcal{O}^α que modela um ponto P_0 do domínio $\mathbf{P} \subseteq \mathfrak{R}^2$. A fuzzificação de P_0 será então:

$$\mathcal{O}^\alpha = \{(p, \mu_\alpha(p)) \mid \mathcal{O}^\alpha \in \mathbf{S}, p \in \mathbf{P}\}, \text{ onde}$$

- $p \in \mathbf{P}$ é um ponto (x, y) qualquer.
- $\mathbf{P} \subseteq \mathfrak{R}^2$ é um domínio espacial.
- $\mathbf{S} = \mathcal{F}(\mathbf{P})$ é o domínio nebuloso definido em \mathbf{P} .
- $\mu_\alpha(p) = 1 - d^n(P_0, p) / \mathbf{K}$, se $d^n(P_0, p) < \mathbf{K}$ e $\mu_\alpha(p) = 0$ caso contrário.
- $d^n(P_0, p)$ é a distância de um ponto qualquer até P_0 elevado a n . O fator n determina a forma de distribuição da relevância. Quando n é igual a 1 a distribuição é linear ao longo do espaço.
- O fator \mathbf{K} determina a taxa de distribuição. Este fator poderá ser função de propriedades advindas de um banco de dados associado.

5.2.2 Segmentos de Reta Nebulosos

A modelagem dos segmentos de reta é feita através da agregação de um objeto de reta nebulosa e dois pontos nebulosos que limitam os extremos do segmento (Figura

5-4). A modelagem dos pontos, digamos σ_1 e σ_2 , já foi descrita acima. A modelagem da reta é igualmente feita utilizando-se o princípio da extensão.

A equação de uma reta r_0 no \mathfrak{R}^2 é escrita como $ax + by + c = 0$. Desta forma uma reta r centrada em r_0 pode ser vista como $r = f(x,y)$, onde uma reta nebulosa seria escrita como:

$$r(\mathbf{X}, \mathbf{Y}) = \{(p, \mu(p)) \mid \mu(p) = \sup_{r=f(x,y)} \min[\mu_X(x), \mu_Y(y)]\} \text{ onde,}$$

- $p \in \mathbf{P}$ é um ponto (x, y) qualquer pertencente à reta r_0 .
- $\mathbf{P} \subseteq \mathfrak{R}^2$ é um domínio espacial.

A construção final do segmento de reta nebuloso R é composto pela agregação dos dois pontos nebulosos σ_1 e σ_2 e a reta r :

$$R = \sigma_1 \cup \sigma_2 \cup r = \{(p, \mu_R(p)) \mid \mu_R = \max[\mu_{\sigma_1}(p), \mu_{\sigma_2}(p), \mu_r(p)]\}$$

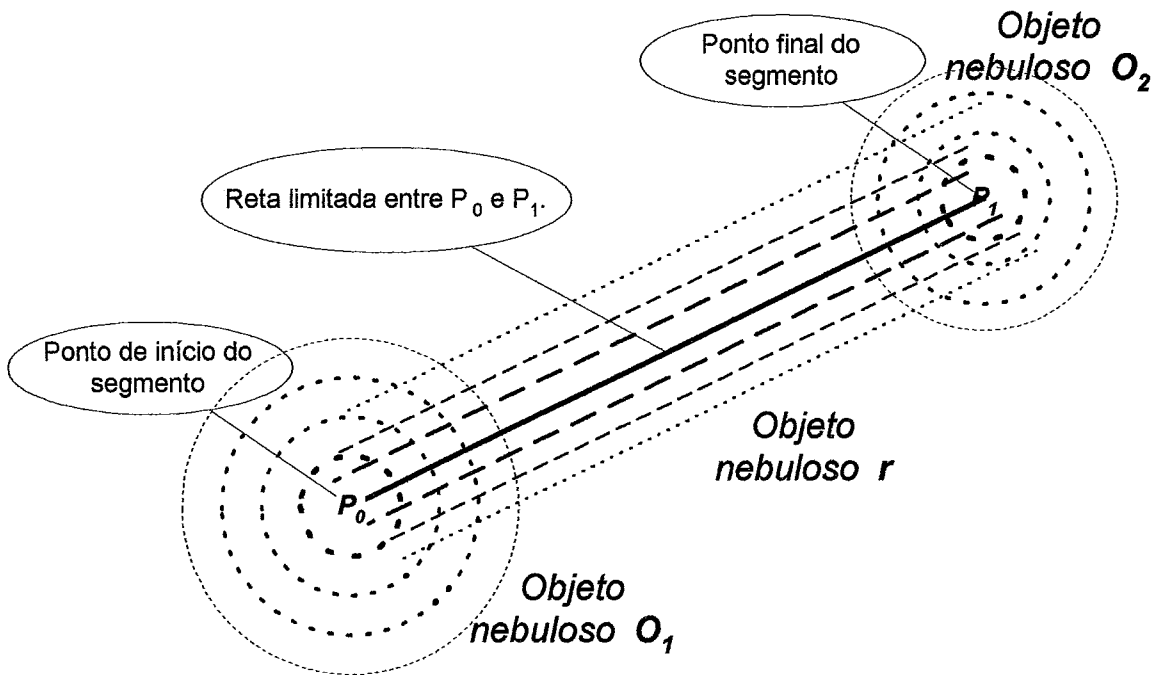


Figura 5-4 - Segmento de reta nebuloso

5.2.3 Áreas de Contorno Nebulosas.

Uma área nebulosa, como mostrado na seção 5.1.3, é construída através de uma linha poligonal fechada limitando uma área no seu interior. Iremos considerar como já

calculada a pertinência da linha poligonal. Dado um domínio espacial $\mathbf{P} \subseteq \mathfrak{R}^2$ e um conjunto de objetos nebulosos $\mathbf{S} = \mathcal{F}(\mathbf{P})$, a área nebulosa \mathcal{O} pode ser dividida em dois objetos nebulosos \mathcal{o}_1 e \mathcal{o}_2 onde,

- $\mathcal{o}_1 = \{(p, \mu_{\mathcal{o}_1}(p)) \mid \mu_{\mathcal{o}_1}(p) > 0\}$ é o objeto nebuloso que representa a linha poligonal de contorno da área nebulosa,
- $\mathcal{o}_2 = \{(p, \mu_{\mathcal{o}_2}(p)) \mid \mu_{\mathcal{o}_2}(p) = \sup \mu_{\mathcal{o}_1}(p), \text{ se } p \text{ interior ao contorno} \}$ é o objeto nebuloso que representa a área interior ou núcleo do objeto e
- $\mathcal{O} = \mathcal{o}_1 \cup \mathcal{o}_2$ - é construído pela agregação do contorno e o núcleo da área, desta forma, a área nebulosa seria definida como:

$$\mathcal{O} = \{ (p, \mu_{\mathcal{O}}(p)) \mid \mu_{\mathcal{O}} = \text{máx} [\mu_{\mathcal{o}_1}(p), \mu_{\mathcal{o}_2}(p)] \}$$

A modelagem semântica de cada objeto nebuloso espacial descrito irá ser feita nas funções de pertinência $\mu_X(x)$ e $\mu_Y(y)$, que compõe cada objeto.

Capítulo VI

6. Linguagem de Definição e Consulta

A linguagem criada aqui tem por finalidade permitir a criação e ativação do modelo nebuloso. Seu formato está baseado numa extensão do SQL, como é a maioria das linguagens propostas na literatura. Sua implementação é feita utilizando um *parser* do tipo LALR(Aho, Sethi e Ullman, 1988) construído pelo BISON versão 1.24. Esse gerador, desenvolvido originariamente por Robert Corbett e Richard Stallman, é compatível com o *Yacc*(Johnson, 1975). O *parser* traduz a linguagem para um bloco de *comandos* básicos do sistema. O sistema pode receber, como entrada, uma lista de *comandos* básicos, desta forma permitimos múltiplas implementações na linguagem proposta.

A linguagem é constituída de *sentenças* divididas em duas partes: uma para definição do modelo e outra para definição e execução de consultas.

O sistema de “compilação” da linguagem executa apenas uma passagem no arquivo, assim a definição do modelo deve seguir uma ordem precisa. Cada componente, é em geral, depende de componentes anteriores. Devemos então ter certeza se um componente já foi definido ao referenciá-lo. Para manter esta consistência, o sistema exige uma ordem entre os grupos de *sentenças* de definição do modelo. O sistema gera dois relatórios, para compilação e execução do modelo, como será explicado no item 6.3.

6.1 Definição do modelo

Para definição do modelo devemos primeiro criar e definir as tabelas que irão adquirir os dados da entrada. A seguir, criamos os componentes que irão compor o modelo, que representam variáveis lingüísticas, termos, modificadores, operadores e regras. Uma vez definidos, estabeleceremos as conexões do modelo, aplicando os modificadores nos termos e esses nas variáveis. Depois de feita a composição, podemos definir regras sobre as variáveis e termos definidos.

BEGIN

1. Definição de interfaces de dados.
2. Definição de componentes: Variáveis lingüísticas, Variáveis Solução, Termos Lingüísticas, Modificadores.
3. Definição de uso de Modificadores em termos.
4. Definição de uso de termos em variáveis lingüísticas.
5. Definição de regras.
6. Execução de consultas.

End;

6.1.1 Definição de interfaces de dados

O sistema desenvolvido funciona de forma a definir uma “visão nebulosa” dos dados oriundos de um banco de dados preciso, desse modo, as interfaces de dados constituem a parte mais importante de um modelo criado no sistema. Elas devem ser declaradas no início do modelo, permitindo a definição das variáveis que estarão a elas associadas.

A criação das interfaces de dados faz-se de duas formas: uma para interfaces de entrada e outra para interfaces de saída. Cada interface é representada internamente na forma de uma tabela com o nome especificado dentro do modelo. As sentenças de criação devem indicar esse nome e o nome externo da tabela no banco de dados de origem. Opcionalmente, pode ser indicado o tipo de interface a ser criada. Os tipos são:

- arquivo texto, cujo formato é indicado na seção 4.2.1;
- tabelas oriundas da interface BDE da Borland, onde devemos ainda indicar o nome do banco de dados;
- SIGEPI(Nobre,Braga,1994) e o
- Gerente de Objetos Armazenáveis GOA++.

Se o tipo não for indicado o comando cria tabelas em modo de arquivo texto.

Interface de Entrada:

```
CREATE TABLE <Nome> FROM <Nome externo> [Tipo de interface] ;
```

Interface de Saída:

```
CREATE TABLE <Nome> AS <Nome externo> [Tipo de interface] ;  
[Tipo de interface] : ARQTEXT | BDE <Base de Dados> | SIG | GOA
```

O sistema permite que se atribua nomes iguais para as tabelas de entrada e saída, uma vez que ele distingue as tabelas de saída diferentemente das tabelas de entrada. Um *comando* independente é criado para cada *sentença*:

Sentença		Comando(s)	
CREATE TABLE	<Nome> FROM <Arquivo>;	CreateData	<nome> <Arquivo>
CREATE TABLE	<Nome> FROM <Arquivo> ARQTEXT;	CreateData	<nome> <Arquivo>
CREATE TABLE	<Nome> FROM <Opcional> SIG;	CreateDataSIG	<nome> <Opcional>
CREATE TABLE	<Nome> FROM <Arquivo> DBF;	CreateDataDBF	<nome> <Arquivo>
CREATE TABLE	<Nome> FROM <Base de Dados> <Tabela> BDE;	CreateDataBDE	<nome> <Base de Dados> <Tabela>
CREATE TABLE	<Nome> FROM <Coleção> <Atributo> GOA;	CreateDataGOA	<nome> <Coleção> <Atributo>
Interfaces de saída			
CREATE TABLE	<Nome> AS <Arquivo>;	CreateOutputTbl	<nome> <Arquivo>
CREATE TABLE	<Nome> AS <Arquivo> ARQTEXT;	CreateOutputTbl	<nome> <Arquivo>
CREATE TABLE	<Nome> AS <Opcional> SIG;	CreateOutputTblSIG	<nome> <Opcional>
CREATE TABLE	<Nome> AS <Arquivo> DBF;	CreateOutputTblDBF	<nome> <Arquivo>
CREATE TABLE	<Nome> AS <Base de Dados> <Tabela> BDE;	CreateOutputTblBDE	<nome> <Base de Dados> <Tabela>
CREATE TABLE	<Nome> AS <Coleção> <Atributo> GOA;	CreateOutputTblGOA	<nome> <Coleção> <Atributo>

Tabela 6-1 - Comandos de criação de interfaces de dados

Exemplo:

O exemplo a seguir (Tabela 6-2) é utilizado nas consultas descritas no capítulo 7. É criada uma tabela especial para o sistema de informação geográfica SIGEPI (Braga, Nobre, et al., 1994). Uma interface de saída para o mesmo sistema é definida. O nome externo é ignorado nesse caso, podendo ser qualquer um. As outras interfaces de saída são arquivos em formato texto. Quando variáveis criadas dentro do modelo não pertencem originalmente ao banco de dados, como variáveis - solução, pode-se criar arquivos extras para expor essas variáveis; podendo ser um recurso interessante.

Sentença	Comando(s)
BEGIN	
CREATE TABLE Tocantins FROM tabl SIG;	CreateDataSIG Tocantins tabl.dat
CREATE TABLE Mapa AS mapa.dat SIG;	CreateOutputTblSIG Mapa mapa.dat
CREATE TABLE Visitas AS Visitas.dat ARQTEXT;	CreateOutputTbl Visitas Visitas.dat
CREATE TABLE Epidemica AS epidem.dat ARQTEXT;	CreateOutputTbl Epidemica Epidem.dat
CREATE TABLE Relatorio11 AS Relat11.dat ARQTEXT;	CreateOutputTbl Relatorio11 Relat11.dat
...	
END;	

Tabela 6-2 - Exemplo de criação de tabelas

6.1.2 Definição de componentes

Os componentes do modelo podem ser declarados em qualquer ordem, visto que a conexão entre eles se dar-se-ão na seção posterior.

Termos Lingüísticos

Os termos lingüísticos podem ser criados com três curvas básicas: Triangulares, Trapezoidais e Gaussianas. Os pontos que definem a curva de cada termo é dada de forma normalizada, ou seja, no intervalo real [0,1]:

Sintaxe:

```
CREATE LINGTERM <Nome> AS TRIANGLE (Inicio, Meio, Fim) ;
| TRAPEZ (Inicio, Meio1, Meio2, Fim) ;
| GAUSS (Inicio, Meio, Fim)
< ALFA corte alfa > ;
```

Sentença	Comando(s)
CREATE LINGTERM <Nome> AS TRIANGLE <Inicio, Meio, Fim>;	CreateTerm <nome> <Inicio><Meio><Fim> TRIANGULAR 0
CREATE LINGTERM <Nome> AS TRIANGLE <Inicio, Meio, Fim> ALFA <Corte alfa>;	CreateTerm <nome> <Inicio><Meio><Fim> TRIANGULAR <Corte alfa>;
CREATE LINGTERM <Nome> AS TRAPEZ <Inicio, Meio1, Meio2, Fim>;	CreateTerm2 <nome> <Inicio><Meio1><Meio2><Fim> TRAPEZ 0
CREATE LINGTERM <Nome> AS TRAPEZ <Inicio, Meio1, Meio2, Fim> ALFA <Corte alfa>;	CreateTerm2 <nome> <Inicio><Meio1><Meio2><Fim> TRAPEZ <Corte alfa>;
CREATE LINGTERM <Nome> AS GAUSS <Inicio, Meio, Fim>;	CreateTerm <nome> <Inicio><Meio><Fim> GAUSS 0
CREATE LINGTERM <Nome> AS GAUSS <Inicio, Meio, Fim> ALFA <Corte alfa>;	CreateTerm <nome> <Inicio><Meio><Fim> GAUSS <Corte alfa>;

Tabela 6-3 - Comandos de criação de termos lingüísticos

Modificadores

Os modificadores, acrescentados posteriormente aos termos lingüísticos, são também declarados independentemente. Para cada tipo de modificador <Fator> é um numérico real que define o fator de filtragem do mesmo.

Sintaxe:

```
CREATE HEDGE <Nome> AS < CONCENTRATOR |
                        INTENSIFICACION |
                        DILATATION> <Fator>;
```

Sentença	Comando(s)
CREATE HEDGE <Nome> AS CONCENTRATOR <Fator>;	CreateHedge <nome> CONCENTRADOR <Fator>
CREATE HEDGE <Nome> AS INTENSIFICACION <Fator>;	CreateHedge <nome> INTENSIFICADOR <Fator>
CREATE HEDGE <Nome> AS DILATATION <Fator>;	CreateHedge <nome> DILATADOR <Fator>

Tabela 6-4 - Comandos de criação de modificadores

Variáveis

As variáveis lingüísticas podem ser inteiras ou reais, podendo-se acrescentar o domínio e a interface usada com o banco de dados. Não existe ordem entre as sentenças adicionais de interface de dados ou domínio, ou seja, pode-se acrescentar o domínio e depois indicar a interface ou vice-versa. O domínio pode não ser indicado, nesse caso a variável tentará buscar essa informação da interface de dados.

Sintaxe:

```
CREATE LINGVAR <INTEGER | REAL> <Nome Variável> [Domínio | Dados]
[Dados] OVER <Nome da tabela> AS <Nome do dado>;
[Domínio] DOMAIN <Valor mínimo>, <Valor máximo>;
```

Sentença	Comando(s)
CREATE LINGVAR INTEGER <Nome> OVER <Tabela> AS <Nomedado>;	CreateVarInt <Nome> <Tabela> <Nomedado>
CREATE LINGVAR INTEGER <Nome> DOMAIN <Mínimo>, <Máximo> OVER <Tabela> AS <Nomedado>;	CreateVarInt <Nome> <Mínimo> <Máximo> <Tabela> <Nomedado>
CREATE LINGVAR REAL <Nome> OVER <Tabela> AS <Nomedado>;	CreateVarReal <Nome> <Tabela> <Nomedado>
CREATE LINGVAR REAL <Nome> DOMAIN <Mínimo>, <Máximo> OVER <Tabela> AS <Nomedado>;	CreateVarReal <Nome> <Mínimo> <Máximo> <Tabela> <Nomedado>

Tabela 6-5 - Comandos de criação de variáveis lingüísticas

As variáveis solução não possuem ligação com interface de dados, uma vez que seus valores são calculados a partir de regras, de toda forma, é obrigatório indicar o domínio das mesmas.

Sintaxe:

CREATE SOLVAR <INTEGER | REAL> <Nome> DOMAIN <Mínimo>, <Máximo>;

Sentença	Comando(s)
CREATE SOLVAR INTEGER <Nome> OVER <Tabela> AS <Nomedado>;	CreateSolInt <Nome> <Tabela> <Nomedado>
CREATE SOLVAR INTEGER <Nome> DOMAIN <Mínimo>, <Máximo>;	CreateSolInt <Nome> <Mínimo> <Máximo>
CREATE SOLVAR INTEGER <Nome> DOMAIN <Mínimo>, <Máximo> OVER <Tabela> AS <Nomedado>;	CreateSolInt <Nome> <Mínimo> <Máximo> <Tabela> <Nomedado>
CREATE SOLVAR REAL <Nome> OVER <Tabela> AS <Nomedado>;	CreateSolReal<Nome> <Tabela> <Nomedado>
CREATE SOLVAR REAL <Nome> DOMAIN <Mínimo>, <Máximo>;	CreateSolReal<Nome> <Mínimo> <Máximo>
CREATE SOLVAR REAL <Nome> DOMAIN <Mínimo>, <Máximo> OVER <Tabela> AS <Nomedado>;	CreateSolReal<Nome> <Mínimo> <Máximo> <Tabela> <Nomedado>

Tabela 6-6 - Comandos de criação de variáveis solução

Exemplo:

O exemplo a seguir(Tabela 6-8) define um grupo de variáveis lingüísticas e os termos Alto, Médio e Baixa. Esses termos são os mais comumente utilizados na modelagem inicial de um sistema nebuloso. Utilizamos a curva triangular de forma que cada termo abrange sempre todo o intervalo normalizado. Isto permite que qualquer valor do domínio sempre possua pertinência para todos os termos definidos. Um modificador foi criado para expandir as possibilidades dos termos em caso de uma distribuição não uniforme dos dados.

6.1.3 Definição conexão de componentes

Os tipos de conexão feitas são quatro, a primeira é feita entre as interfaces de dados e as variáveis lingüísticas. Este processo é feito implicitamente na definição das mesmas. O segundo tipo é feito ligando-se as regras às variáveis solução, como será visto mais adiante. Os dois últimos tipos são entre os termos lingüísticos e as variáveis e, entre os modificadores e os termos lingüísticos. A sintaxe deles é descrita abaixo:

Sintaxe:

USE HEDGE <Nome do modificador> **IN** <Nome do termo>;

USE LINGTERM <Nome do termo> **IN** <Nome da variável>;

Sentença	Comando(s)
USE HEDGE <Modificador> IN <Termo>;	CreateTermHedge <Modificador> <Termo> UseHedge <Modificador> <Termo>
USE LINGTERM <Termo> IN <Variável>;	UseTerm <Termo> <Variável>

Tabela 6-7 - Comandos de conexão de componentes

Sentença	Comando(s)
BEGIN	
...	...
CREATE LINGVAR INTEGER <i>LaminasExaminadas</i> OVER <i>Tocantins</i> AS LAMEXA;	CreateVarInt <i>LaminasExaminadas Tocantins</i> LAMEXA
CREATE LINGVAR INTEGER <i>LamPosFalciparum</i> OVER <i>Tocantins</i> AS LAMPOSF;	CreateVarInt <i>LamPosFalciparum Tocantins</i> LAMPOSF
...	...
CREATE SOLVAR REAL <i>Visitas</i> DOMAIN <i>0,1</i> OVER <i>Tocantins</i> AS VISITAS;	CreateSolReal <i>Visitas 0 1 Tocantins</i> VISITAS
CREATE LINGTERM <i>Baixo</i> AS TRIANGLE (<i>0.0, 0.0, 1.0</i>);	CreateTerm <i>Baixo 0 0 1 TRIANGULAR 0</i>
CREATE LINGTERM <i>Médio</i> AS TRIANGLE (<i>0.0, 0.5, 1.0</i>);	CreateTerm <i>Medio 0 0.5 1 TRIANGULAR 0</i>
CREATE LINGTERM <i>Alto</i> AS TRIANGLE (<i>0.0, 1.0, 1.0</i>);	CreateTerm <i>Alto 0 1 1 TRIANGULAR 0</i>
CREATE LINGTERM <i>Improvável</i> AS TRIANGLE (<i>0.0, 0.0, 1.0</i>);	CreateTerm <i>Improvavel 0 0 1 TRIANGULAR 0</i>
CREATE LINGTERM <i>Provável</i> AS TRIANGLE (<i>0.0, 1.0, 1.0</i>);	CreateTerm <i>Provavel 0 1 1 TRIANGULAR 0</i>
CREATE HEDGE <i>Muito</i> AS CONCENTRATOR <i>3.0</i> ;	CreateHedge <i>Muito</i> CONCENTRADOR 5
...	...
END;	

Tabela 6-8 - Exemplo de criação de componentes nebulosos

Exemplo:

O exemplo abaixo mostra primeiro a conexão de um modificador a um termo *Baixo*. Esta ação cria um termo alternativo “*Muito-Baixo*”, não eliminando contudo o termo *Baixo* originalmente criado. Este termo passa a existir independentemente e é conectado a uma variável.

Sentença	Comando(s)
BEGIN	...
...	...
USE HEDGE <i>Muito</i> IN <i>Baixo</i> ;	UseHedge <i>Muito Baixo</i>
...	...
USE LINGTERM <i>Baixo</i> IN <i>LaminasExaminadas</i> ;	UseTerm <i>Baixo LaminasExaminadas</i>
USE LINGTERM <i>Medio</i> IN <i>LaminasExaminadas</i> ;	UseTerm <i>Medio LaminasExaminadas</i>
USE LINGTERM <i>Alto</i> IN <i>LaminasExaminadas</i> ;	UseTerm <i>Alto LaminasExaminadas</i>
USE LINGTERM <i>Muito-Baixo</i> IN <i>PerFalciparum</i> ;	UseTerm <i>Muito-Baixo PerFalciparum</i>
...	...
END ;	

Tabela 6-9 - Exemplo de conexão de componentes

6.1.4 Definição de regras.

As regras são definidas de forma livre e são compostas de duas estruturas básicas, Proposições e Condições. Cada proposição pode ser uma condição única ou um grupo de outras proposições associadas por um operador lógico nebuloso entre parênteses. Os operadores definidos têm os mesmos nomes dos operadores lógicos clássicos e implementam os operadores lógicos nebulosos padrão de Zadeh(1965). As condições fazem relacionamento entre variáveis lingüísticas e termos. A única condição, que relaciona duas variáveis, tem o propósito de realizar a operação de junção.

Sintaxe:

IF [Proposição] **THEN** [Condição];

[Condição]: <Variável> **IS** <Termo >
| <Variável> **IS (GREATER THAN)** < Termo | Variável >
| <Variável> **IS (LESS THAN)** < Termo | Variável >
| <Variável> **IS (EQUAL)** < Variável >
| <Variável> **=** <Valor | Variável>
| <Variável> **<** <Valor | Variável>
| <Variável> **>** <Valor | Variável>
| <Variável> **<=** <Valor | Variável>
| <Variável> **>=** <Valor | Variável>

[Proposição]: ([Proposição] **AND** [Proposição])
| ([Proposição] **OR** [Proposição])
| **NOT** [Proposição]
| ([Proposição])
| [Condição]

Proposições

As proposições possuem um comando único para as proposições binárias, ou seja as que combinam duas outras proposições e as unárias, no caso o operador de negação e todas as suas formas. Cada comando recebe um parâmetro para indicar o tipo de operador que será usado para criar a proposição.

Sentença	Comando(s)
[Condição]	CreateCondition <Condição> ...
([Proposição] AND [Proposição])	CreatePremiseBin <Proposição> ANDMIN <Condição1> <Condição2>
([Proposição] OR [Proposição])	CreatePremiseBin <Proposição> ORMAX <Proposição1> <Proposição2>
NOT [Proposição]	CreatePremiseUna <Proposição> NOT <Proposição1>

Tabela 6-10 - Comandos de criação de proposições

Condições

As condições(Tabela 6-11) possuem operadores de comparação básicos entre variáveis. Os operadores nebulosos possuem operadores para as comparações lingüísticas. Cada tipo de comparação possui um comando específico para criação da condição. A condição “**IS (EQUAL)**” é utilizada para criar a operação de junção entre variáveis, desta forma, somente será utilizada entre variáveis. A condição <Variável> **IS** <Termo > é usada para comparação de semelhança entre termos e variáveis. As formas de comparação são definidas no capítulo 2. As outras duas formas de comparação: “**IS (GREATER THEN)**” e “**IS (LESS)**” podem realizar-se entre variáveis ou entre termo e variável. O sistema procura, primeiramente, a existência de um termo com o nome dado, se não encontrar ele procura na lista de variáveis. Aconselha-se não utilizar nomes iguais para termos e variáveis.

Os operadores de comparação serão efetuados entre valores numéricos ou variáveis dependendo da identificação feita pelo *parser*.

Regras

As regras(Tabela 6-12) são construídas de forma recursiva a partir da condição e da proposição definidas. Ao decompor uma sentença, definindo uma regra, o sistema vai criando cada condição e proposição e armazenando no banco do esquema. A

variável a qual a regra vai ser associada e o termo que irá se associar à variável é colhido na condição após a sentença **THEN**.

Sentença	Comando(s)
<Variável> IS <Termo >	CreateCondition <Condição> <Variável> <Termo>
<Variável> IS (GREATER THEN) <Termo Variável>	CreateCondGT <Condição> <Variável> <Termo Var>
<Variável> IS (LESS THEN) <Termo Variável>	CreateCondLT <Condição> <Variável> <Termo Var>
<Variável> IS (EQUAL) <Variável>	CreateCondEQ <Condição> <Variável> <Variável>
<Variável> > <Valor >	CreateCondgt1 <Condição> <Variável> <Valor>
<Variável> > <Variável>	CreateCondgt2 <Condição> <Variável> <Variável>
<Variável> >= <Valor >	CreateCondgte1 <Condição> <Variável> <Valor>
<Variável> >= <Variável>	CreateCondgte2 <Condição> <Variável> <Variável>
<Variável> < <Valor >	CreateCondl t1 <Condição> <Variável> <Valor>
<Variável> < <Variável>	CreateCondl t2 <Condição> <Variável> <Variável>
<Variável> <= <Valor >	CreateCondl te1 <Condição> <Variável> <Valor>
<Variável> <= <Variável>	CreateCondl te2 <Condição> <Variável> <Variável>
<Variável> = <Valor >	CreateCondeq1 <Condição> <Variável> <Valor>
<Variável> = <Variável>	CreateCondeq2 <Condição> <Variável> <Variável>

Tabela 6-11 - Comandos de criação de condições

Sentença	Comando(s)
<Proposição>	CreateCondition <Condição1> CreatePremiseBin <Proposição1> <Tipo> <Condição1> <Condição2> ... CreatePremiseBin <Proposição> <Tipo> <Proposição1> <Proposição2> ...
<Condição>	CreateCondition <Condição>...
IF <Proposição> THEN <Condição>;	CreateRule2 <Nome> <Proposição> <Condição>

Tabela 6-12 - Comandos de criação de regras

Exemplo:

Supondo uma regra do tipo “IF A THEN B” o sistema cria, no exemplo abaixo, a proposição A e a nomeia de **P0**. A condição final B é criada com o nome de **P1**. Cada proposição, seja feita por uma condição ou operador, é armazenada como uma nova proposição e é dado o nome **P<índice>**. Um contador interno é incrementado para cada nova proposição que é criada.

Sentença	Comando(s)
<pre> BEGIN ... IF PerVisitadas IS Baixo THEN Visitas IS Aumentar ; ... END;</pre>	<pre> CreateCondition P0 PerVisitadas Baixo CreateCondition P1 Visitas Aumentar CreateRule2 R0 P0 P1</pre>

Tabela 6-13 - Exemplo de criação de regras

6.2 Execução

A linguagem implementa uma extensão na declaração de consultas do SQL(Tabela 6-14) para a utilização de termos e variáveis nebulosas. A cláusula **WITH** acrescentada determina o grau de relevância mínima requisitado na consulta. A cláusula de seleção **WHERE** pode ser omitida, onde o sistema realiza apenas uma projeção das variáveis.

Sintaxe:

```

SELECT <Lista de variáveis linguísticas>
FROM <Lista de interfaces de entrada>
TO <Nome da interface de saída>
[ WHERE [Proposição] ]
WITH <Valor de relevância>
```

Exemplos:

O exemplo a seguir(Tabela 6-15) mostra a conversão de uma projeção simples. A sentença **SELECT** ativa dentro do sistema cada variável que será projetada. Qualquer seleção ou junção sempre incluirá uma projeção, ou seja definirá as variáveis que serão enviadas para a interface de saída.

O comando de projeção simples recebe como parâmetro a interface de saída para aonde os dados devem ser enviados. Ao se terminar um comando de projeção ou seleção todas as variáveis do sistema são desativadas para possibilitar um próximo comando.

A sentença **FROM** aceita uma lista de interfaces de entrada que são postas numa lista temporária de tabelas pelo comando **SetTmpTbl**. Após o término da seleção ou projeção, todas as tabelas são retiradas da lista.

Sentença	Comando(s)
<pre> BEGIN CREATE TABLE Tocantins FROM tabl SIG; ... CREATE LINGVAR INTEGER LaminasExaminadas OVER Tocantins AS LAMEXA; ... SELECT LamPosFalciparum, LamPosVivax, LamPosMalariae, LPFalciparumVivax, LamPositivasTotal, LaminasExaminadas FROM Tocantins TO Mapa; ... END;</pre>	<pre> CreateDataSIG Tocantins tabl.dat ... CreateVarInt LaminasExaminadas Tocantins LAMEXA ... Validate LamPosFalciparum Validate LamPosVivax Validate LamPosMalariae Validate LPFalciparumVivax Validate LamPositivasTotal Validate LaminasExaminadas SelTmpTbl Tocantins Projec Mapa ... </pre>

Tabela 6-15 - Exemplo de execução de consultas de projeção

O exemplo a seguir (Tabela 6-16) faz uma seleção e envia os resultados para um arquivo texto, como definido no exemplo anterior. As variáveis a serem projetadas são ativadas e a tabela correspondendo a interface de entrada é posta na lista temporária. A seguir o predicado composto de uma proposição é montado da mesma forma que nas regras. A relevância final do predicado também é enviada para a interface de saída. Quando usamos operadores entre proposições A e B, devemos colocar entre parênteses a sentença e as condições envolvidas na operação, da forma: ((A) Operador (B)). Os parênteses podem ser aninhados recursivamente.

Sentença	Comando(s)
<pre> BEGIN ... SELECT PerPositivasTotal, CasasExistentes, Retorno FROM Tocantins TO Epidemica WHERE ((((PerPositivasTotal IS Alto)OR (PerPositivasTotal IS Medio))AND ((CasasExistentes IS Alto)OR (CasasExistentes IS Medio)))OR (Retorno IS Provavel)); ... END;</pre>	<pre> ... Validate PerPositivasTotal Validate Retorno Validate CasasExistentes SelTmpTbl Tocantins CreateCondition P10 PerPositivasTotal Alto CreateCondition P11 PerPositivasTotal Medio CreatePremiseBin P12 ORMAX P10 P11 CreateCondition P13 CasasExistentes Alto CreateCondition P14 CasasExistentes Medio CreatePremiseBin P15 ORMAX P13 P14 CreatePremiseBin P16 ANDMIN P12 P15 CreateCondition P17 Retorno Provavel CreatePremiseBin P18 ORMAX P16 P17 Select P18 Epidemica ... </pre>

Tabela 6-16 - Exemplo de execução de consultas com projeção e seleção

O exemplo seguinte(Tabela 6-17) permite uma sentença extra para designar um mínimo de relevância desejado para o predicado de seleção. No exemplo, registros com relevância menor que 30% são desconsiderados.

Sentença	Comando(s)
<pre>BEGIN ... SELECT CasasExistentes, PerVisitadas, Visitas FROM Tocantins TO Mapa WHERE ((Visitas IS Aumentar) AND ((CasasExistentes IS Medio) OR (CasasExistentes IS Alto))) WITH 0.3; END;</pre>	<pre>... Validate CasasExistentes SelTmpTbl Tocantins CreateCondition P13 Visitas Aumentar CreateCondition P14 CasasExistentes Medio CreateCondition P15 CasasExistentes Alto CreatePremiseBin P16 ORMAX P14 P15 CreatePremiseBin P17 ANDMIN P13 P16 Select P17 Mapa MIN 0.3</pre>

Tabela 6-17 - Exemplo de execução de consultas com mínimo de relevância

6.3 Relatórios de erro

O sistema gera dois relatórios correspondentes a compilação e execução do arquivo fonte de definição do modelo, como pode ser visto no exemplo da Tabela 6-18. O arquivo de compilação processado pelo *parser* fornece um relatório de cada elemento ou consulta definida. Caso alguma sentença não tenha sido reconhecida, o arquivo indicará que o elemento que deveria ter sido definido ali não foi reconhecido. Caso a ordem tenha sido violada, todas as sentenças seguintes não serão reconhecidas. O arquivo tem a forma:

<Nome do arquivo de definição(*.fql)>.log

Exemplo:

Capitulo71.log
<pre>Definindo tabela 1 Definindo tabela 2 Definindo Componente 1 ... Definindo Componente 24 Utilizando modificador 1 Utilizando termo 1 ... Utilizando termo 42 Definindo regra 1 ... Definindo regra 4 Definindo consulta1 Arquivo compilado</pre>

Tabela 6-18 - Exemplo de relatório de execução dos comandos da linguagem

O arquivo de execução é gerado pelo próprio sistema. Ele conterá a lista de erros ocorridos durante a definição e consulta ao modelo. Caso nenhum erro ocorra, esse arquivo ficará vazio. Ele tem a forma:

<Nome do arquivo de definição(*.fql)>.err

Exemplo:

As sentenças abaixo pertencem ao processamento de um arquivo com erro na interface de entrada. Isso causa a falha em todos os comandos que dependem dela ou seja, as variáveis e as premissas ligadas a interface de entrada:

Capitulo71.err
Interface de dados Tocantins nao encontrada
...
Interface de dados de origem nao encontrada
Variavel LaminasExaminadas nao encontrada
Variavel LamPosFalciparum nao encontrada
Variavel LamPosVivax nao encontrada
Variavel LamPositivasTotal nao encontrada
Variavel PerFalciparum nao encontrada
Variavel PerFalciparum nao encontrada
...
Premissas nao encontradas
Premissa e/ou condicao da regra nao encontrada(s)
Interface de saida Positivas nao encontrada

Tabela 6-19 - Exemplo de relatório de erros na criação do modelo nebuloso

6.4 Estrutura

A estrutura da linguagem de consulta é dividida em suas partes léxica, sintática e semântica. O interpretador léxico foi construído utilizando-se o FLEXBISON versão. 2.4.7, sendo os *tokens*(Aho,Sethi e Ullman,1988), ou símbolos terminais da linguagem definidos no arquivo de definição sintática e exportados para o intepretador léxico.

Os *tokens*, ou símbolos terminais, podem ser identificadores, que são grupos de caracteres usados para definir nomes de componentes ou arquivos em disco, ou palavras-chave da linguagem implementada.

Identificadores:

IDNAME

IDARQNAME

IDREALVALUE

O interpretador léxico foi gerado sem sensibilidade de caso, desta forma os símbolos terminais(Tabela 6-20) podem ser escritos em letras maiúsculas ou minúsculas.

Símbolos terminais

ENDLINE	REAL	ARQTEXT	REAL
SEPARATOR	WHERE	BDE	OVER
OPAREN	TO	SIG	DOMAIN
CPAREN	WITH	DBF	AS
GREATER	IF	GOAGREATER	SELECT
LESS	OVER	LESS	FROM
EQUAL	DOMAIN	EQUAL	WHERE
OR	AS	OR	TO
AND	SELECT	AND	WITH
NOT	FROM	NOT	IF
ZBEGIN	IS	BEGIN	IS
END	IN	END	IN
CREATE	THEN	CREATE	THEN
USE	TABLE	USE	TABLE
EXECUTE	TRIANGLE	EXECUTE	TRIANGLE
QUERY	TRAPEZ	QUERY	TRAPEZ
LINGVAR	GAUSS	LINGVAR	GAUSS
SOLVAR	ALFA	SOLVAR	ALFA
LINGTERM	CONCENTRATOR	LINGTERM	CONCENTRATOR
HEDGE	INTENSIFICATION	HEDGE	INTENSIFICATION
INTEGER	DILATATION	INTEGER	

Tabela 6-20 - Lista de palavras-chave da linguagem

O arquivo de gramática utilizado pode ser visto no apêndice A, seu padrão permite a utilização de outros pacotes compatíveis com o *YACC*. A linguagem implementada permite um grupo mínimo de operações para os objetivos do protótipo. Propomos uma expansão futura que irá permitir maiores possibilidades de manipulação.

Capítulo VII

7. Aplicação: Dados de análise de Malária

Uma das áreas de grande utilização de Sistemas de Informação Geográfica são as áreas de controle epidemiológico. O controle de doenças transmissíveis ou infecciosas necessita de controle e vigilância de seus dados ao longo do espaço de infestação.

Nestas áreas, utiliza-se, em grande parte o conhecimento de especialistas de área não matemática onde o conhecimento não pode ser facilmente extraído ou modelado. Para tanto, é muitas vezes, necessário ferramentas que facilitem ou permitam ao agente de saúde lidar com os dados coletados.

A doença utilizada como teste neste trabalho atende a este comportamento. Para melhor compreensão das análises feitas aqui, faremos um breve resumo do comportamento e fatores desta doença (Bailey, 1982).

Esta análise de dados foi feita incorporando-se o sistema de manipulação de dados nebulosos (Capítulo 5) no SIGEPI (Braga, Nobre, et al., 1994). Este sistema foi desenvolvido para análise epidemiológica.

7.1 Malária

A malária é uma doença infecciosa que afeta o homem e é transmitida por algumas espécies de mosquito. Ela atingia em 1982, cerca de 150 milhões de pessoas no mundo e tem atingido em grandes regiões da África, América do Sul e sudoeste da Ásia. Sua situação vem piorando principalmente em países com alta taxa de população de baixa renda, onde as condições sanitárias rurais são precárias.

Vários são os fatores que influenciam na sua transmissão e propagação. Entre os fatores mais importantes estão as condições econômicas e sociais das regiões afetadas, a geografia dos locais e o desenvolvimento de tecnologias de controle. Em muitos países, o inseto transmissor está se tornando imune a muitos tipos de inseticidas.

Hoje em dia, existe consenso que as técnicas de modelagem matemática para controle e combate deste tipo de doença devem utilizar métodos que melhor incorporem os conceitos da realidade da doença.

7.1.1 Ciclo de vida

A Malária humana é causada por um protozoário do gênero *Plasmódium*. Ela é transmitida ao homem pela fêmea do mosquito do gênero *Anopheles*. O homem funciona em realidade como um hospedeiro intermediário, apesar de ser o único que apresenta sintomas da infecção, pois o mosquito não é afetado por ela.

O ciclo funciona da seguinte forma: suponhamos um indivíduo inicialmente picado pela fêmea do mosquito. As glândulas salivares do mosquito contêm células do *Plasmódium* num estágio denominado *esporozoito*. Eles caminham para o sangue onde começam a reproduzir-se gerando *Merozoitos*. Estes últimos instalam-se nos glóbulos vermelhos do sangue destruindo-os. Este ciclo pode gerar cerca de um bilhão de parasitas, quando então surgem os sintomas da malária.

Dentro dos glóbulos vermelhos hospedeiros irão se formar *gametócitos*, que são células pré-sexuais do parasita. Neste estágio, se o indivíduo for novamente picado o mosquito levará estes *gametócitos* para terminar seu desenvolvimento. Nele os *gametócitos* irão terminar seu desenvolvimento gerando ao final o *esporozoito* que irá se instalar na nas glândulas salivares do mosquito reiniciando-se o ciclo.

7.1.2 Espécies

Existem 4 espécies distintas de protozoários do gênero *Plasmódium* que causam a Malária humana. Cada espécie diferencia-se principalmente pela intensidade e forma de infestação.

- *Plasmódium falciparum* - É o mais mortal dos tipos. A infestação primária de Merozoites desaparece após instalar-se totalmente nas células vermelhas do sangue. O período de incubação é de 12 dias e os sintomas de febre tendem a surgir de 36 ou de 48 em 48 horas. Tem problemas de irregularidade de sintomas. É mais comum em áreas tropicais ou sub-tropicais, podendo também ocorrer em áreas temperadas.
- *Plasmódium malariae* - A infestação primária igualmente desaparece como no tipo anterior. O período de incubação é de 30 dias e os sintomas ocorrem em ciclos de 3 dias. É encontrado em áreas similares ao anterior, porém com menos freqüência.
- *Plasmódium Vivax* - A infestação primária de *Merozoites* não desaparece e continua sua reprodução mesmo depois da infecção dos glóbulos vermelhos. Este tipo gera um retorno da doença após períodos de repouso. Ciclo de febre de 2 a 3 dias. É mais

benigna e o período de incubação é de 13-17 dias. Tem áreas geográficas menos restritas e é mais encontrado em áreas temperadas.

- *Plasmódium ovale* - Remanesce como o tipo anterior e é usualmente confundida com a mesma na questão dos sintomas. É principalmente encontrada nas áreas tropicais da África e algumas vezes no oeste do Pacífico.

Pode-se notar que o *P.falciparum* e o *P.malariae* são tipos mais graves, contudo os outros dois trazem o problema da doença permanecer no indivíduo para futura ativação. Mais de um tipo pode infectar o indivíduo. Preocupa-se principalmente em combinações como *P.falciparum* e *P.vivax* onde o padrão clínico pode complicar muito.

7.1.3 Ambiente de desenvolvimento

Apesar da grande diversidade e espalhamento pelo mundo, a Malária tende a restringir-se em regiões entre 65° N e 40° S de latitude, entre altitudes menores que 3000 metros. As áreas de montanha e deserto, onde não há muito água, são naturalmente inadequadas para o seu desenvolvimento, pois a água é necessária para o desenvolvimento do mosquito transmissor e hospedeiro.

A temperatura também é um fator crítico. Em áreas onde a temperatura mensal é menor que 16°C, mesmo em presença do mosquito, é muito fria para o desenvolvimento do parasita. Existem condições diferentes para a existência do parasita e do mosquito, afetando inclusive o tamanho do ciclo de vida dos mesmos.

7.1.4 Tratamento e controle

O método de tratamento difere para cada espécie. Um grupo de drogas relativamente acessível economicamente e efetivas são as 4-Aminoquinolinas, contudo a latência do *P.Vivax* e *P.Ovale* não é atingida por elas.

Uma das medidas de controle é eliminar o vetor de transmissão da doença, isto é o mosquito. Várias experiências e tratamentos com o inseticidas tem sido feitos. De toda forma, das 2000 espécies de mosquito, apenas 400 são do gênero *Anopheles* e destes apenas 60 podem carregar malária sob condições naturais.

Como podemos observar, o tratamento e controle da malária envolvem o conhecimento e controle da geografia das áreas infestadas, tipos de infestação, além de

outros fatores. Essa área, bem como outras doenças infecciosas, é bastante fértil para utilização de técnicas especializadas de manipulação e tratamento de dados, como Sistemas de Informação Geográfica e modelagem matemática avançada.

7.2 Análise e representação de Dados Convencionais

7.2.1 Descrição dos dados

Utilizamos dados de Malária coletados no distrito sanitário de Tocantins como base de teste(Nobre, Macedo, et al.1990). Eles são coletados através de informações oriundas de cada paciente diagnosticado. Os estados são divididos em distritos sanitários e cada distrito dividido em municípios(Figura 7-1). Os pacientes são provenientes de diversos pontos do município. Esses pontos são chamados localidades e podem ser cidades, fazendas, vilas, engenhos, etc. Essas localidades podem também abrigar um posto de saúde ou de coleta de dados. Os postos de saúde são responsáveis pelo tratamento de pacientes que a eles se dirigem, bem como o envio de agentes sanitários a localidades onde não hajam postos de saúde. Aqui são centralizados os dados e enviados para os postos sanitários do município que centralizarão os dados de todas as localidades do município.

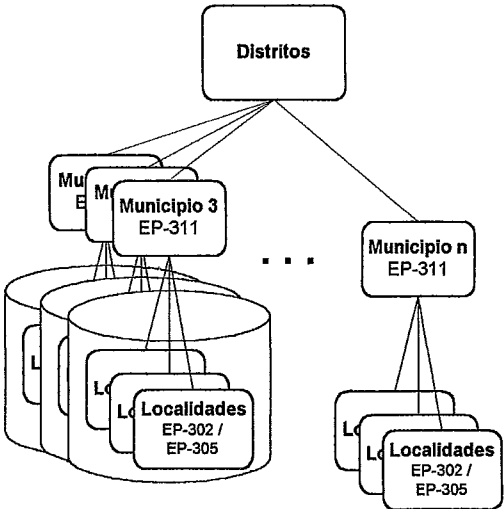


Figura 7-1 - Distribuição espacial dos dados

Existem dois tipos de coleta de informações, a busca ativa e a busca passiva. Como podemos observar na Figura 7-2, na busca ativa, os agentes sanitários vão até os locais determinados e fazem uma pesquisa diretamente junto à moradia do(s)

Estes dados são posteriormente agregados, num relatório denominado EP-311 onde os pesquisadores e gerentes do departamento de saúde irão averiguar as estratégias de vigilância e controle adequadas, como aumentar número de agentes, áreas a visitar, medicamentos utilizados, etc.

7.2.2 Modelagem

Dentre as várias informações descritas acima contidas nas fichas de EP-311, as seguintes serão utilizadas em cada registro de localidade:

- **LAMEXA** - Número de indivíduos examinados. A sigla refere-se a número de lâminas examinadas.
- **LAMPOSF** - Número de indivíduos que acusaram somente o *Plasmodium falciparum* no sangue.
- **LAMPOSV** - Número de indivíduos que acusaram somente o *Plasmodium vivax* no sangue.
- **LAMPOSM** - Número de indivíduos que acusaram somente o *Plasmodium malariae* no sangue.
- **LAMPOSFV** - Número de indivíduos que acusaram o *Plasmodium falciparum* e *Plasmodium vivax* no sangue.
- **LAMPOSTOT** - Número total de indivíduos que acusaram o *Plasmodium* no sangue.
- **CASEXI** - Número de casas existentes na área de pesquisa.
- **CASVIS** - Número de casas visitadas na área de controle.

Criamos algumas variáveis extras para poder trabalhar com índices relativos de cada localidade:

- *PerFalciparum* - Percentual de indivíduos que acusaram somente o *Plasmodium falciparum* no sangue em relação ao número de lâminas examinadas:
 $PerFalciparum = \text{LAMPOSF} / \text{LAMEXA}$
- *PerVivax* - Percentual de indivíduos que acusaram somente o *Plasmodium vivax* no sangue em relação ao número de lâminas examinadas:
 $PerVivax = \text{LAMPOSV} / \text{LAMEXA}$
- *PerMalariae* - Percentual de indivíduos que acusaram somente o *Plasmodium malariae* no sangue em relação ao número de lâminas examinadas:

$PerMalariae = \text{LAMPOSM} / \text{LAMEXA}$

- *PerFalcVivax* - Percentual de indivíduos que acusaram o *Plasmodium falciparum* e *Plasmodium vivax* no sangue em relação ao número de lâminas examinadas:

$PerFalcVivax = \text{LAMPOSFV} / \text{LAMEXA}$

- *PerPositivasTotal* - Percentual de total de indivíduos que acusaram o *Plasmodium* no sangue em relação ao número de lâminas examinadas:

$PerPositivasTotal = \text{LAMPOSTOT} / \text{LAMEXA}$

- *PerVisitadas* - Percentual de casas visitadas na área de controle em relação ao número de casas existentes:

$PerVisitadas = \text{CASVIS} / \text{CASEXI}$

- **VISITAS** - Variável criada pelo sistema para controle do aumento ou diminuição de visitas nas diversas áreas.
- **RETORNO** - Variável criada para indicar a possibilidade de retorno da doença devido a ação do *Plasmodium vivax*.

O modelo será criado utilizando a linguagem de definição de modelos nebulosos descrita no capítulo 6. O primeiro passo será criar uma interface de dados de entrada para o SIGEPI. Criamos dois tipos de interface de saída: uma reenviando ao sistema para representação no mapa e outra para arquivos texto em disco.

A tabela de entrada (*Tocantins*) e a interface de saída (*Mapa*) correspondem às tabelas de pesquisa efetuadas sobre o mapa de uma área analisada. Cada registro das tabelas pesquisadas no mapa corresponde a um município da região. O mapa utilizado nas consultas corresponde ao distrito sanitário de Porto Nacional no estado de Tocantins (Figura 7-3).

Comandos:

```
CREATE TABLE Tocantins FROM tab1 SIG;  
CREATE TABLE Mapa AS mapa.dat SIG;  
CREATE TABLE Visitas AS Visitas.dat ARQTEXT;  
CREATE TABLE Epidemica AS epidem.dat ARQTEXT;  
CREATE TABLE Relatorio11 AS Relat11.dat ARQTEXT;  
CREATE TABLE Relatorio12 AS Relat12.dat ARQTEXT;
```

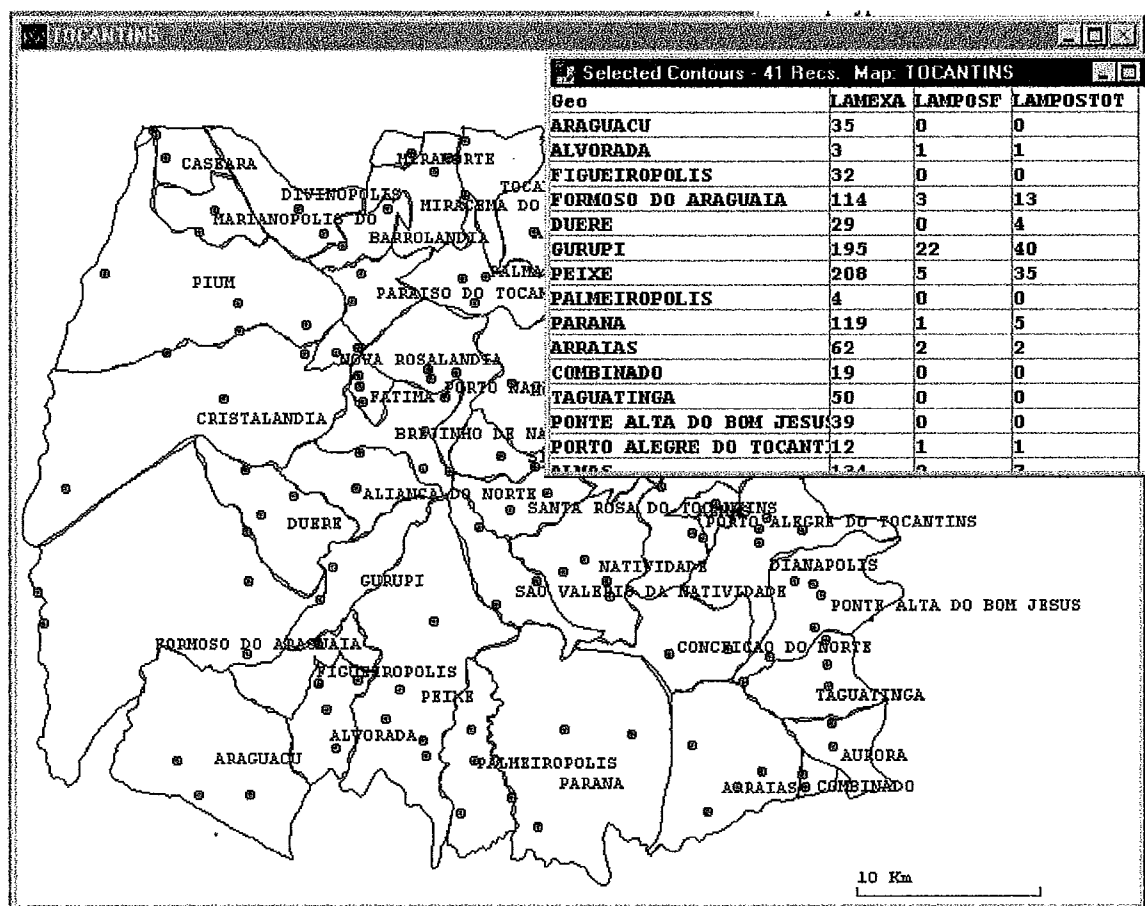


Figura 7-3 - Mapa da área a ser pesquisada

Em seguida definimos as variáveis lingüísticas correspondentes às variáveis do banco de dados. As variáveis originais de coleta foram definidas como inteiras, uma vez que são contagens numéricas. Não especificamos a faixa de valores do domínio, pois ele será obtido diretamente pela interface de dados. As variáveis que indicam percentuais relativos foram declaradas como reais. O seu domínio encontra-se entre os valores máximos de 0 e 1, contudo, podemos ter no banco de dados uma tabela inteira que não contenha valores percentuais acima de um certo valor. Isto nos permite analisar as condições relativas dentro dos dados em questão.

Comandos:

```

CREATE LINGVAR INTEGER LaminasExaminadas
CREATE LINGVAR INTEGER LamPosFalciparum
CREATE LINGVAR INTEGER LamPosVivax
CREATE LINGVAR INTEGER LamPosMalariae
CREATE LINGVAR INTEGER LPFalciparumVivax
CREATE LINGVAR INTEGER LamPositivasTotal
CREATE LINGVAR INTEGER CasasExistentes
CREATE LINGVAR INTEGER CasasVisitadas

OVER Tocantins AS LAMEXA ;
OVER Tocantins AS LAMPOSF ;
OVER Tocantins AS LAMPOSV ;
OVER Tocantins AS LAMPOSM ;
OVER Tocantins AS LAMPOSFV ;
OVER Tocantins AS LAMPOSTOT ;
OVER Tocantins AS CASEXI ;
OVER Tocantins AS CASVIS ;

```

```

CREATE LINGVAR REAL PerFalciparum OVER Tocantins FROM LAMPOSF / LAMEXA;
CREATE LINGVAR REAL PerVivax OVER Tocantins FROM LAMPOSV / LAMEXA;
CREATE LINGVAR REAL PerMalariae OVER Tocantins FROM LAMPOSM / LAMEXA;
CREATE LINGVAR REAL PerFalcVivax OVER Tocantins FROM LAMPOSFV / LAMEXA;
CREATE LINGVAR REAL PerPositivasTotal OVER Tocantins FROM LAMPOSTOT / LAMEXA;
CREATE LINGVAR REAL PerVisitadas OVER Tocantins FROM CASVIS / CASEXI;

```

Mais duas variáveis serão criadas, desta vez como variáveis-solução(seção 4.2.2), ou seja, serão resultantes de regras sobre as outras variáveis. O domínio foi arbitrariamente determinado entre 0 e 1. O sistema irá inferir à variável *Visitas* o número de inspeções que deve ser feita na localidade em questão, indica se deve aumentar ou permanecer com a frequência atual.

A variável *Retorno* avalia a possibilidade de retorno da doença, indicando como possível ou improvável, devido à infecção causada pelo *Plasmodium vivax*.

Comandos:

```

CREATE SOLVAR REAL Visitas DOMAIN 0,1 OVER Tocantins AS VISITAS;
CREATE SOLVAR REAL Retorno DOMAIN 0,1 OVER Tocantins AS VISITAS;

```

Os termos lingüísticos, basicamente utilizados nas variáveis comuns, serão Alto, Médio e Baixo. Usaremos a curva triangular padrão. Os termos Aumentar e Permanecer será usado para o número de casas visitadas e os termos: Improvável e Provável serão usados para as possibilidades de retorno da doença.

Comandos:

```

CREATE LINGTERM Baixo AS TRIANGLE (0.0, 0.0, 1.0);
CREATE LINGTERM Médio AS TRIANGLE (0.0, 0.5, 1.0);
CREATE LINGTERM Alto AS TRIANGLE (0.0, 1.0, 1.0);

CREATE LINGTERM Permanecer AS TRIANGLE (0.0, 0.0, 1.0);
CREATE LINGTERM Aumentar AS TRIANGLE (0.0, 1.0, 1.0);

CREATE LINGTERM Improvável AS TRIANGLE (0.0, 0.0, 1.0);
CREATE LINGTERM Provável AS TRIANGLE (0.0, 1.0, 1.0);

```

Em alguns casos, desejamos separar valores mais próximos de uma pertinência mais alta. Para isto criamos um modificador concentrador e o associamos a cada termo lingüístico.

Comandos:

```
CREATE HEDGE Muito AS CONCENCTRATOR 3.0;  
USE HEDGE Muito IN Baixo ;  
USE HEDGE Muito IN Medio ;  
USE HEDGE Muito IN Alto ;
```

Os termos lingüísticos devem ser devidamente associados às variáveis definidas no modelo. Cada grupo Alto, Médio e Baixo é associado às variáveis simples. Os outros serão associados às variáveis solução através de regras:

Comandos:

```
USE LINGTERM Baixo IN LaminasExaminadas ;  
USE LINGTERM Medio IN LaminasExaminadas ;  
USE LINGTERM Alto IN LaminasExaminadas ;  
  
USE LINGTERM Baixo IN LamPosFalciparum ;  
USE LINGTERM Medio IN LamPosFalciparum ;  
USE LINGTERM Alto IN LamPosFalciparum ;  
  
USE LINGTERM Baixo IN LamPosVivax ;  
USE LINGTERM Medio IN LamPosVivax ;  
USE LINGTERM Alto IN LamPosVivax ;  
  
USE LINGTERM Baixo IN LamPosMalariae ;  
USE LINGTERM Medio IN LamPosMalariae ;  
USE LINGTERM Alto IN LamPosMalariae ;  
  
USE LINGTERM Baixo IN LPFalciparumVivax ;  
USE LINGTERM Medio IN LPFalciparumVivax ;  
USE LINGTERM Alto IN LPFalciparumVivax ;  
  
USE LINGTERM Baixo IN LamPositivasTotal ;  
USE LINGTERM Medio IN LamPositivasTotal ;  
USE LINGTERM Alto IN LamPositivasTotal ;  
  
USE LINGTERM Baixo IN CasasExistentes ;  
USE LINGTERM Medio IN CasasExistentes ;  
USE LINGTERM Alto IN CasasExistentes ;  
  
USE LINGTERM Baixo IN CasasVisitadas ;  
USE LINGTERM Medio IN CasasVisitadas ;  
USE LINGTERM Alto IN CasasVisitadas ;  
  
USE LINGTERM Muito-Baixo IN PerFalciparum ;  
USE LINGTERM Medio IN PerFalciparum ;  
USE LINGTERM Alto IN PerFalciparum ;  
  
USE LINGTERM Baixo IN PerVivax ;  
USE LINGTERM Medio IN PerVivax ;  
USE LINGTERM Alto IN PerVivax ;  
  
USE LINGTERM Baixo IN PerMalariae ;  
USE LINGTERM Medio IN PerMalariae ;  
USE LINGTERM Alto IN PerMalariae ;  
  
USE LINGTERM Baixo IN PerFalcVivax ;  
USE LINGTERM Medio IN PerFalcVivax ;  
USE LINGTERM Alto IN PerFalcVivax ;  
  
USE LINGTERM Muito-Baixo IN PerPositivasTotal ;  
USE LINGTERM Medio IN PerPositivasTotal ;  
USE LINGTERM Alto IN PerPositivasTotal ;  
  
USE LINGTERM Baixo IN PerVisitadas ;  
USE LINGTERM Medio IN PerVisitadas ;  
USE LINGTERM Alto IN PerVisitadas ;
```

O presente trabalho não tem por finalidade gerar resultados com validade epidemiológica. Esses dados serviram para mostrar a utilidade das ferramentas e técnicas desenvolvidas. Com base nisso, propusemos algumas regras para mostrar as possibilidades de administrar-se controle e vigilância de uma doença infecciosa.

“Se o percentual de lâminas positivas por lâminas examinadas for elevado o número de casas visitadas deve aumentar”.

“Se o percentual de lâminas positivas por lâminas examinadas for baixo o número de casas visitadas deve permanecer”.

Podemos verificar que se o número de casos da doença aumenta, devemos aumentar a fiscalização nas áreas de ocorrência.

Comandos:

```
IF PerVisitadas IS Baixo THEN Visitas IS Aumentar ;  
IF PerPositivasTotal IS Baixo THEN Visitas IS Permanecer ;
```

“Se o número de lâminas positivas de *Plasmodium vivax* é alto o retorno da doença é provável” .

“Se o número de lâminas positivas de *Plasmodium vivax* é baixo o retorno da doença é improvável” .

Podemos verificar que a taxa de retorno da doença é proporcional a taxa de infecção pelo *Plasmodium vivax*.

Comandos:

```
IF PerVivax IS Alto THEN Retorno IS Provável ;  
IF PerVivax IS Baixo THEN Retorno IS Improvável ;
```

7.2.3 Descrição das análises

Inicialmente faremos uma observação geral dos dados contidos na base. Para uma análise superficial verificaremos o índice de aparecimento de cada *plasmodium*, através do número de indivíduos examinados em cada caso. O exemplo apresentado aqui utiliza dados coletados nos meses de Janeiro, Abril e Maio de 1992.

Comando:

```
SELECT LamPosFalciparum, LamPosVivax, LamPosMalariae,  
LPFalciparumVivax, LamPositivasTotal, LaminasExaminadas  
FROM Tocantins  
TO Mapa ;
```

A Figura 7-4 mostra a tabela gerada pela consulta. Cada registro correspondendo a uma área do mapa tem seus valores substituídos pelo termo nebulosos de maior pertinência e a pertinência respectiva. O registro destacado mostra o município com maior número de casos positivos. A representação no mapa(Figura 7-5) mostra, em graus de cinza, os municípios com maior número de casos. É importante observar que

uma mesma faixa de valores como [5,9] pode ter termos nebulosos diferentes, como “Paraíso do Tocantins” e Formoso do Araguaia” contudo com diferentes relevâncias. Por outro lado, dois municípios como “Gurupi” e “Peixe” possuem faixas diferentes [5,9] e [10,15] e um mesmo termo nebuloso com relevâncias diferentes. Podemos com isso extrair um grau de generalização e também de classificação de padrões de acordo com a modelagem feita.

Selected Contours - 41 Recs. Map. TOCANTINS						
Geo	LAMEXA	LAMPOSF	LAMPOSV	LAMPOSM	LAMPOSFV	LAMPOSTOT
ALIANÇA DO NORTE	Baixo(99%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)
ALMAS	Medio(81%)	Baixo(89%)	Medio(86%)	Baixo(100%)	Baixo(100%)	Baixo(81%)
ALVORADA	Baixo(99%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)
APARECIDA DO RIO NEGRO	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)
ARAGUACU	Baixo(90%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)
ARRAIAS	Baixo(95%)	Baixo(89%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(93%)
BARROLANDIA	Baixo(99%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)
BREJINHO DE NAZARE	Baixo(83%)	Baixo(95%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(96%)
CASEARA	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)
COMBINADO	Baixo(96%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)
CRISTALANDIA	Baixo(67%)	Baixo(74%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(81%)
DIVINOPOLIS	Baixo(97%)	Baixo(100%)	Baixo(71%)	Baixo(100%)	Baixo(100%)	Baixo(93%)
DUIRE	Baixo(93%)	Baixo(100%)	Baixo(86%)	Baixo(100%)	Baixo(100%)	Baixo(96%)
FATIMA	Baixo(98%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)
FIGUEIROPOLIS	Baixo(94%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)
FORMOSO DO ARAGUAIA	Medio(72%)	Baixo(89%)	Alto(86%)	Baixo(100%)	Baixo(100%)	Baixo(70%)
GURUPI	Medio(77%)	Baixo(79%)	Alto(86%)	Baixo(100%)	Baixo(100%)	Medio(74%)
LIZARDIA	Baixo(95%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)
MIRACEMA DO TOCANTINS	Baixo(67%)	Baixo(95%)	Baixo(71%)	Baixo(100%)	Baixo(100%)	Baixo(89%)
MIRANORTE	Baixo(98%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)
MONTE DO CARMO	Baixo(97%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)
NATIVIDADE	Medio(96%)	Alto(100%)	Baixo(86%)	Baixo(100%)	Alto(100%)	Alto(100%)
NOVA ROSALANDIA	Baixo(98%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)
NOVO ACORDO	Baixo(88%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)
PALMAS	Medio(93%)	Medio(84%)	Alto(100%)	Baixo(100%)	Baixo(100%)	Medio(89%)
PALMEIROPOLIS	Baixo(99%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)
PARAISO DO TOCANTINS	Medio(92%)	Baixo(84%)	Alto(100%)	Baixo(100%)	Baixo(100%)	Medio(74%)
PARANA	Baixo(71%)	Baixo(95%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(96%)
PEIXE	Medio(98%)	Baixo(95%)	Alto(100%)	Baixo(100%)	Baixo(100%)	Medio(81%)
PIUM	Baixo(81%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)
PONTE ALTA DO BOM JESUS	Baixo(87%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)
PONTE ALTA DO TOCANTINS	Baixo(98%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)
PORTO ALEGRE DO TOCANTINS	Baixo(96%)	Baixo(95%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(96%)
PORTO NACIONAL	Alto(100%)	Medio(74%)	Alto(86%)	Baixo(100%)	Baixo(100%)	Medio(96%)
RIO SONO	Baixo(91%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)
SANTA ROSA DO TOCANTINS	Baixo(85%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)
SANTA TEREZA	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)
SÃO VALERIO DA NATIVIDADE	Medio(71%)	Baixo(79%)	Alto(86%)	Baixo(100%)	Baixo(100%)	Medio(74%)
SILVANOPOLES	Baixo(92%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)
TAGUATINGA	Baixo(79%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)
TOCANTINIA	Baixo(99%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)	Baixo(100%)

Figura 7-4 - Número de casos de cada Plasmodium

A análise anterior avalia os dados de forma absoluta. Vamos verificar agora a condição dos dados de forma relativa, ou seja desejamos saber o índice de ocorrência dos casos em relação à amostra examinada. Uma vez que estas variáveis não pertencem originalmente à base de dados, vamos enviar os resultados para um arquivos separados. A consulta será feita em dois passos. A Tabela 7-1 mostra os resultados obtidos para os percentuais de *Falciparum* e *Vivax*, junto com o total de casos positivos.

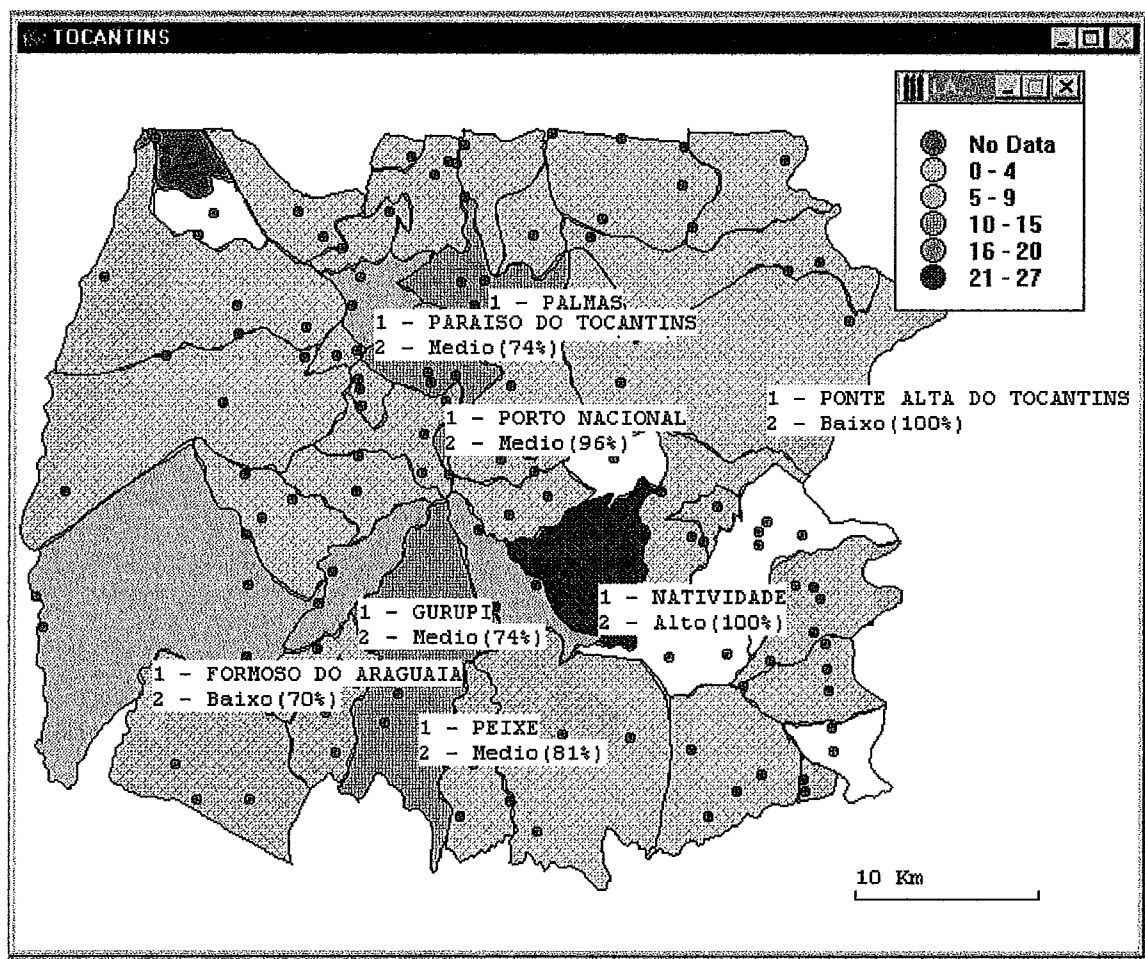


Figura 7-5 - Representação cromática do numero de casos positivos

Utilizamos um modificador no termo “Baixo” ao invés do termo original. Certos tipos de dados contém alguns poucos valores bem altos e concentram grande parte do restante numa faixa de valores mais baixos. Para evitar uma avaliação tendenciosa utilizamos um modificador nos termos. Esta estratégia permitirá aumentar os valores na faixa média.

Comando:

```
SELECT PerFalciparum, PerVivax, PerPositivasTotal
FROM Tocantins
TO Relatorio11;
```

```
SELECT PerMalariae, PerFalcVivax, PerPositivasTotal
FROM Tocantins
TO Relatorio12;
```

Vamos verificar agora as áreas onde temos número elevado de lâminas positivas e existe grande número de casas. Estas áreas são de maior risco epidêmico, pois são mais populosas. Acrescentamos à consulta a possibilidade de retorno da doença causada

pele *Plasmodium vivax*, desta forma, podemos de uma só vez verificar áreas de maior interesse. Limitamos os registros a serem representados em um mínimo de 10%.

<i>PerFalciparum</i>	<i>PerVivax</i>	<i>PerPositivasTotal</i>
0.000000 (100.0%) Muito-Baixo	0.000000 (100.0%) Baixo	0.000000 (100.0%) Muito-Baixo
0.000000 (100.0%) Muito-Baixo	0.000000 (100.0%) Baixo	0.000000 (100.0%) Muito-Baixo
0.000000 (100.0%) Muito-Baixo	0.000000 (100.0%) Baixo	0.000000 (100.0%) Muito-Baixo
0.035714 (80.1%) Muito-Baixo	0.107143 (78.6%) Baixo	0.142857 (57.1%) Medio
0.000000 (100.0%) Muito-Baixo	0.090909 (81.8%) Baixo	0.090909 (54.8%) Muito-Baixo
0.042105 (76.8%) Muito-Baixo	0.063158 (87.4%) Baixo	0.105263 (49.2%) Muito-Baixo
0.013158 (92.3%) Muito-Baixo	0.092105 (81.6%) Baixo	0.144737 (57.9%) Medio
0.000000 (100.0%) Muito-Baixo	0.000000 (100.0%) Baixo	0.000000 (100.0%) Muito-Baixo
0.022222 (87.3%) Muito-Baixo	0.000000 (100.0%) Baixo	0.022222 (87.3%) Muito-Baixo
0.285714 (85.7%) Medio	0.000000 (100.0%) Baixo	0.285714 (85.7%) Medio
0.000000 (100.0%) Muito-Baixo	0.000000 (100.0%) Baixo	0.000000 (100.0%) Muito-Baixo
0.000000 (100.0%) Muito-Baixo	0.000000 (100.0%) Baixo	0.000000 (100.0%) Muito-Baixo
0.000000 (100.0%) Muito-Baixo	0.000000 (100.0%) Baixo	0.000000 (100.0%) Muito-Baixo
0.166667 (66.7%) Medio	0.000000 (100.0%) Baixo	0.166667 (66.7%) Medio
0.021739 (87.5%) Muito-Baixo	0.032609 (93.5%) Baixo	0.054348 (70.8%) Muito-Baixo
0.240506 (96.2%) Medio	0.012658 (97.5%) Baixo	0.341772 (68.4%) Alto
0.072727 (62.4%) Muito-Baixo	0.109091 (78.2%) Baixo	0.181818 (72.7%) Medio
0.000000 (100.0%) Muito-Baixo	0.000000 (100.0%) Baixo	0.000000 (100.0%) Muito-Baixo
0.000000 (100.0%) Muito-Baixo	0.000000 (100.0%) Baixo	0.000000 (100.0%) Muito-Baixo
0.000000 (100.0%) Muito-Baixo	0.000000 (100.0%) Baixo	0.000000 (100.0%) Muito-Baixo
0.098039 (52.0%) Muito-Baixo	0.000000 (100.0%) Baixo	0.098039 (52.0%) Muito-Baixo
0.000000 (100.0%) Muito-Baixo	0.000000 (100.0%) Baixo	0.000000 (100.0%) Muito-Baixo
0.000000 (100.0%) Muito-Baixo	0.000000 (100.0%) Baixo	0.000000 (100.0%) Muito-Baixo
0.038462 (78.7%) Muito-Baixo	0.000000 (100.0%) Baixo	0.038462 (78.7%) Muito-Baixo
0.045161 (75.3%) Muito-Baixo	0.038710 (92.3%) Baixo	0.083871 (57.6%) Muito-Baixo
0.000000 (100.0%) Muito-Baixo	0.000000 (100.0%) Baixo	0.000000 (100.0%) Muito-Baixo
0.000000 (100.0%) Muito-Baixo	0.000000 (100.0%) Baixo	0.000000 (100.0%) Muito-Baixo
0.000000 (100.0%) Muito-Baixo	0.000000 (100.0%) Baixo	0.000000 (100.0%) Muito-Baixo
0.000000 (100.0%) Muito-Baixo	0.000000 (100.0%) Baixo	0.000000 (100.0%) Muito-Baixo
0.000000 (100.0%) Muito-Baixo	0.000000 (100.0%) Baixo	0.000000 (100.0%) Muito-Baixo
0.111111 (47.1%) Muito-Baixo	0.097222 (80.6%) Baixo	0.208333 (83.3%) Medio
0.042254 (76.7%) Muito-Baixo	0.098592 (80.3%) Baixo	0.140845 (56.3%) Medio
0.000000 (100.0%) Muito-Baixo	0.000000 (100.0%) Baixo	0.000000 (100.0%) Muito-Baixo
0.000000 (100.0%) Muito-Baixo	0.000000 (100.0%) Baixo	0.000000 (100.0%) Muito-Baixo
0.000000 (100.0%) Muito-Baixo	0.000000 (100.0%) Baixo	0.000000 (100.0%) Muito-Baixo
0.000000 (100.0%) Muito-Baixo	0.500000 (100.0%) Alto	0.500000 (100.0%) Alto
0.000000 (100.0%) Muito-Baixo	0.000000 (100.0%) Baixo	0.000000 (100.0%) Muito-Baixo
0.019608 (88.7%) Muito-Baixo	0.039216 (92.2%) Baixo	0.058824 (68.7%) Muito-Baixo
0.000000 (100.0%) Muito-Baixo	0.000000 (100.0%) Baixo	0.000000 (100.0%) Muito-Baixo
0.000000 (100.0%) Muito-Baixo	0.000000 (100.0%) Baixo	0.000000 (100.0%) Muito-Baixo
0.000000 (100.0%) Muito-Baixo	0.000000 (100.0%) Baixo	0.000000 (100.0%) Muito-Baixo
0.000000 (100.0%) Muito-Baixo	0.000000 (100.0%) Baixo	0.000000 (100.0%) Muito-Baixo

Tabela 7-1 - Percentuais de casos positivos

Comandos:

```

SELECT      PerPositivasTotal, CasasExistentes, Retorno
FROM        Tocantins
TO          Epidemica
WHERE       (((PerPositivasTotal IS Alto)OR(PerPositivasTotal IS Medio))AND
            ((CasasExistentes IS Alto) OR(CasasExistentes IS Medio )))OR
            (Retorno IS Provavel));

```

```

SELECT      LamPositivasTotal, CasasExistentes
FROM        Tocantins
TO          Mapa
WHERE       (((PerPositivasTotal IS Alto)OR(PerPositivasTotal IS Medio))AND
            ((CasasExistentes IS Alto)OR(CasasExistentes IS Medio )))OR
            (Retorno IS Provavel));
WITH       0.1

```

Na Figura 7-6 podemos ver a representação dos registros selecionados. A maioria dos registros não apresenta valor muito elevado na relevância, contudo o registro correspondente a “DIVINÓPOLIS” tem relevância de 100%. Podemos verificar que o número de lâminas positivas e casas existentes na área é incompatível, dessa forma verificamos que a alta presença de *Plasmodium vivax* força a área a ser selecionada. Por outro lado ainda podemos ver que o número de casas existentes na região é nulo. Isso mostra uma inconsistência dos dados, visto que o número de casos positivos não é nulo. Outras seleções podem ser feitas de modo a avaliar a consistência dos dados.

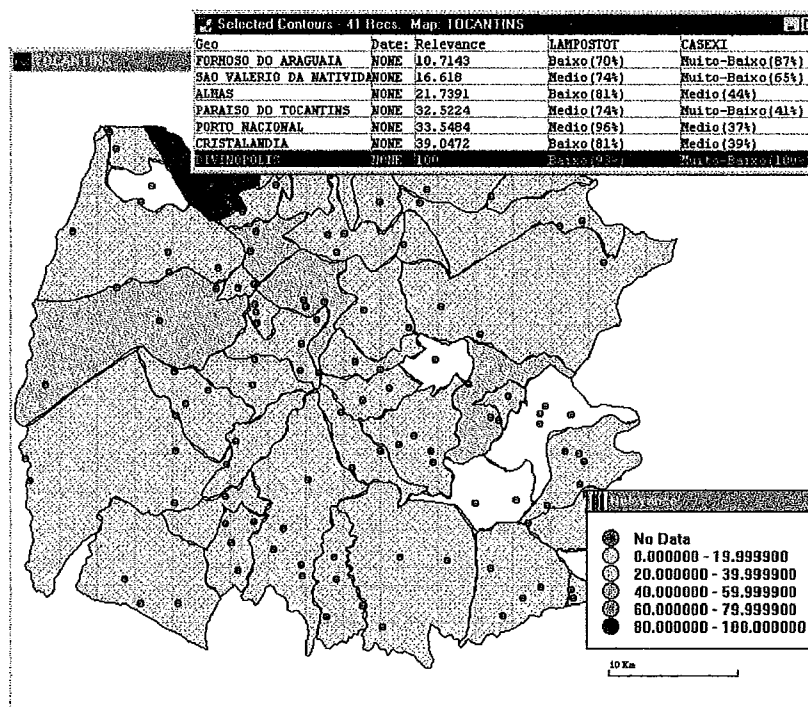


Figura 7-6 - Representação de áreas selecionadas

A vigilância epidemiológica é baseada em sistemas de prevenção. As áreas com maior incidência de casos ou áreas com pouca vigilância devem ser melhor cuidadas. A consulta a seguir testa a variável solução *Visitas* construída a partir da relação entre o número de casas visitadas e existentes. A consulta verifica ainda as áreas onde se reflete maior população em função do número de casas existentes.

Neste exemplo, requisitamos os registros cujo predicado atinge pelo menos 30% e representamos o resultados no mapa da Figura 7-7.

Comandos:

```
SELECT CasasExistentes, PerVisitadas, Visitas
FROM Tocantins
TO Mapa
WHERE ((Visitas IS Aumentar) AND
((CasasExistentes IS Medio) OR CasasExistentes IS Alto))
WITH 0.3;
```

```
SELECT CasasExistentes, PerVisitadas, Visitas
FROM Tocantins
TO Visitas
WHERE ((Visitas IS Aumentar) AND
((CasasExistentes IS Medio) OR (CasasExistentes IS Alto)));
```

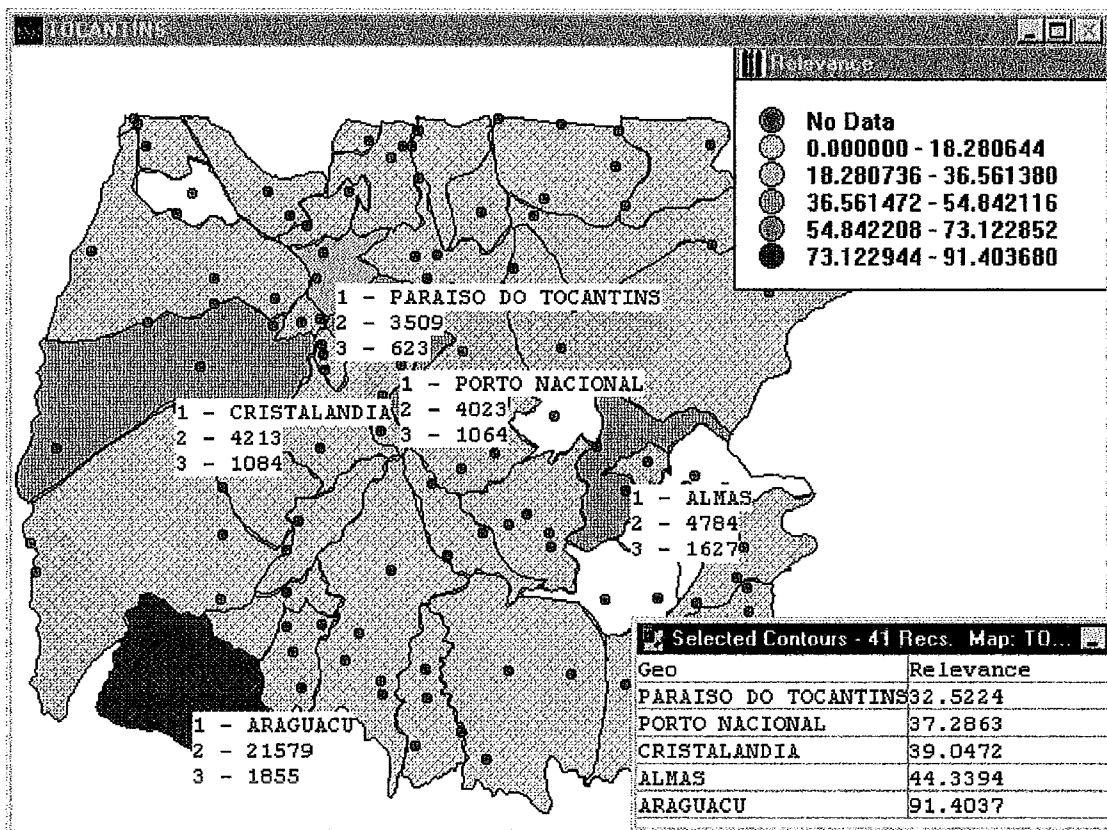


Figura 7-7 - Representação de visitas

Podemos verificar que a consulta realizada mostrou em sua maioria regiões com baixa relevância. O município de “Araguaçu”, contudo, mostra um índice bem alto de relevância, indicando a necessidade de atenção por parte dos órgãos de saúde.

7.3 Conclusões

As consultas realizadas aqui são baseadas nas variáveis coletadas ao longo de regiões do mapa. O cálculo é feito sobre tabelas resultantes de pesquisas, desta forma a modelagem e consulta funcionam similarmente às que seriam feitas num banco de dados convencional. As representações contudo, permitem a possibilidade de mostrar espacialmente a relevância dos predicados das consultas.

A maioria das consultas resultou em registros de baixa relevância. Os dados coletados em regiões de poucos recursos tendem a ser pouco consistentes. Podemos verificar então que o sistema mostra-se útil para verificação dessas inconsistências.

Capítulo VIII

8. Conclusões

O desenvolvimento deste trabalho produziu várias conclusões de interesse. Algumas destas relativas aos trabalhos encontrados sobre o assunto.

Muitos trabalhos são encontrados na literatura sobre bancos de dados nebulosos. Várias implementações são encontradas, contudo não se encontra uma dissertação muito extensa sobre as implementações do cálculo relacional com valores nebulosos. Os algoritmos para a definição das operações de seleção, projeção e principalmente junção não estão muito claramente especificados. Cada autor parece especificar um algoritmo que atenda aos requisitos da aplicação implementada.

A operação de junção traz questões ainda de maior discussão. A operação de junção está relacionada com a determinação da igualdade entre campos de duas relações, entretanto, a comparação entre duas variáveis nebulosas pode ter vários enfoques, como mostrado na seção 2.2.9. Os procedimentos encontrados levam em conta, em geral, bancos de dados já com valores nebulosos, onde a comparação entre termos nebulosos faz-se por tabelas de similaridade(Petry,1996) ou pela existência de termos iguais nos dois atributos(Li,Liu,1990).

Os vários trabalhos de aplicação de sistemas nebulosos em SIGs mostram basicamente dois enfoques. O primeiro acopla um banco de dados nebuloso a uma base de dados espacial, onde o tratamento nebuloso somente se liga à base espacial através dos processos de representação. O segundo enfoque aplica a teoria nebulosa para modelar as incertezas espaciais. Este enfoque tem a maior parte de seus trabalhos ligados a definição das incerteza dos contornos de regiões.

O primeiro enfoque beneficia-se das representações dos dados nebuloso no espaço, melhor visualizando as variáveis do banco. Os trabalhos nesta linha não possuem um aprofundamento muito grande nas capacidades dos bancos de dados nebulosos, não explorando um sistema de regras, por exemplo. Uma das contribuições deste trabalho é a ênfase e aprofundamento nas questões de bancos de dados fornecendo capacidades dedutivas(regras) e oferecendo este módulo para a utilização em SIGs.

Não foram encontrados trabalhos na área de bancos de dados visando modelar dados convencionais de forma nebulosa. Os processos de conversão e tratamento de dados precisos de forma nebulosa(*fuzzificação*) é mais explorado em sistemas de controle(Cox,1992,1994a,1994b).

A modelagem de dados precisos em nebulosos não se mostrou uma tarefa trivial. Várias questões surgem quando realizamos a *fuzzificação* de dados precisos. Deve-se primeiramente entender as características conceituais dos dados a serem modelados. Os termos lingüísticos que serão usados devem se adequar em forma e valor ao domínio das variáveis que serão modeladas. O sistema apresentado contribui para esta tarefa fornecendo, diferentemente da maioria dos sistemas, uma forma de adequar os termos nebulosos a uma faixa normalizada do domínio, permitindo sua utilização em várias variáveis mantendo uma consistência conceitual.

Outro problema que foi notado na modelagem dos dados foi a má distribuição dos dados dentro do domínio da variável. Em alguns casos os dados se concentravam num ponto extremo do domínio em função da existência de alguns poucos valores no extremo oposto. Isto ocasionava a classificação da maioria dos valores em apenas um termo nebuloso.

A inconsistência dos dados também deve ser tratada. Um dos problemas comuns em arquivos coletados, a partir de áreas com poucos recursos técnicos, é encontrarmos variáveis somente com valores nulos ou com valores fora do domínio. O sistema de regras possibilita um teste destas inconsistências, contudo elas também poderão ser geradas por regras em desacordo com os dados do modelo. De todo modo, podemos evidenciar as inconsistências através dos termos nebulosos especiais de indecisão e indefinição. Isso contribui para a avaliação e detecção de inconsistências na base de dados por parte dos analistas.

Outra questão a ser ressaltada com relação a modelagem dos dados é a interação e comparação entre os dados precisos e nebulosos(Capítulo 2). Não se encontra uma formalização sobre os métodos usados. Vários autores sugerem técnicas diferentes. Pensamos que o sistema de modelagem deve fornecer o máximo possível de modelos a serem empregados de forma a melhor adequar a ferramenta a cada problema específico.

Os trabalhos que aplicam os sistemas nebulosos na modelagem espacial fazem propostas específicas sobre problemas específicos, estando em geral voltados para o tratamento de contornos com imprecisão. A filosofia do sistema apresentado aqui se baseia em converter os dados em seu formato original para um modelo nebuloso. Apresentamos aqui uma forma de modelar objetos espaciais da mesma forma como poderíamos modelar dados convencionais.

O sistema desenvolvido atendeu às expectativas do proposto, os resultados apresentados mostram a sua flexibilidade e os exemplos apresentados mostram sua utilidade dentro das áreas de análise da exploração de conhecimento.

O modelo espacial proposto deverá ser ainda implantado, incluindo extensões na linguagem desenvolvida, para a modelagem deste tipo de entidade.

O sistema foi desenvolvido de forma a permitir várias expansões e alterações futuras. Sugerimos a expansão das interfaces de entrada e saída, incluindo principalmente as de comunicação em rede. Alguns protótipos já foram testados utilizando comunicação em rede via *softwares* de simulação de máquinas paralelas.

O sistema de consulta pode ser ainda implementados de múltiplas formas para permitir maior utilização dos recursos do SGBD ao qual o sistema esteja acoplado.

Seria interessante uma expansão das capacidades de modelagem, com novas classes de variáveis, termos e operadores, além disso, poderíamos também desenvolver regras utilizando proposições condicionais qualificadas (seção 2.3.1). Isto permitiria testes de qualidade de modelos, bem como a implementação de processos adaptativos para criação de modelos nebulosos.

Capítulo IX

9. Referências Bibliográficas

1. Aho AV, Sethi R, Ullman JD. *Compilers Principles, Techniques, and Tools*. Reading, Massachusetts: Guardino K and DeWolf JT, editors. Addison-Wesley; 1988;
2. Alesheikh AA and Li R. "Rigorous Uncertainty Models of Line and Polygon Objects in GIS_". *In:GIS/LIS 1996: CONFERENCE*. 1996
3. Altman D. "Fuzzy set theoretic approaches for handling imprecision in spatial analysis" *International Journal of Geographical Information Systems* 8(3):271-89. 1994;
4. Bailey NTJ. *Biomathematics of Malaria*. 1 ed. Taylor and Francis, editor. 1982;
5. Banai R. "Fuzziness in Geographical Information Systems: contributions from the analytic hierarchy process" *International Journal of Geographical Information Systems* 7(4):315-29. 1993;
6. Batty P. "Exploiting relational database technology in a GIS" *Computers and Geosciences* 18(4):453-62. 1992;
7. Beckmann N, Kriegel H, Schneider R, and Seeger B. "The R*-Tree: An Efficient and Robust Method for Points and Rectangles". *In:Proceedings of the 1990 ACM SIGMOD - International Conference on Management of Data*. pp.322-331. Atlantic City,NJ. 1990
8. Belchior, A.D., *Um Modelo Fuzzy para Avaliação da Qualidade de Software*, D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 1997;
9. Bernardes MC, Wu S-T, and D'Ottaviano IML. "Fuzzy sets on drawing fair plane curves". *In:Anais do VIII SIBGRAPI*. I. pp.87-94. 1995
10. Berthier ANR and Muntz R. "F-G: A Fuzzy Algebra for Aproximate in Databases". *In:X Brazilian Symposium on Databases*. pp.365-376. Recife, Brazil. 1995
11. Bosc P and Pivert O. "Fuzzy querying in conventional databases". *Anonymous Fuzzy Logic for the Management of Uncertainty*. 32. 1997;
12. Bosc P and Pivert O. "Extending SQL Retrieval Features for the Handling of Flexible Queries". *In:Dubois D, Prade H, and Yager. Fuzzy Information Engineering: A Guided Tour of Applications*. 15. New York. Wiley Computer Publishing, John Wiley & Sons, INC. 1997;

13. Braga AL, Nobre FFP, Araújo Jr. NS, Oliveira PRV, Pinheiro RS. "PROJETO - SIGEPI - Sistema de Informação Geográfica para uso em Epidemiologia" *IV Congresso Brasileiro de Informática em Saúde* 2(1):171-5. 1994;
14. Buckles BP, Petry FE. "A fuzzy representation of data for relational databases" *Fuzzy Sets and Systems* 7:213-26. 1982;
15. Burrough PA. Principles of Geographical Information Systems for Land Resource Assessment. Clarendon Press, editor.Oxford; 1986;
16. Burrough PA. "Fuzzy mathematical methods for soil survey and land evaluation" *Journal of Soil Science* 40:477-92. 1989;
17. Burrough PA. "Are GIS data structures too simple minded?" *Computers and Geosciences* 18(4):395-400. 1992;
18. Burrough PA. "Development of intelligent geographical information systems" *International Journal of Geographical Information Systems* 6(1):1-11. 1992;
19. Burrough PA, MacMillan RA, Van Deursen W. "Fuzzy classification methods for determining land suitability from soil profile observations and topography" *Journal of Soil Science* (43):477-92. 1992;
20. Burrough PA, Frank AU. Geographic Objects with Indeterminate Boundaries. London: Burrough PA and Frank AU, editors.Taylor and Francis; 1996;
21. Chang L and Burrough PA. Fuzzy reasoning: A new quantitative aid for land evaluation. *Soil Survey and Land Evaluation*. 7(2). 69-80. 1987;
22. Chrisman NR. "Modeling errors in overlaid categorical maps". In:Goodchild M and Gopal S. The accuracy of spatial databases. London. Taylor and Francis. 1989;
23. Codd EF. "A Relational Model of Data for Large Shared Data Banks" *Communications of ACM* 6:377-87. 1970;
24. Coleman A. "Boundaries as a framework for understanding land-use patterns". In:Kishimoto H and Kumerly and Frey. Geography and its Boundaries. Zurich. Kummerly and Frey. 1980;
25. Cox E. The Fuzzy Systems Handbook. Boston: AP Professional; 1994;
26. Cox E. Fuzzy Logic for Bussiness and Industry. 1994;
27. Cox E. "Fuzzy Fundamentals" *IEEE Spectrum* 18 1992;

28. Davidson DA, Theocharopoulos SP, Blosma RJ. "A land evaluation project in Greece using GIS and based on boolean and fuzzy set methodologies" *International Journal of Geographical Information Systems* 8(4):369-84. 1994;
29. Davis Jr CA, Borges KAdV. "Object-Oriented GIS in Praticce" *URISA Journal* :786-95. 1994;
30. Dijkmeijer J and Hoop S. "Topologic Relations Between Fuzzy Areas Objects". *In:Proceedings of the 7th International Symposium on Spatial Data Handling*. pp.377-393. 1996
31. Dubois D, Prade H. "Mesuring properties of fuzzy sets: A general technique and its use in fuzzy query evaluation" *Fuzzy Sets and Systems* 38:137-52. 1990;
32. Dunn R. Statistical Mapping. *The American Statistician*. 41(2). 153-156. 1987;
33. Dutta S. "Qualitative Spatial Reasoning: A Semi-quantitative Approach Using Fuzzy Logic". *In: Buchmann A, Günther O, Smith TR, and Wang F. Design and Implementation of Large Spatial Databases*. Springer-Verlag. 1989;
34. Dutta S. "Approximate Spatial Reasoning: Integrating Qualitative and Quantitative Constraints" *International Journal of Approximate Reasoning* 5:307-30. 1991;
35. Dutton G. "Handling Positional Uncertainty in Spatial Databases". *In: Proceedings of the 5th International Symposium on Spatial Data Handling*. 2. pp.460-469. Charleston, South Carolina. 1992
36. Egenhofer JJ. "Why Not SQL!" *International Journal of Geographical Information Systems* 6(2):71-85. 1992;
37. Egenhofer MJ and Herring JR. "High-Level Spatial Data Structures for GIS". *In: Maguire DJ, Goodchild MF, and Rhind DW. Geographic Information Systems: Principles and Aplications*. 16. Ny,USA. Longman Scientific & Technical. 1991;
38. Egenhofer MJ. Object-Oriented GISs: The Principles.[Unpublished] 1996;
39. Fisher PF and Pathirana D. "Evaluation of fuzzy membership of land cover classes in suburban areas of North-East Ohio.". *In: ASPRS-ACSM Fall Convention*. pp.125-132. 1989
40. Fisher PF. "Knowledge-based approaches to determining and correcting areas of unreliability in geographic databases". *In: Goodchild M, Gopal S, and Taylor and Francis. The Accuracy of spatial databases*. London. Taylor and Francis. 1989;

41. Fisher PF and Pathirana D. Evaluation of fuzzy membership of land cover classes in the suburban zones. *Remote Sensing and the Environment*.(34). 121-132. 1990;
42. Frank AU. "Spatial concepts, geometric data models, and geometric data structures" *Computers and Geosciences* 18(4):409-17. 1992;
43. Franklin WmR. "Computer Systems and Low-Level Data Structures for GIS". In:Maguire DJ, Goodchild MF, and Rhind DW. Geographic Information Systems: Principles and Applications. 15. Ny,USA. Longman Scientific & Technical. 1991;
44. Goodchild MF. "Geographical data modeling" *Computers and Geosciences* 18(4):401-8. 1992;
45. Guttman A. "The R-Trees:A dynamic index structure for spatial searching". In:Proceedings of the 1990 ACM SIGMOD - International Conference on Management of Data. pp.47-57. 1984
46. Hall G, Wang F, and Subaryono S. Comparison of boolean and fuzzy classification methods in land suitability analysis using GIS. *Environment and Planning B*. 24 497-516. 1991;
47. Herring JR. "TIGRIS: A data model for an object-oriented Geographic Information System" *Computers and Geosciences* 18(4):443-52. 1992;
48. Jiang B and Kainz W. Fuzzy Overlay Analysis With Linguistic Degree Terms.[Unpublished] 1996;
49. Johnson, S.C. AT&T Bell Laboratories, editor. Yacc - yet another compiler compiler. N.J. Murray Hill. 1975; 32. Computing Science Technical Report.
50. Klir GG, Yuan B. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Upper Saddle River, New Jersey: Prentice Hall; 1995;
51. Klir GG, Yuan B. *Fuzzy databases and information retrieval systems*. 1 ed. New Jersey: Prentice-Hall; 1995;
52. Kolia VJ, Voliotis A. "Fuzzy reasoning in the development of geographical information systems. FRIS: a prototype soil information system with fuzzy" *International Journal of Geographical Information Systems* 5(2):209-23. 1991;
53. Kosko B and Isaka S. Fuzzy Logic. *Scientific American*. 269(1). 62-67. 1993;
54. Lacroix M and Lavency P. "Preferences: Putting more knowledge into queries.". In:Proceedings of the 13 VLDB Conference. Brighton. 1987

55. Leung Y. Towards a flexible framework for recognition. *Environment and Planning. A*(16). 203-215. 1984;
56. Leung Y. On the Imprecision of Boundaries. *Geographical Analysis*. 19 125-151. 1987;
57. Li D, Liu D. A Fuzzy PROLOG Database System. Tauton,Somerset,England: Foulk PWDr, editor.Research Studies Press LTD.; 1990;
58. Maguire DJ. "An Overview and Definition of GIS". In:Maguire DJ, Goodchild MF, and Rhind DW. Geographic Information Systems: Principles and Applications. 1. Ny,USA. Longman Scientific & Technical. 1991;
59. Maguire DJ and Dangermond J. "The Functionality of GIS". In:Maguire DJ, Goodchild MF, and Rhind DW. Geographic Information Systems: Principles and Applications. 21. Ny,USA. Longman Scientific & Technical. 1991;
60. Maguire DJ, Raper JF. "Design models and functionality in GIS" *Computers and Geosciences* 18(4):387-94. 1992;
61. Maguire DJ. "What is an Object-Oriented GIS?". In:Proceedings of the Fourteenth Annual ESRI User Confrence. pp.76-89. 1994
62. Mediano MR, Casanova MA, and Dreux M. "A Family of storage Methods for Geographic Data". In:9no Simpósio Brasileiro de Banco de Dados. pp.310-323. 1995
63. Middelkoop H. "Uncertainty in a GIS: a test for quantifying interpretation output" *ITC-Journal* 3:225-33. 1990;
64. Milne P, Minton S, Smith JL. "Geographical object-oriented databases-a case study" *International Journal of Geographical Information Systems* 7(1):39-55. 1993;
65. Molenaar M. "A syntax for the representation of fuzzy spatial objects". In:Molenaar M and Hoop S. Advanced Geographic Data Modeling. Netherlands. Netherlands Geodetic Commission. 1994;
66. Molenaar M. "A Syntatic Approach for Handling the Semantics of Fuzzy Spatial Objects". In:Burrough PA and Frank AU. Geographic Objects With Indeterminate Boudaries. London. Taylor and Francis. 1995;
67. Molenaar M. "The Extensional Uncertainty of Spatial Objects". In:Proceedings of the 7th International Symposium on Spatial Data Handling. pp.377-393. 1996

68. Morehouse S. "The ARC/INFO Geographic Information System" *Computers and Geosciences* 18(4):435-41. 1992;
69. Nobre FF and Carvalho MS. Spatial and Temporal Analysis of Epidemiological Data.[Unpublished] 1996;
70. Nobre FF, Macedo MMA, Pires MLM et al. Sig-Malaria: Um sistema de Informação Geográfica para Controle de Malária. IDRC Reports. 17-19. 1993;
71. Nola AD, Pedrycz W, Sessa S. "Fuzzy relational structures: The state-of-art" *Fuzzy Sets and Systems* 75:241-62. 1995;
72. Openshaw S. "Developing Appropriate Spatial Analysis Methods for GIS". In:Maguire DJ, Goodchild MF, and Rhind DW. Geographic Information Systems: Principles and Applications. 25. Ny,USA. Longman Scientific & Technical. 1991;
73. Pequet D. "Data Structures for a Knowledge-Based Geographic Information System". In: Proceedings of the International Symposium on Spatial Data Handling. 2. pp.372-391. Zurich, Switzerland. 1984
74. Petry F and Cobb M. "Fuzzy Querying Binary Relationships in Spatial Databases". In:IEEE International Conference on Cybernetics and Society. pp.378-385. 1995
75. Petry F. Fuzzy Databases. Boston/Dordrecht/London: Zimmermann HC, editor.Kluwer Academic Publishers; 1996;
76. Portier MMM. Definition of CIGALES: A geographical Information System Query Language.[Unpublished] 1996;
77. Ramirez, M.R., Sistemas Gerenciadores de Bancos de Dados para Geoprocessamento, M.Sc., Universidade Federal do Rio de Janeiro/COPPE, 1994;
78. Robinson V. "On the human definition of the dry boundaries". In:Kishimoto H and Kumerly and Frey. Geography and its Boundaries. Zurich. Kummerly and Frey. 1980;
79. Robinson V. Some Implications of Fuzzy Set Theory applied to Geographic Databases. *Computing, Environment and Urban Systems*.(12). 89-98. 1988;
80. Robinson V. "Implications of Fuzzy Set Theory for Geographic Databases". In:Proceedings AUTOCARTO 7 Conference. 12. pp.89-98. 1988

81. Robinson V. Interactive MACHINE Acquisition of a Fuzzy Spatial Relation. *Computers and Geosciences*. 6 857-872. 1990;
82. Ross TJ. *Fuzzy Logic with Engineering Applications*. New York: Cox L and Morriss JM, editors. McGraw-Hill, Inc.; 1995;
83. Saaty RW. The Analytic Hierarchy Process - What is it and how it is used. *Mathematical Modeling*. 9 161-176. 1987;
84. Samet H. *The Design and Analysis of Spatial Data Structures*. New York: Harrison MA and Addison-Wesley, editors. Addison-Wesley; 1990;
85. Sheno S, Melton A, Fan LT. "An equivalence classes model of fuzzy relational databases" *Fuzzy Sets and Systems* 38:153-70. 1990;
86. Strauch, J.C.M. and Mattoso, M.L.d.Q. COPPE - PESC, editor. *Sistemas de Informações Geográficas Orientados a Objetos*. Rio de Janeiro. COPPE -PESC. 1994; ES-331/94. p.1 ES-331/94.
87. Tahani V. "A fuzzy model of document retrieval systems". *In: Inf.Proc. & Manag.* 12. pp.177-187. 1976
88. Tahani V. "A conceptual framework for fuzzy query processing: A step toward very intelligent database systems.". *In: Inf.Proc. & Manag.* 13. pp.289-303. 1977
89. Times VC and Salgado AC. "Uma modelagem Orientada a Objetos para Aplicações Geográficas". *In: IX Simposio Brasileiro de Banco de Dados*. I. pp.293-309. Sao Carlos. 1994
90. Wang F, Hall G, Subaryono S. "Fuzzy Information Representation and Processing in Conventional GIS software: Database Design and Application" *International Journal of Geographical Information Systems* 4(3):261-83. 1990;
91. Wang F. Fuzzy supervised classification of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*.(28). 194-201. 1990;
92. Wang F. "Towards a natural language user interface: an approach of fuzzy query" *International Journal of Geographical Information Systems* 8(2):143-62. 1994;
93. Wang F, Hall G. "Fuzzy representation of geographical boundaries in GIS" *International Journal of Geographical Information Systems* 10(5):573-90. 1996;
94. Worboys MF. "Object-Oriented approaches to geo-referenced information" *International Journal of Geographical Information Systems* 8(4):385-99. 1994;

95. Worboys MF, Yearsley C, Story P, Jayawardena DPW, and Bofakos P. "Computational support for spatial information handling: models and algorithms". In: Worboys MF. Innovations in GIS. 6. London. Taylor & Francis. 1994;
96. Zadeh L. "Fuzzy Logic" *IEEE Computer* 1988;
97. Zadeh LA. "Fuzzy Sets" *Information and Control* 8(8):338-53. 1965;

Apêndice A

10. Gramática utilizada no Parser do sistema

```

language : BEGIN
          DefTables
          DefComp
          DefUseHedges
          DefUseTerms
          DefRules
          DefQueries
          END ;

DefTables:    CREATE CrteTable ;
              | DefTables CREATE CrteTable ;

DefComp:      CREATE CrteComp ;
              | DefComp CREATE CrteComp ;

DefUseHedges: USE UseHedge ;
              | DefUseHedges USE UseHedge ;

DefUseTerms: USE UseTerm ;
              | DefUseTerms USE UseTerm ;

DefRules:     IF Proposition THEN Condition ;
              | DefRules IF Proposition THEN Condition
              ;

DefQueries:   CrteQuery ;
              | DefQueries CrteQuery ;

/*****Criacao de Tabelas*****/

CrteTable    : TABLE IDNAME FROM IDARQNAME IntType
              | TABLE IDNAME AS IDARQNAME IntType
IntType      : ARQTEXT | BDE IDARQNAME | SIG | GOA

/*****Criacao de componentes*****/

CrteComp:    CrteVar
              | CrteVarSol
              | CrteTerm
              | CrteHedge

/*****Criacao de variaveis*****/

CrteVarSol:  SOLVAR INTEGER IDNAME DOMAIN IDREALVALUE ,
IDREALVALUE
              | SOLVAR REAL IDNAME DOMAIN IDREALVALUE ,
IDREALVALUE

CrteVar:     LINGVAR INTEGER IDNAME CrteLingDef
              | LINGVAR REAL IDNAME CrteLingDef

```

```
CrteLingDef:  DOMAIN IDREALVALUE , IDREALVALUE
             | OVER IDNAME AS IDNAME
             | CrteLingDef DOMAIN IDREALVALUE , IDREALVALUE
             | CrteLingDef OVER IDNAME AS IDNAME
```

*****Criacao de Termos Linguisticos*****

```
CrteTerm:    LINGTERM IDNAME AS MShipType ALFA
             IDREALVALUE
             | LINGTERM IDNAME AS MShipType
```

```
MShipType:  | TRIANGLE ( IDREALVALUE , IDREALVALUE ,
             IDREALVALUE )
             | TRAPEZ ( IDREALVALUE, IDREALVALUE, IDREALVALUE,
             IDREALVALUE )
             | GAUSS ( IDREALVALUE , IDREALVALUE ,
             IDREALVALUE )
```

*****Criacao de Modificadores*****

```
CrteHedge:   HEDGE IDNAME AS CONCENTRATOR IDREALVALUE
             | HEDGE IDNAME AS INTENSIFICATION IDREALVALUE
             | HEDGE IDNAME AS DILATATION IDREALVALUE
```

*****Aplicacao de modificadores e termos*****

```
UseHedge:    HEDGE IDNAME IN IDNAME
UseTerm:     LINGTERM IDNAME IN IDNAME
```

*****Definicao de regras*****

```
Proposition: ( Proposition AND Proposition )
             | ( Proposition OR Proposition )
             | NOT Proposition
             | ( Proposition )
             | Condition
```

```
Condition:   IDNAME IS IDNAME
             | IDNAME IS GREATER IDNAME
             | IDNAME IS LESS IDNAME
             | IDNAME IS (IDNAME) IDNAME
             | IDNAME IS EQUAL IDNAME
```

*****Definição e execução de consultas*****

```
CrteQuery:  SELECT AttList
            FROM TableList
            TO IDNAME
            | SELECT AttList
            FROM TableList
            TO IDNAME
            WHERE Proposition
            | SELECT AttList
            FROM TableList
            TO IDNAME
```

```
      WITH      IDREALVALUE
|  SELECT      AttList
      FROM      TableList
      TO        IDNAME
      WHERE     Proposition
      WITH      IDREALVALUE
```

```
AttList:  IDNAME
| AttList , IDNAME
```

```
TableList: IDNAME
| TableList , IDNAME
```