

**PROLOG-CC: Um Sistema Pericial Difuso
Aplicado à Ciência do Solo**

Jucilene M. Siqueira
jucilenesiqueira@live.com.pt

Vitor Rocio
DCeT Universidade Aberta.
vjr@univ-ab.pt

Resumo

As técnicas da inteligência artificial são ferramentas computacionais apropriadas para modelar processos de transformação que ocorrem na natureza. Tal fato sugere o desenvolvimento de um sistema pericial suportado pelo paradigma da programação em lógica, baseado em regras com raciocínio difuso, possibilitando captar o conhecimento dos peritos. O sistema pericial proposto neste artigo consiste em regras que simulam o conhecimento e o processo de raciocínio do perito humano. São descritos os processos de codificação, inferência e descodificação de um sistema pericial difuso aplicável à análise de fertilidade dos solos. A estimativa do valor final é calculada mediante o mecanismo de agregação de regras denominado Descodificador de Peso Ordenado – DPO. Este sistema pericial é um protótipo que pode ser expandido para sistemas de maior complexidade como, por exemplo, a engenharia de biosistemas, visando automatizar as análises periciais.

Palavras-chave: Sistemas periciais, PROLOG, lógica difusa, automatização, engenharia de biosistemas.

Abstract

Artificial intelligence techniques are appropriate computational tools for modelling transformation processes that occur in nature. That fact alone suggests the development of an expert system supported by the logic programming paradigm, based on fuzzy reasoning rules, which enables expert knowledge encoding. The expert system proposed in this paper consists of rules that simulate the knowledge and reasoning processes of an human expert. We describe the encoding, inference and decoding processes of a fuzzy expert system that applies to the analysis of soil fertility. The estimated final value is calculated through the mechanism of rule aggregation called Ordered Weight Decoder - OWD. This expert system is a prototype that can be extended to more complex systems such as, for example, biosystems engineering, aiming to automate the expert analysis.

Keywords: Expert systems, PROLOG, fuzzy logic, automation, biosystems engineering

1. Introdução

No campo da investigação, o homem procura modelos matemáticos para descrever a engenharia dos sistemas naturais, no intuito de controlar o meio ambiente. Tais sistemas abrangem principalmente processos de transformação. Na maioria dos casos, estes processos são tidos como sistemas complexos por apresentarem um comportamento não linear, muitas vezes com falhas e ambiguidades, comportando-se como um sistema naturalmente difuso. Em particular, os processos ligados à ciência dos solos, em que a composição e características de um solo determinam a sua fertilidade agrícola, configuram-se em modelos transformacionais. Seguramente, o raciocínio humano, no intuito de interpretar e controlar estes modelos, segue o mesmo padrão.

Máquinas de simulação visando repetir os processos de transformação vêm sendo aplicadas intensamente na engenharia de biosistemas. Estas máquinas podem interpretar automaticamente estes modelos, adotando as técnicas da inteligência artificial (IA), mais precisamente os sistemas periciais. Para estes processos de transformação, torna-se importante construir uma ferramenta computacional que seja capaz de realizar simulações a partir da interpretação do raciocínio do perito.

A modelação fundamentada numa abordagem simbólica é uma opção indicada, visto que as informações podem ser representadas por sequências de símbolos, tais como, fatos e regras lógicas, podendo ser implementados numa linguagem de programação em lógica (PROLOG). Em muitas situações, o raciocínio pericial é difuso, ou seja, as afirmações sobre os objectos são vagas. Por exemplo, dizer que uma pessoa é alta não nos dá exactamente a altura dessa pessoa, mas poderá ser suficiente para concluir que será admitida na Força Aérea. De um modo geral, a linguagem PROLOG opera sob uma abordagem clássica da lógica, necessitando possuir mecanismos de inferência que processem valores de verdade parciais, de acordo com a lógica difusa. Portanto, o objetivo deste trabalho é mostrar a utilização do PROLOG em sua capacidade de simular o raciocínio difuso do perito no âmbito da ciência dos solos, definindo um modelo que designamos por PROLOG Convencional Capacitado – PROLOG-CC, para diferenciar-se do sistema PROLOG convencional.

2. Sistemas periciais difusos

Grande parte da base de conhecimento adquirido a fim de modelar processos de transformação ou fenómenos naturais é formada por um conjunto de termos linguísticos que classificam grupos com parâmetros de valores aproximativos.

A lógica difusa serve de base para os modelos de raciocínio que são aproximados ou não exatos. Os conceitos de lógica difusa podem ser utilizados para traduzir em termos matemáticos a informação imprecisa contida em frases expressas em linguagem natural, tais como *muito*, *pouco*, *quase*, etc., permitindo graduações na pertinência de um elemento a um dado conjunto difuso, ou seja, de calcular a possibilidade de um elemento pertencer com maior ou menor intensidade àquele conjunto.

Matematicamente, um conjunto difuso é definido como um subconjunto A contido no conjunto universo X e pode ser caracterizado em forma de pares ordenados, da seguinte maneira: $A = (x, \mu(x))$ para todo $x \in X$ onde $\mu(x)$ é a função de associação conhecida como grau de pertinência de x em A , isto é, uma função de graduação do elemento x em A .

Normalmente, o valor $\mu(x)$ é um número real que varia entre 0 e 1, sendo 1 o valor do membro que pertence completamente ao conjunto e, 0 o valor cujo elemento não pertence ao conjunto. Os graus de pertinência de x ser membro de um conjunto A refletem uma ordem não baseada em probabilidade, mas na possibilidade de x pertencer a A (Zadeh, 1965). Por exemplo, na figura 1 pode ser visualizado o grau de pertinência de um elemento do conjunto A associando às diferentes tonalidades de cinzas. Cada tonalidade de cinza representa um valor ($\mu(x)$) ou grau de pertença aos conjuntos A e B .

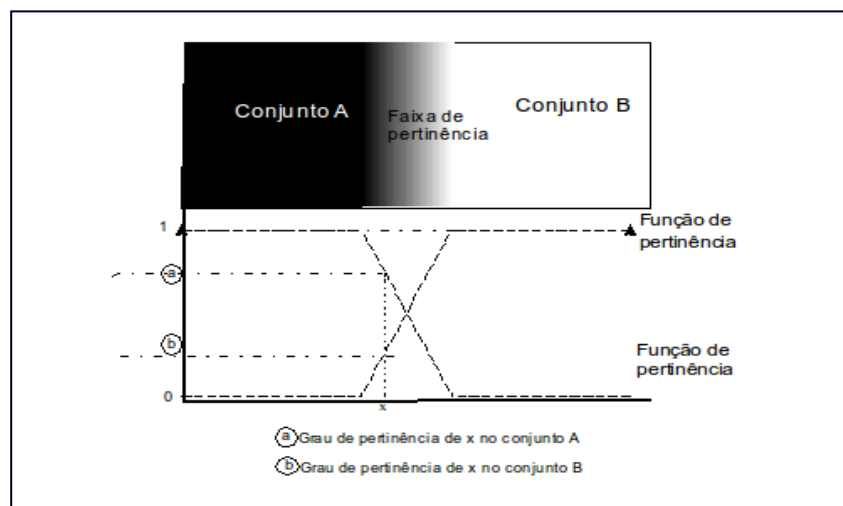


Figura 1 – Representação gráfica de funções de pertinência

3. Raciocínio difuso

O processo de raciocínio difuso que simula processos de transformação de modo automático baseia-se em três fases principais: codificação, inferência e decodificação.

Por codificação, entende-se como o processo de descrição dos conjuntos através de funções de pertinência, e da mesma forma, entende-se por decodificação como o processo de fusão destes conjuntos. A inferência é o processo de raciocínio que conecta a fase codificação à decodificação, onde as atividades de codificação ocorrem como um transmissor, enquanto as atividades de decodificação como um receptor.

Na *codificação*, os valores de entrada, fornecidos no formato numérico, são convertidos em valores linguísticos. Na fase de *inferência*, os valores linguísticos, resultados da *codificação*, são combinados de forma a gerar valores linguísticos de saída de acordo com o objetivo do sistema. Por fim, na *descodificação*, os valores de saída são, novamente, convertidos em valores numéricos que serão casualmente fornecidos a outros dispositivos do sistema.

4. PROLOG em sistemas difusos

Ao longo da história da IA foram várias as propostas de novas linguagens de programação, com complexidade cada vez mais elevada. Uma linguagem é um meio de comunicação por excelência, através da qual se descrevem os agentes, os seus mundos envolventes e as mais diversas interações: os objetos, as funções e as relações (Bratko, 2001). O desenho de um

agente exige uma linguagem de programação adequada ao processamento simbólico, isto é, à manipulação de relações e de outras estruturas simbólicas capazes de nos darem conta da complexidade dos objetos que existem no nosso mundo (Coelho, 1995). Sendo assim, o PROLOG é uma das principais linguagens usadas por pesquisadores em IA, com muitas aplicações desenvolvidas neste campo, especialmente na forma de sistemas periciais servindo de programas que raciocinam a solução de problemas complexos usando regras (Bramer, 2005).

A linguagem PROLOG permite representar adequadamente o conhecimento, especificar formalmente os programas, formalizar bases de dados, e comunicar em língua natural (Coelho, 1995). Ao contrário das linguagens de programação convencionais, que não possuem estruturas básicas para codificar relacionamentos lógicos, em PROLOG, é fácil atribuir relações entre objetos (Henderson, 2007).

A introdução da lógica difusa dentro da programação em lógica promoveu o desenvolvimento de vários modelos PROLOG. Estes modelos fazem parte do mecanismo de inferência do PROLOG com a variante difusa, que está apta a manusear verdade parcial.

A maioria dos sistemas PROLOG para conjuntos difusos aplicam a resolução difusa introduzida por (Lee, 1972). São exemplos o Prolog-ELF system (Ishizuka, 1985) e o F-Prolog language (Li, 1990). Entretanto, não existe nenhum método em comum para decodificação PROLOG. Não existe um acordo sobre qual lógica difusa deveria ser usada. A maioria deles usa a lógica do min-max (modelagem por operações de conjunção e disjunção) e outros sistemas usam a lógica Lukasiewicz. A propósito, a semântica de um programa f-Prolog é similar ao programa PROLOG convencional com exceção de alguns fatos serem associados a valores de verdade e de algumas regras conterem o operador implicação (\rightarrow) associado a valores que traduzem sua força implicante.

O sistema, quando usa a lógica do min-max, possui a mesma estrutura básica e as características da linguagem PROLOG. Suporta quase todas as definições da lógica clássica, mas interpretada como uma lógica de múltiplos valores (Guilherme, 1990).

A lógica Lukasiewicz providencia uma estrutura formal no sistema PROLOG do ponto de vista puramente matemático, enfatizando o desenvolvimento de extensões PROLOG para lógica de múltiplos valores, baseado em uma rigorosa estrutura lógica. Portanto, se situa em oposição às abordagens heurísticas. É esperado obter resultados sólidos e completos (Klawonn, 1994).

5. O modelo PROLOG-CC

A partir do exposto acerca dos modelos citados em literaturas, foi possível reunir num único modelo as distintas características, tais como, lógica max-min e a lógica Lukasiewicz, em PROLOG convencional, capacitando-o a efetuar o raciocínio difuso, o que designamos neste trabalho como PROLOG Convencional Capacitado – PROLOG-CC.

A principal vantagem do PROLOG-CC consiste em que os interpretadores e compiladores PROLOG podem mutuamente processar o raciocínio clássico e difuso de forma consistente e são disponíveis em muitos sistemas. E, assim, os sistemas periciais podem ser facilmente construídos na linguagem PROLOG-CC, que está capacitada em processar o raciocínio difuso

com a inclusão de cláusulas difusas, predicados difusos e regras difusas.

PROLOG-CC é programado da mesma forma que a linguagem PROLOG, seguindo a ideia de programação em lógica. Portanto, serão utilizados mecanismos de inferência com base num interpretador em PROLOG, onde as fórmulas manusearão o conceito de verdade parcial, pelo cálculo do grau de pertinência.

A diferença básica entre o PROLOG e o PROLOG-CC consiste em que, no PROLOG, a sua base de dados é composta de fatos e regras, com grau de pertinência igual a 1. No PROLOG-CC, a base de dados é construída por fatos e regras que estão associados a diferentes graus de pertinência e, assim, possuem um valor no intervalo $[0,1]$ referente a esse grau de associação.

Por exemplo, no domínio da ciência dos solos, quando se define em PROLOG uma regra fértil (`solo_agricola_alfa`), afirma-se que, sem dúvida, o solo agrícola denominado de *alfa* é fértil. No entanto, em PROLOG-CC, pode-se expressar a intensidade, isto é, um valor entre $[0,1]$, da afirmação e reescrever a cláusula como: fértil (`solo_agricola_alfa`, 0.8). Significa isto que o solo agrícola alfa tem uma fertilidade 0.8, considerando 0, solo agrícola não fértil e 1, solo agrícola completamente fértil, respectivamente.

6. Codificação com PROLOG-CC

O PROLOG, como na maioria das linguagens de alto nível, não tem facilidades (*built-in*) em operar com conjuntos ordinários ou difusos. Portanto, é usual executar operações com conjuntos pelo uso de objetos, tais como as listas e estruturas.

Os conjuntos difusos na fase de codificação em PROLOG-CC podem ser representados por estruturas e listas associando os elementos do conjunto ao seu respectivo grau de pertinência determinado pela função de pertinência, sendo que a função de pertinência calcula um valor no intervalo específico a que o conjunto se reporta.

A operação de *codificação* é simples. O grau de pertinência de cada valor linguístico é o valor verdade das proposições difusas.

O primeiro passo na *codificação* é dado pela entrada dos valores x , contidos no universo de discurso da respectiva variável linguística, e, assim, determina-se o grau no qual estas entradas pertencem a cada dos apropriados conjuntos difusos pela função de pertinência $\mu(x)$.

Os valores de entrada são sempre um valor numérico limitado pelo universo do raciocínio. Os limites do universo do raciocínio podem ser determinados por julgamento dos peritos. Em vários sistemas difusos usa-se uma variedade diferente de entradas. Enquanto algumas entradas podem ser medidas diretamente (peso, altura, velocidade, etc.), outras podem ser medidas unicamente por uma estimativa de um perito (Negnevitsky, 2005).

As funções de pertinência de cada predicado são definidas em PROLOG-CC através de uma lista de pares valor=> grau de pertinência, que correspondem a pontos do gráfico da função $\mu(x)$. O restante gráfico é obtido por interpolação de segmentos de recta entre pontos consecutivos da lista. A figura 2 ilustra a definição de algumas funções de pertinência na ciência dos solos, nomeadamente para o GSB (grau de saturação de bases) e PST (percentagem de sódio trocável), através de um operador # do PROLOG-CC. O interpretador

PROLOG-CC permite determinar o grau de pertinência de cada valor de GSB ou PST a cada um dos conjuntos difusos *alto*, *médio* e *baixo*. Por exemplo, de acordo com a figura 2, um GSB de 55 é médio com grau de pertinência 0.75 e é alto com grau de pertinência 0.25. O grau de pertinência ao conjunto difuso *baixo* é 0, o que equivale, na lógica clássica, a dizer que o facto *baixo*(gsb) é falso.

```

%%CONHECIMENTO PERICIAL
%% Definição dos conjuntos difusos relativos a GSB e
PST
alto(gsb) # [0=>0, 50=>0, 70=>1, 100=>1].
medio(gsb) # [0=>0, 30=>0, 50=>1, 70=>0, 100=>1].
baixo(gsb) # [0=>1, 30=>1, 50=>0, 100=>0].

alto(pst) # [0=>0, 7=>0, 10=>1, 100=>1].
medio(pst) # [0=>0, 4=>0, 7=>1, 10=>0, 100=>1].
baixo(pst) # [0=>1, 4=>1, 7=>0, 100=>0].
    
```

Figura 2 - Codificação

7. Inferência com PROLOG-CC

O processo inferencial difuso é basicamente o cálculo do grau de pertinência ($\mu(x)$) resultante das operações lógicas que compõem as regras difusas embutidas no processo de raciocínio difuso.

O valor verdade das proposições que compõem estas regras inferenciais é obtido pela combinação dos valores verdade de predicados lógicos obtidos na fase de codificação, onde os operadores ou conectivos lógicos que compõem o complexo inferencial são geralmente: NÃO (\neg), E (\wedge), OU (\vee) e IMPLICAÇÃO (\rightarrow). As equações (1), (2) e (3) definem o cálculo dos valores verdade em expressões com os conectivos NÃO, E e OU:

$$\mu(A \vee B) = \max(\mu(A), \mu(B)) \quad (1)$$

$$\mu(A \wedge B) = \min(\mu(A), \mu(B)) \quad (2)$$

$$\mu(\neg A) = 1 - \mu(A) \quad (3)$$

Num sistema pericial difuso, o que se pretende é calcular, em cada regra, o grau de pertinência da parte consequente da regra.

Por exemplo, na regra inferencial difusa *Antecedente_x \rightarrow_z Consequente_y*, em que *x*, *y* e *z* são os graus de pertinência do antecedente, consequente e da implicação, respetivamente, pretende-se calcular o valor *y*. Entretanto, é a partir do valor *z* que se calcula o grau de pertinência da consequência da regra adotada no processo de raciocínio difuso. Sendo necessário fundamentar a escolha do tipo de implicação lógica adotada.

No artigo publicado por Sukhamay Kundu & Jianhua Chen, intitulado como “Fuzzy Logic or Lukasiewicz Logic: a clarification” (Kundu, 1994), faz-se uma exposição acerca do estudo de Pavelkas que afirma que o único modo natural de formalizar a lógica difusa para valores verdade na unidade do intervalo $[0,1]$ seria usando o operador de implicação de Lukasiewicz (LK) ou qualquer forma isomórfica deste.

Na lógica difusa baseada na implicação de Lukasiewicz (LK): $A \rightarrow B$, a regra de inferência *Modus-ponens* assume a forma $\frac{\langle A, p \rangle, \langle A \rightarrow B, q \rangle}{\langle B, r \rangle}$ onde os grau de pertinência são calculados

pelas expressões (4) e (5):

$$q = \min(1, 1 - p + r) \quad (4)$$

$$r = \max(0, p + q - 1), \text{ onde } p, q \in [0, 1] \quad (5)$$

Aqui, foi inferida a fórmula B de $(A, A \rightarrow B)$ juntamente com seu valor verdade (grau de pertinência) r dos valores verdade p de A e q de $A \rightarrow B$. Nota-se que, se não se conhecem os valores verdade para A ou $A \rightarrow B$, então não é possível aplicar a regra difusa *Modus-ponens*. Isto mostra que em lógica difusa, é preciso representar fórmulas com valores verdade juntos.

É necessário portanto definir o valor de q para cada implicação da base de conhecimento. No caso do PROLOG-CC, aplicado ao domínio da ciência do solo, assume-se que todas as implicações são certas, pelo que o valor de q é sempre 1.

As regras difusas são definidas em PROLOG-CC recorrendo a um operador de implicação próprio, $:\prec$, que substitui o operador $:-$ do PROLOG convencional. A figura 3 ilustra quatro predicados, correspondentes aos valores da variável linguística de saída. A deficiência do solo no que se refere à saturação de bases trocáveis (sbt) é considerada muito forte, se o PST for alto ou se for médio, sendo o GSB baixo. As restantes regras determinam as condições em que a deficiência é considerada forte, leve ou nula.

O interpretador PROLOG-CC calcula o valor verdade de cada regra, de acordo com as expressões (1) a (5) acima. Por exemplo, na 1ª regra da figura 3, se $\text{alto}(\text{gsb})$ tiver um valor verdade de 0.25 e $\text{alto}(\text{pst})$ tiver um valor verdade de 0.5, o valor verdade do antecedente é $p = \min(0.25, 0.5) = 0.25$. Uma vez que, nesta aplicação, consideramos que a implicação é certa, isto é $q = 1$, obtemos, pela equação (5), $r = \max(0, p + q - 1) = p = 0.25$.

```
%% Regras de inferência difusa
muito_forte(def_sbt) :< alto(gsb), alto(pst).
muito_forte(def_sbt) :< medio(gsb), alto(pst).
muito_forte(def_sbt) :< baixo(gsb), alto(pst).
muito_forte(def_sbt) :< baixo(gsb), medio(pst).

forte(def_sbt) :< alto(gsb), medio(pst).
forte(def_sbt) :< medio(gsb), medio(pst).
forte(def_sbt) :< baixo(gsb), baixo(pst).

leve(def_sbt) :< medio(gsb), baixo(pst).

nulo(def_sbt) :< alto(gsb), baixo(pst).
```

Figura 3 - Inferência

É importante referir que um sistema baseado em regras difusas tem que satisfazer algumas propriedades quanto à elaboração das regras que serão descodificadas. E assim, um conjunto de regras difusas tem de ser completo no sentido em que nenhuma outra condição possível seja desprezada. Também o conjunto de regras difusas tem que ser consistente, no sentido em que nenhuma conclusão seja contraditória. E, por fim, o conjunto de regras difusas deveria ser conciso, com um mínimo de ou nenhuma redundância (Chen, 2001).

8. Descodificação em PROLOG-CC

A execução da descodificação diz respeito ao ato de combinar, comparar ou agregar conjuntos. Estas operações são de máxima importância em situações que envolvem processamento de informação ou dados. Portanto, é a etapa do sistema pericial em que as consequências das regras são convertidas num único valor de saída do sistema. É um processo de complexidade superior à do processo de codificação; existem várias escolhas a serem feitas e muitos métodos diferentes têm sido propostos (Torra, 2007). Em geral, no entanto, a descodificação resume-se em dois passos:

1. cálculo da função de pertinência agregada;
2. *defuzzification* ou determinação do valor real correspondente ao valor difuso

Recentemente, Mesiar & Komorníková (citado por (Detyniecki, 2001)) propuseram um conjunto de propriedades fundamentais que agrupam todas as definições de funções de pertinência agregadas sob condições simplificadas.

Assim, definem um operador de agregação com uma função $Agreg : \cup_{n \in N} [0,1]^n \rightarrow [0,1]$ que satisfaz:

- (a) $Agreg(x) = x$
- (b) $Agreg(0, \dots, 0) = 0$
- (c) $Agreg(1, \dots, 1) = 1$
- (d) $Agreg(x_1, \dots, x_n) \leq Agreg(y_1, \dots, y_n)$ se $(x_1, \dots, x_n) \leq (y_1, \dots, y_n)$

Estas condições são normalmente recorrentes em todas as outras definições de operadores de agregação propostas na literatura. No presente trabalho, optou-se por uma função agregada obtida pelo método de Mamdani (Mamdani&Assilia, 1975), sendo um dos métodos mais simples e eficientes, e perfeitamente adequado ao domínio de actuação do sistema pericial.

Quanto à *defuzzification*, têm sido propostos vários métodos na literatura, que determinam um valor único a partir de uma função agregada. Por exemplo, Centro de Área (CoA), Centro de Somas (CoS), Centro da Maior Área (CMA), Primeiro da Máxima (PoM), Médio da Máxima (MoM), Método da Altura (MA) e Média Ponderada Ordenada (OWA).

O operador de média ponderada ordenada (em inglês: Ordered Weighting Averaging (OWA)) foi proposto por Yager em 1988 para modelar agregação de sistemas inteligentes (Torra, 2007).

Baseado no método OWA, foi elaborado para este trabalho, um descodificador denominado como Descodificador de Peso Ordenado (*DPO*). Tendo em conta que, após a fase de inferência, poderemos obter valores verdade não nulos para dois ou mais predicados, torna-se necessário definir como combiná-los. Em ciência do solo, não é possível usar métodos baseados em médias, uma vez que um solo com deficiências de acordo com determinado critério não pode ser compensado pela não existência de deficiências de acordo com outro critério. Ou seja, um valor que aponta para deficiências fortes de acordo com uma regra tem mais peso que um valor que aponta para deficiências leves de acordo com outra regra, e deve contribuir com maior expressão no resultado final.

O *DPO* é assim definido como a soma infinitesimal de todos os pontos da função de pertinência agregada, ponderados pelo valor de deficiência correspondente. Assim, se f for a função de pertinência agregada:

$$DPO = \int_0^{100} x \cdot f(x) dx$$

Graficamente, o *DPO* é o volume de um gráfico tridimensional delimitado por $f(x)$ e por um plano inclinado a 45°.

Este método devolve um valor que representa o peso total que um atributo possa apresentar. E assim, a conclusão da análise pericial é baseada na descodificação de valores que figuram os pesos para cada regra executada.

9. Resultados e conclusões

Este artigo apresenta o modelo PROLOG-CC, como base para construção de sistemas periciais executados sobre o PROLOG convencional através de um interpretador. Desta

forma, é evitada a implementação de um método de resolução difusa, o que sugere um bom potencial de eficiência, simplicidade e flexibilidade. Este modelo tem funcionado em combinação de lógica clássica e lógica difusa no mesmo compilador. Isto representa uma grande vantagem, porque permite modelar muitos problemas usando lógica difusa. Por exemplo, regras, fatos e operadores de agregação podem ser adicionados sem esforço.

Em relação à funcionalidade, a operação pericial em PROLOG-CC apresenta rotinas simples, sem complexidades, uma vez que, em posse das regras e fatos e sua inclusão no sistema, solicita a análise pericial e obtém automaticamente os resultados.

A base de conhecimento difusa foi construída com uma estrutura semelhante a uma base de conhecimento em PROLOG convencional, o que apresenta uma vantagem em relação a outras linguagens, facilitando o entendimento e tradução do raciocínio do perito.

O PROLOG-CC, com as especificidades do sistema pericial de análise de solos, opera eficientemente quando se trata do processo de raciocínio, seja na fase de codificação, seja na fase de decodificação.

Quanto ao desempenho do sistema pericial aplicado em processos de transformação, a simulação e modelação de sistemas difusos estão relacionadas com a quantidade e precisão dos resultados que se desejam obter. O sistema pericial apresentado foi testado num conjunto de 10 amostras de solos, sendo que os resultados obtidos foram compatíveis com os de um perito humano em todos os casos (Siqueira&Rocio, 2011). Tais resultados são proporcionais às regras que são executadas e aos predicados difusos incluídos. Portanto, quanto à aplicabilidade do sistema, este não produz resultados que não foram previstos pelo elaborador das regras, aquele que traduziu em regras, o raciocínio do perito. O sistema apenas reproduz as regras e enquadra os dados inseridos na base de dados. Se o raciocínio do perito falha ou a regra não está em conformidade com o perito, o sistema falha igualmente. Portanto, este sistema pericial baseado em regras depende exclusivamente da conformação das regras e do correto processamento de codificação, inferência e decodificação.

Uma vez que a adequação do sistema pericial está diretamente relacionada com as regras, estas é que devem ser examinadas e analisadas de modo a condizerem com o raciocínio do perito.

Mostrámos, assim, que a partir do raciocínio difuso do perito é possível construir um sistema pericial para simular e modelar processos de transformação investigados na engenharia do biosistemas em PROLOG-CC. Futuramente, prevemos expandir o sistema para outras áreas da engenharia de biosistemas, generalizando o interpretador do PROLOG-CC para incluir funções de pertinência mais complexas, bem como inferências difusas que não considerem apenas implicações certas. Tencionamos ainda alargar o espectro de operadores de agregação e métodos de decodificação.

Referências

Bramer, M. (2005). *Logic Programming with Prolog*. Ed Springer.

Bratko, I. (2001). *PROLOG – Programming for Artificial Intelligence* (3th ed ed.). England: Pearson Education Limited.

- Chen, G. P. (2001). *Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems*. CRC Press.
- Coelho, H. (1995). *Inteligência Artificial em 25 lições*. Lisboa, Lisboa: Fundação Calouste Gulbenkian.
- Detyniecki, M. (2001). Fundamentals on Aggregation Operators. *Berkeley initiative in Soft Computing* . <http://www.cs.berkeley.edu/~marcin/agop.pdf>., United States of America.
- Guilherme, I. (1990). *Sistema de Manipulação de Conjuntos e Relações Nebulosas e Programação em Lógica Nebulosa –Prolog Nebuloso*. Tese de Mestrado da Faculdade de Engenharia Elétrica - Universidade Estadual de Campinas, Campinas- São paulo.
- Henderson, H. (2007). *Artificial Intelligence – Mirrors for the Mind*. Chelsea House Publishers.
- Ishizuka, M. K. (December de 1985). Prolog-Elf Incorporating Fuzzy Logic Pages . (L. Publisher Ohmsha, Ed.) *Journal New Generation Computing* , 3, Number 4, pp. 479-486.
- Klawonn, F. &. (1994). A Lukasiewicz Logic Based Prolog. *Mathware & Soft Computing* , 1, pp. 5-29.
- Kundu, S. &. (1994). *Fuzzy Logic or Lukasiewicz Logic: A Clarification* (Vols. October 16 - 19,1994). (M. Z. Zbigniew W. & Ras, Ed.) Charlotte, North Carolina, USA: Springer.
- Lee, R. (1972). Fuzzy Logic and The Resolution Principle . *J.Assoc.Comput.Mech* , 19, pp. 109-119.
- Li, D. &. (1990). *A Fuzzy Prolog Database System*. New York: John Wiley & Sons.
- Mamdani, E.H. and S. Assilian (1975), "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-Machine Studies*, Vol. 7, No. 1, pp. 1-13, 1975.
- Negnevitsky, M. (2005). *Artificial Intelligence – A guide to Intelligent Systems* (2nd Edition ed.). Pearson Education Limited.
- Siqueira, J., Rocio, V. (2011). “Sistema Pericial em Prolog no diagnóstico do potencial agrícola de solos”. Atas do VI Congresso Ibérico de Agro-Engenharia, Évora, Portugal, 5 a 7 de Setembro de 2011 (aceite para publicação).
- Torra, V. &. (2007). *Modeling Decisions-Information Fusion and Aggregation Operators*. Springer-Verlag Berlin Heidelberg.
- Zadeh, L. (1965). Fuzzy Sets. *Information and Control* , 8, pp. 338-353.