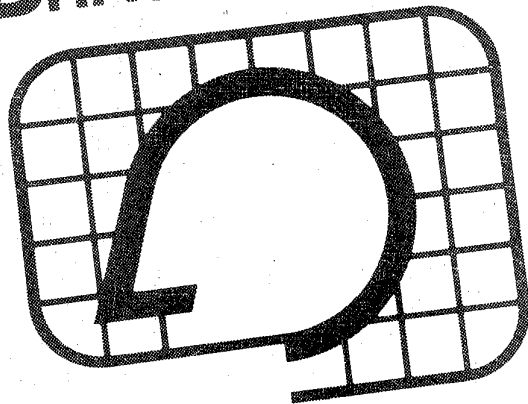


**2º SIMPÓSIO  
BRASILEIRO DE**



**INTELIGÊNCIA  
ARTIFICIAL**

006.306  
S612

---

**ANAIS**

**2º SIMPÓSIO BRASILEIRO DE INTELIGÊNCIA ARTIFICIAL**

**20-21-22 DE NOVEMBRO DE 1985**

**INPE - INSTITUTO DE PESQUISAS ESPACIAIS**

**SÃO JOSÉ DOS CAMPOS - SP**

SAEQ  
UM AMBIENTE PARA DESENVOLVIMENTO DE PROTÓTIPOS DE  
SISTEMAS ESPECIALISTAS BASEADO EM LÓGICA

Autores: Newton José Vieira e Roberto Lins de Carvalho

Dentre as áreas de estudo da Inteligência Artificial, a dos Sistemas Especialistas tem se destacado como uma das que mais tem produzido resultados práticos. Já existem sistemas especialistas para os mais diversos campos de especialidade, como diagnóstico médico, prospecção de minerais, configuração de sistemas de computação, elucidação de estruturas químicas, análise de circuitos elétricos, etc., alguns com desempenho comparável, ou mesmo melhor, que o de especialistas humanos (1).

A tarefa de construir um sistema especialista é usualmente muito complexa, podendo levar vários anos para ser executada. Isso não é de estranhar, tendo em vista que o conhecimento especializado é adquirido pelo especialista humano em muitos anos de estudo, experimentação e prática. O isolamento e transferência de tal conhecimento para um programa é dificultado pelo fato de que quem detém o conhecimento e quem concebe o programa são, normalmente, pessoas distintas, leigos na especialidade um do outro.

Dentre os fatores fundamentais para o eventual sucesso da tarefa, está o envolvimento de especialistas, não apenas para suprir o conhecimento, mas também para criticar e avaliar o desempenho do sistema durante o desenvolvimento. Para manter aceso esse envolvimento, é de particular conveniência a obtenção de resultados "visíveis" em prazo curto. É conveniente obter um ambiente onde os especialistas observem uma evolução contínua (e estimulante) do sistema, como resultado de seu envolvimento.

A obtenção de resultados visíveis em prazo curto pode ser conseguida através da construção de um protótipo, já nos primeiros estágios do desenvolvimento. Já existem vários sistemas que podem ser utilizados para construção de protótipos, cada um baseado num paradigma de representação (ou mais de um), mais adequado para certos tipos de problemas do que para outros. O objetivo desse trabalho, é desenvolver um ambiente baseado em lógica, reunindo em um todo coerente, idéias de áreas como Prova Automática de Teoremas (2), Programação Lógica (3), "Question Answering Systems" (4) e de alguns sistemas especialistas já desenvolvidos, principalmente daqueles "baseados em regras" (5). Algumas dessas idéias já foram experimentadas em um primeiro protótipo do sistema SAFO (6).

Vários mecanismos foram imaginados para compor o

ambiente a ser proposto. Alguns deles ainda não estão suficientemente "ajustados", e alguns outros requerem um estudo mais elaborado para a sua introdução. Para dar uma idéia do sistema, mencionamos brevemente 5 mecanismos.

### 1. Definições

A entrada de fatos sobre o domínio da aplicação é feita, basicamente, através de fórmulas da lógica de primeira ordem, em uma forma estilizada. Definições explícitas de objetos, funções e relações podem ser feitas.

O conceito de definição é muito importante no sistema, sob o ponto de vista do processo dedutivo: ele é utilizado para a hierarquização do "raciocínio". O sistema procura manter uma dedução no mais alto nível possível, só recorrendo ao uso de uma definição, quando isso se fizer realmente necessário. E o uso de uma definição sempre é feito no sentido de substituir o termo definido pela sua definição.

Essa estratégia hierárquica tem duas vantagens: (a) é eficiente, evitando o uso de definições quando desnecessário, e (b) propicia um ambiente adequado para o fornecimento de explicações mais claras do processo dedutivo.

### 2. Consultas

Existem, basicamente, dois tipos de consultas que o usuário pode fazer a uma base de conhecimentos: (a) simplificações de formas funcionais, e (b) perguntas.

Solicitando a simplificação de uma forma funcional, o usuário pode obter o valor de uma função aplicada a argumentos específicos.

Quanto às perguntas, essas podem ser de vários tipos: (b.1) perguntas simples, que o sistema responde sim ou não; (b.2) perguntas com extração de respostas, onde o sistema fornece uma ou mais tuplas de objetos que satisfaçam a certa condição; (b.3) perguntas com interação, onde o sistema, na ausência de informações que se apliquem a uma dada meta, consulta o usuário; e (b.4) perguntas com diálogo, onde o usuário pode interromper um processo dedutivo para fazer outra consulta, entrar com novas informações, etc., com a opção de fazer o sistema voltar ao processo dedutivo interrompido.

### 3. Associação de Diretivas

Algumas diretivas podem ser associadas a predicados de tal forma que, quando o sistema seleciona uma meta que contenha o predicado, durante uma dedução, a diretiva associada é levada a efeito.

Por exemplo, predicados que dizem respeito a dados de laboratório, os quais devem ser solicitados do usuário sempre que necessários, podem ser marcados através de uma associação com a diretiva lab. Outros predicados, que dizem respeito a opções de múltipla escolha, onde o usuário deverá escolher a que se aplique ao caso corrente, podem ser marcados com a diretiva escolha. Se o usuário quiser fazer um rastreamento seletivo de um processo dedutivo, ele pode associar a predicados específicos, que eventualmente sejam atingidos em uma dedução, a diretiva rastreio.

Essas, e quaisquer outras diretivas, podem ser desassociadas a qualquer momento que o usuário deseje.

#### 4. Disparos

Disparos são ações imperativas que podem ser executadas pelo sistema, sob requisição do usuário. Existem 3 tipos básicos de disparos: (a) simples, (b) condicional e (c) repetitivo.

Os disparos simples podem ser embutidos ("built-in") ou definidos. Em particular, existem disparos embutidos para definições de objetos, de funções, de relações, e de disparos; existem disparos para consultas, associação e desassociação de diretivas, manutenção da base de conhecimentos, etc. Um disparo pode ser definido como uma sequência de um ou mais disparos simples, condicionais e/ou repetitivos.

Um disparo condicional é um conjunto de sequências de disparos, cada sequência prefixada por uma condição; qualquer sequência que estiver prefixada por uma condição que se verificar é candidata a ser executada; apenas uma é escolhida.

Existem 2 tipos de disparos repetitivos: um que especifica a repetição de um disparo condicional até que todas as condições falhem; e outro que especifica uma iteração sobre uma sequência de disparos, para cada tupla de objetos que satisfaça a uma condição ou que seja membro de uma lista.

#### 5. Controle da Estratégia de Controle

O sistema possui uma estratégia padrão para controle do processo dedutivo. Essa estratégia é baseada na redução de problemas a sub-problemas, com alguns mecanismos especiais para escolha de sub-problemas, de forma a incrementar a eficiência; isso temperado com o conceito de raciocínio hierarquizado mencionado no item 1.

Entretanto, uma maior eficiência pode ainda ser conseguida com heurísticas dependentes do domínio de

aplicação. Além disso, a sequência de perguntas do sistema para o usuário deve ser o mais "natural" possível (para o usuário); e parte dessa naturalidade poderá ser conseguida através de ajustes de estratégia para a aplicação específica.

Assim, existirá uma forma do usuário modificar o comportamento do procedimento estratégico, para propiciar um sistema que se comporte mais eficiente e/ou naturalmente nas suas deduções e/ou diálogos com o usuário.

## 6. Referências

- <1> Hayes-Roth, F., Waterman, D.A., Lenat, D.B. (eds), Building Expert Systems, Addison-Wesley, 1983.
- <2> Loveland, D.W., Automated Theorem Proving: A Logical Basis, North-Holland, 1978.
- <3> Kowalski, R., Logic for Problem Solving, North-Holland, 1979.
- <4> Green, C.C., The Application of Theorem Proving to Question-Answering Systems, Stanford AI Project MEMO AI-96, 1969 (Ph.D. Thesis).
- <5> Buchanan, B.G., Shortliffe, E.H. (eds), Rule Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project, Addison-Wesley, 1983.
- <6> Vieira, N.J., SAEQ - Manual do Usuário, PUC/RJ - Departamento de Informática, 1984 (circulação interna).