

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS

Engenharia de Computação

SEMINÁRIO DE I.A.

“SISTEMAS ESPECIALISTAS”

César Henrique Kallas RA: 02099224

Daniel Figueiredo Machado RA:02508224

Turma: Turma 2

CAMPINAS 2005

Introdução

Sistema - "Conjunto de elementos, materiais ou idéiais, entre os quais se possa encontrar ou difinir alguma relação".

Especialista - "Pessoa que se consagra com particular interesse e cuidado a certo estudo. Conhecedor, perito".

Sistemas Especialistas são sistemas que solucionam problemas que são resolvíveis apenas por pessoas especialistas (que acumularam conhecimento exigido) na resolução destes problemas.

Um Sistema de Inteligência Artificial criado para resolver problemas em um determinado domínio (área de interesse específico para as quais podemos desenhar um sistema de IA) cujo conhecimento utilizado é fornecido por pessoas que são especialistas naquele domínio, é denominado Sistema Especialista.

Esses sistemas são programas de computador que tentam resolver problemas que os seres humanos resolveriam "emulando" o raciocínio de um especialista, aplicando conhecimentos específicos e inferências são ditos Sistemas Especialistas.

Um Sistema Convencional é baseado em um algoritmo, emite um resultado final correto e processa um volume de dados de maneira repetitiva enquanto que um Sistema Especialista é baseado em uma busca heurística e trabalha com problemas para os quais não existe uma solução convencional organizada de forma algoritmica disponível ou é muito demorada.

De um modo geral, sempre que um problema não pode ser algoritmizado, ou sua solução conduza a um processamento muito demorado, os Sistemas Especialistas podem ser uma saída, pois possuem o seu mecanismo apoiado em processos heurísticos.

Preservar e transmitir o conhecimento de um especialista humano em uma determinada área.

Um Sistema Especialista não é influenciado por elementos externos a ele, como ocorre com o especialista humano, para as mesmas condições deverá fornecer sempre o mesmo conjunto de decisões.

Sistemas Especialistas

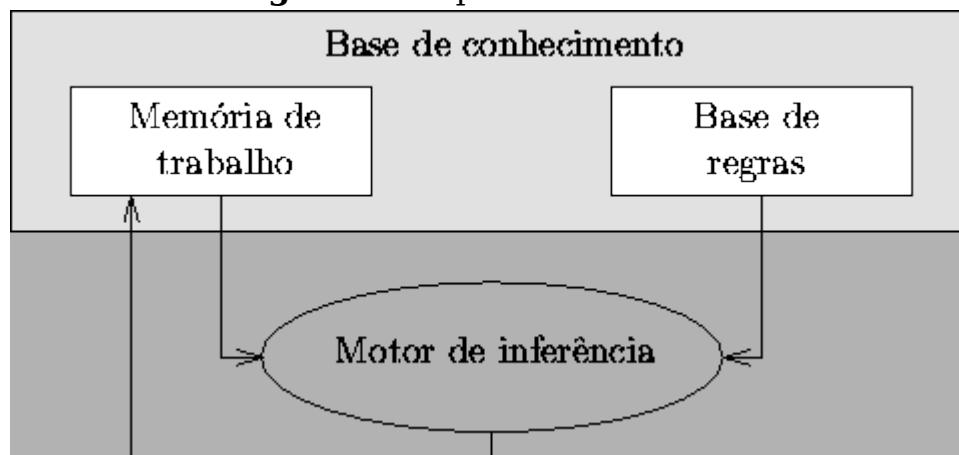
Os sistemas especialistas, SE, em geral, podem ser divididos em três partes: uma base de regras, uma memória de trabalho e um motor de inferência. A base de regras e a memória de trabalho são a chamada base de conhecimento.

Os SEs restringem-se somente a alguns domínios específicos do conhecimento, tentando reproduzir boa parte do conhecimento de um especialista em determinado assunto. Ainda é inviável pensar na implementação de sistemas que respondam e reajam sobre temas gerais utilizando bases de conhecimento, sendo que para a representação desses, uma base extremamente extensa deve ser construída.

Os SE's são concebidos para reproduzir o comportamento de especialistas humanos na resolução de problemas do mundo real, mas o domínio destes problemas é altamente restrito. Os primeiros SE's que obtiveram sucesso em seu objetivo foram o sistema DENDRAL e MYCIN. O sistema DENDRAL é capaz de inferir a estrutura molecular de compostos desconhecidos a partir de dados espectrais de massa e de resposta magnética nuclear. O sistema MYCIN auxilia médicos na escolha de uma terapia de antibióticos para pacientes com bacteremia, meningite e cistite infecciosa, em ambiente hospitalar.

Desde então, muitos SE's foram desenvolvidos para resolver problemas em muitos domínios diferentes, incluindo: agricultura, química, sistemas de computadores, eletrônica, engenharia, geologia, gerenciamento de informações, direito, matemática, medicina, aplicações militares, física, controle de processos e tecnologia espacial.

Figure 3: Arquitetura de um SE



Um SE atual, conforme mostrado na figura 3, apresenta, em geral, uma arquitetura com três módulos: uma base de regras, uma memória de trabalho e um motor de inferência. A **base de regras** e **memória de trabalho** formam a **chamada base de conhecimento do SE**, onde está representado o conhecimento sobre o domínio. O **motor de inferência é o mecanismo de controle do sistema** que avalia e aplica as regras de acordo com as informações da memória de trabalho.

A **memória de trabalho** é apenas uma seqüência de caracteres em um

modelo chamado de Post, em um modelo generalizado pode conter qualquer tipo de estrutura de dados. Mais do que estruturas de dados, as memórias de trabalho de SE's devem respeitar um método de representação de conhecimento, isto é, uma linguagem formal e uma descrição matemática de seu significado. A lógica de primeira ordem é um exemplo típico de formalismo de representação de conhecimento.

A **base de regras** passa a conter condições que representam “perguntas” à representação de conhecimento da memória de trabalho. Estas perguntas, limitadas à comparação de caracteres no modelo de Post, podem ser de diferentes tipos, mas em geral envolvem variáveis a serem instanciadas e eventualmente algum tipo de inferência. A sintaxe das regras varia de acordo com o sistema e pode ser bastante flexível e próxima da linguagem natural.

Outra característica comum nos SE's atuais é a existência de um mecanismo de raciocínio incerto que permita representar a incerteza a respeito do conhecimento do domínio.

O **motor de inferência** controla a atividade do sistema. Esta atividade ocorre em ciclos, cada ciclo consistindo em três fases:

1. Correspondência de dados: onde as regras que satisfazem a descrição da situação atual são selecionadas.
2. Resolução de conflitos: onde as regras que serão realmente executadas são escolhidas dentre as regras que foram selecionadas na primeira fase, e ordenadas.
3. Ação: a execução propriamente dita das regras.

A chave para o desempenho de um SE está no conhecimento armazenado em suas regras e em sua memória de trabalho. Este conhecimento deve ser obtido junto a um especialista humano do domínio e representado de acordo com regras formais definidas para a codificação de regras no SE em questão. Isto **divide um SE em duas partes**: a **ferramenta de programação** que define o formato do conhecimento da memória de trabalho e das regras, além dos aspectos operacionais de sua utilização, e o **conhecimento do domínio** propriamente dito.

Devido a esta separação, atualmente, os SE's são desenvolvidos em geral a partir de **arcabouços de sistemas especialistas (ASE)**: ferramentas que suportam todas as funcionalidades de um SE, restando ao programador apenas codificar o conhecimento especializado de acordo com a linguagem de representação de conhecimento disponível. A existência de ASE's facilitou bastante a implementação de SE's e foi um dos fatores responsáveis por sua disseminação.

AQUISIÇÃO DE CONHECIMENTO

A parte mais sensível no desenvolvimento de um SE é, certamente, a aquisição de conhecimento. Esta não pode limitar-se à adição de novos elementos de conhecimento à base de conhecimentos. É necessário integrar o novo conhecimento ao conhecimento já disponível, através da definição de relações entre os elementos que constituem o novo conhecimento e os elementos já armazenados na base. Dois tipos de mecanismos para a definição

de tais relações foram propostos: ligar os elementos de conhecimento diretamente através de ponteiros, ou reunir diversos elementos relacionados em grupos (em inglês “clustering”).

Outro ponto importante na aquisição de conhecimento é o tratamento de incoerências. Dependendo da forma como o novo conhecimento é adquirido, pode haver erros de aquisição. Estes erros podem resultar da própria natureza do conhecimento, como em dados obtidos através de sensores sujeitos a ruído, ou podem ser gerados pela interface humana existente entre o mundo real e o sistema de representação. Técnicas foram desenvolvidas para evitar erros de aquisição, como, por exemplo, a especificação de regras de aquisição em que o tipo de conhecimento esperado é definido. Estas técnicas são comuns aos sistemas de representação de conhecimento e aos sistemas de gerenciamento de bancos de dados. Por outro lado, uma base de conhecimento pode ser examinada periodicamente com a finalidade de detectar incoerências eventualmente introduzidas no processo de aquisição. Este método é limitado pelo fato de que linguagens de representação razoavelmente expressivas não contam com procedimentos completos de verificação conhecidos. Finalmente, deve-se observar que a adequação do formalismo de representação ao tipo de conhecimento do mundo real a ser representado é fundamental para a eficiência do processo de aquisição.

MÉTODOS DE REPRESENTAÇÃO DE CONHECIMENTO

A parte mais importante no projeto de um SE é a escolha do método de representação de conhecimento. A linguagem associada ao método escolhido deve ser suficientemente expressiva (mas não mais do que o suficiente) para permitir a representação do conhecimento a respeito do domínio escolhido de maneira completa e eficiente. Em tese, uma representação geral como a lógica seria suficientemente expressiva para representar qualquer tipo de conhecimento. No entanto, problemas de eficiência, facilidade de uso e a necessidade de expressar conhecimento incerto e incompleto levaram ao desenvolvimento de diversos tipos de formalismos de representação de conhecimento. A seguir, apresentam-se alguns dos formalismos de representação de conhecimento mais utilizados.

- Lógica

A lógica é a base para a maioria dos formalismos de representação de conhecimento, seja de forma explícita, como nos SE's baseados na linguagem Prolog, seja disfarçada na forma de representações específicas que podem facilmente ser interpretadas como proposições ou predicados lógicos, por exemplo, as listas da forma:

(<atributo>, <objeto>, <valor>, <coeficiente de certeza>)

utilizadas como padrão para a representação de conhecimento no sistema MYCIN. Mesmo os formalismos não-lógicos têm, em geral, seu significado formal descrito através de uma especificação lógica de seu comportamento.

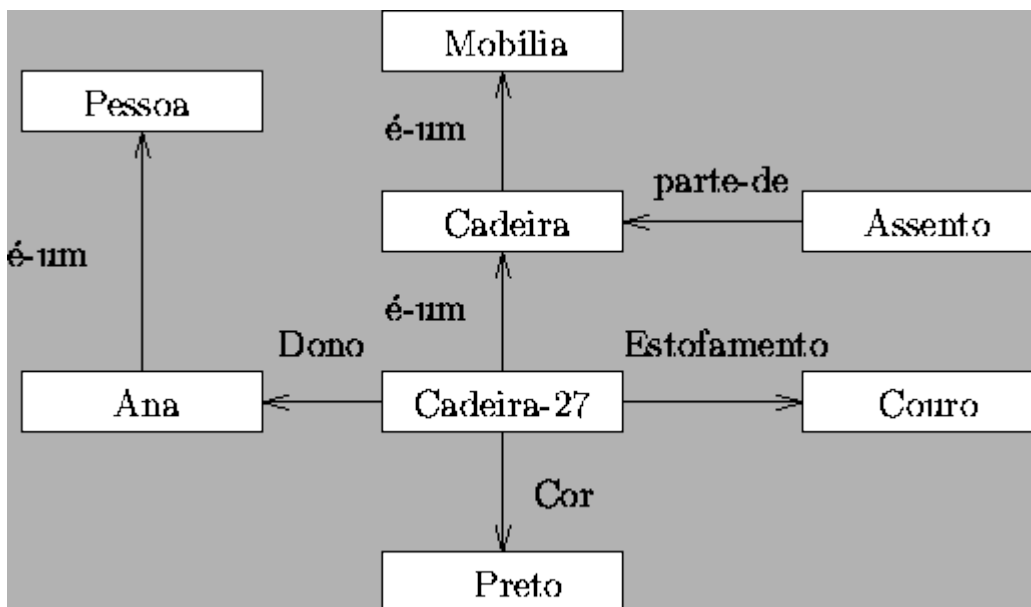
Exemplo: O sistema FASE dispõe de um módulo de representação de conhecimento baseado na lógica de primeira ordem. Todos os formalismos implementados no sistema apresentam as mesmas primitivas de acesso: store, para armazenar um fragmento de conhecimento na base, query, para obter as instâncias que satisfazem às restrições especificadas através das variáveis que ocorrem em seu argumento, e list, que permite examinar o conteúdo da base.

O acesso a uma base de conhecimento pode ser feito através de um padrão de uma regra ou, diretamente, através de comandos.

- Redes semânticas

Rede semântica é um nome utilizado para definir um conjunto heterogêneo de sistemas. Em última análise, a única característica comum a todos estes sistemas é a notação utilizada. Uma rede semântica consiste em um conjunto de nodos conectados por um conjunto de arcos. Os nodos em geral representam objetos e os arcos, relações binárias entre esses objetos. Mas os nodos podem também ser utilizados para representar predicados, classes, palavras de uma linguagem, entre outras possíveis interpretações, dependendo do sistema de redes semânticas em questão.

A utilização do formalismo de nodos e arcos para a representação de conhecimento foi proposta por Quillian. No seu artigo, Quillian propõe um modelo computacional da memória humana chamado memória semântica. Este modelo, onde conceitos são representados por nodos, e relações entre conceitos, por arcos, explica diversos resultados experimentais sobre o comportamento da memória humana, como, por exemplo, o fato de que o reconhecimento de objetos que pertencem a classes mais numerosas toma mais tempo do que o reconhecimento dos pertencentes a classes menos numerosas. Muitas características dos sistemas de redes semânticas desenvolvidos já estavam presentes na proposta de Quillian.



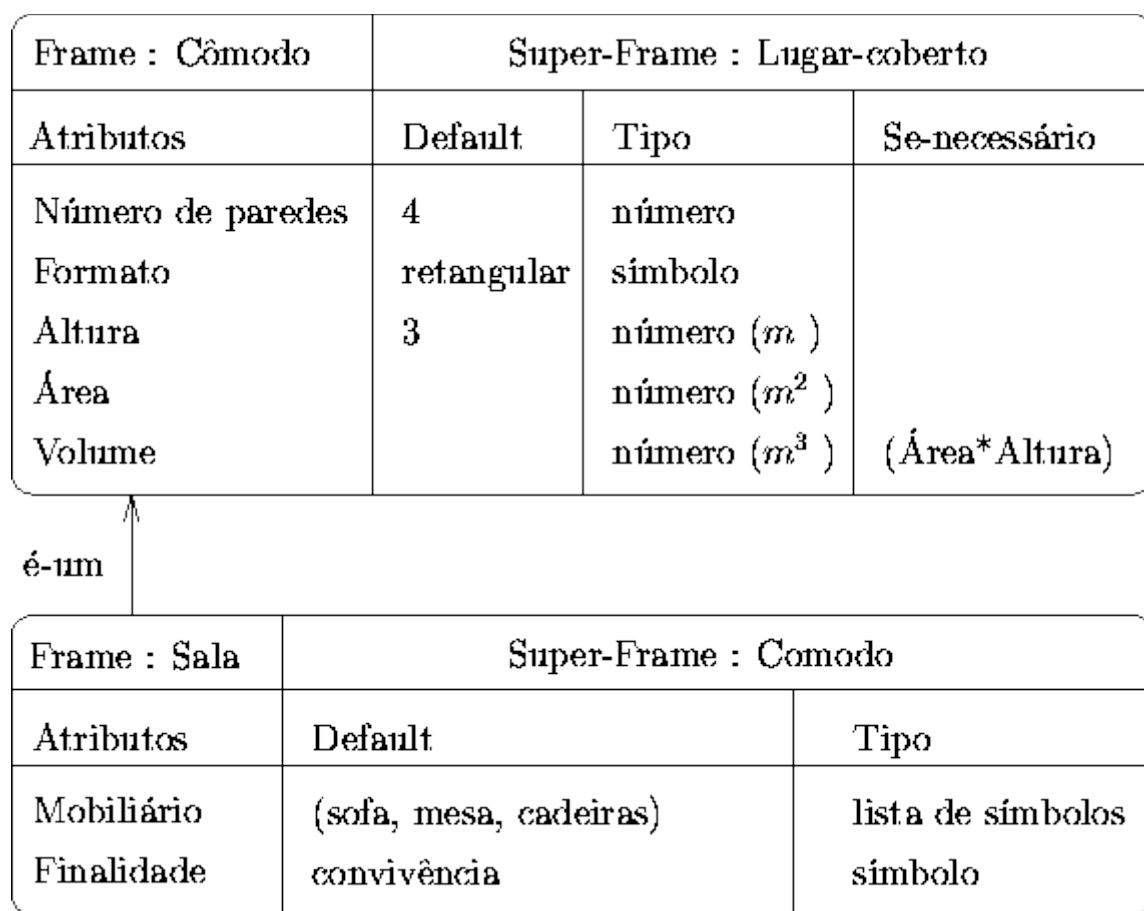
Exemplo: Considere a rede semântica da figura acima, que representa conceitos relacionados com mobiliário. Os arcos é-um e é-parte são bastante comuns em sistemas de redes semânticas. Este tipo de arco é utilizado para determinar a herança de propriedades. Os demais arcos (dono, cor, estofamento) são específicos do domínio e representam propriedades de conceitos. Esses arcos são chamados traços.

- Quadros

Os quadros (do inglês, "frames"), e sua variação, os roteiros (do inglês, "scripts"), foram introduzidos para permitir a expressão das estruturas internas dos objetos, mantendo a possibilidade de representar herança de propriedades como as redes semânticas.

As idéias fundamentais destes métodos foram introduzidas por Marvin Minsky em seu artigo *A framework to represent knowledge*. As aplicações propostas por Minsky para o novo método foram análise de cenas, modelagem da percepção visual e compreensão de linguagem natural. No entanto, o artigo não propõe metodologia de implementação, nem definição formal do método. Desde 1975, diversos sistemas foram implementados baseados na idéia de quadros, e diversas definições formais foram propostas. O método de quadros também está na origem das idéias que levaram às linguagens de programação orientadas a objetos. Os roteiros foram propostos por Schank e Abelson, e consistem em sistemas de quadros especializados na descrição de seqüências de eventos.

Figure 6: Quadros



Em geral, um quadro consiste em um conjunto de atributos que, através de seus valores, descrevem as características do objeto representado pelo quadro. Os valores atribuídos a estes atributos podem ser outros quadros, criando uma rede de dependências entre os quadros. Os quadros são também organizados em uma hierarquia de especialização, criando uma outra dimensão de dependência entre eles. Os atributos também apresentam propriedades, que dizem respeito ao tipo de valores e às restrições de número que podem ser associados a cada atributo. Essas propriedades são chamadas facetas.

Exemplo: Quadros descrevendo um comodo e uma sala são mostrados na figura 6. Por esta descrição, pode-se concluir que uma sala é um tipo de cômodo, normalmente com quatro paredes e de formato retangular, com um

mobiliário específico. As facetas dos atributos especificam os tipos de valores esperados e, se for o caso, procedimentos adequados para calcular o valor do atributo.

FERRAMENTAS PARA CONSTRUÇÃO DE SE's

As dimensões escolhidas foram as seguintes: interface com o usuário, interface de desenvolvimento, interface com o sistema operacional, motor de inferência e métodos de representação de conhecimento. Os aspectos comentados abaixo foram considerados na pesquisa como importantes. Para cada dimensão da análise são ressaltados os aspectos considerados críticos.

- Interface com o usuário

Em qualquer tipo de "software", a interface com o usuário final é fundamental para o seu sucesso. Em particular, um SE, além de apresentar uma interface ergonomicamente bem projetada, deve ainda levar em conta o grau de familiarização do usuário com o domínio de trabalho do sistema e com os computadores em geral.

Algumas técnicas tornam a interface com o usuário mais amigável, por exemplo, o uso de janelas, menus, gráficos de alta resolução, animação, cores, etc. De todo modo, as telas a serem apresentadas ao usuário devem ser de fácil compreensão, e as explicações necessárias devem ser claras e diretas. Mais algumas características interessantes para interfaces com o usuário são: (i) disponibilidade de diversos tipos de interfaces, adaptadas ao tipo de usuário (iniciante, especialista); (ii) possibilidade de interromper a execução do sistema em um determinado ponto e poder retomá-la sem necessidade de reprocessamento; (iii) mensagens de erro claras e informativas; (iv) possibilidade de alterar certas entradas ao sistema e comparar os resultados obtidos; (v) capacidade de capturar e armazenar telas de execução. Os aspectos considerados críticos foram: facilidade para explicação e documentação.

- Interface de desenvolvimento: A interface a ser utilizada pela equipe de desenvolvimento de um SE representa, em geral, a maior parte do código de um ASE, além de ser determinante para o bom andamento dos projetos desenvolvidos com o arcabouço em questão.

A possibilidade de customizar o ambiente pela introdução de novas capacidades, ou pela modificação de capacidades já existentes, através da escrita de código na linguagem de implementação do ASE, é uma característica bastante útil. Caso os programas fonte do arcabouço estejam disponíveis, isto é facilitado.

Correção de erros, manutenção de uma lista de comandos da seção, índice cruzado de símbolos e regras, visualização gráfica dos encadeamentos de regras utilizadas na solução, editor de regras, facilidades de explicação e uma boa documentação são outras características positivas para a interface de desenvolvimento.

Aspectos considerados críticos: facilidade para explicação, documentação, facilidade para customizar explicações e prototipagem rápida.

- Interface com o sistema operacional: Existem atualmente SE's rodando em plataformas que variam de microcomputadores do tipo PC até "mainframes", passando por estações de trabalho e máquinas especializadas em processamento simbólico, as chamadas "Lisp Machines". No entanto, setenta por cento dos SE's pesquisados por Stylianou et al foram desenvolvidos para microcomputadores. Diversas ferramentas disponíveis no mercado rodam ainda em várias plataformas diferentes.

Quanto à linguagem de programação, existem ASE's escritos em linguagem de montagem (do inglês, "assembler"), C, Fortran, Basic, Forth, Pascal e Cobol, além, é claro, daqueles escritos em linguagens para IA como Lisp, Prolog e Smalltalk. Diversos ASE's originalmente escritos em Lisp foram traduzidos para linguagens como C ou C++ para aumentar sua eficiência, portabilidade e compatibilidade com outros "softwares".

A facilidade de comunicação com aplicações convencionais (como bancos de dados, planilhas e sistemas de rede) é considerada uma importante característica em um ASE. O fato dessa facilidade ter sido relegada a um segundo plano nos primeiros ASE's representou uma grande dificuldade para sua disseminação em ambiente comercial.

Uma última característica ligada ao sistema operacional é o grau de segurança oferecido por um ASE. É possível que um SE envolva informações confidenciais ou únicas que não devem ser acessadas e modificadas por qualquer usuário.

Aspectos considerados críticos: facilidade para integração com sistemas já existentes.

- Motor de inferência: As principais características do motor de inferência disponível em um ASE dizem respeito às seguintes funcionalidades: método de raciocínio, estratégia de busca, resolução de conflito e representação de incerteza. Dentre todos os aspectos discutidos nesta seção, foi considerada crítica a existência de raciocínio do tipo encadeamento regressivo.

Modo de raciocínio:

Existem basicamente dois modos de raciocínio aplicáveis a regras de produção: encadeamento progressivo ou encadeamento a frente (do inglês, "forward chaining"), e encadeamento regressivo ou encadeamento para trás (do inglês, "backward chaining"). No encadeamento progressivo, também chamado encadeamento dirigido por dados, a parte esquerda da regra é comparada com a descrição da situação atual, contida na memória de trabalho. As regras que satisfazem a esta descrição têm sua parte direita executada, o que, em geral, significa a introdução de novos fatos na memória de trabalho.

No encadeamento regressivo, também chamado encadeamento dirigido por objetivos, o comportamento do sistema é controlado por uma lista de objetivos. Um objetivo pode ser satisfeito diretamente por um elemento da memória de trabalho, ou podem existir regras que permitam inferir algum dos objetivos correntes, isto é, que contenham uma descrição deste objetivo em suas partes direitas. As regras que satisfazem esta condição têm as instâncias correspondentes às suas partes esquerdas adicionadas à lista de objetivos

correntes. Caso uma dessas regras tenha todas as suas condições satisfeitas diretamente pela memória de trabalho, o objetivo em sua parte direita é também adicionado à memória de trabalho. Um objetivo que não possa ser satisfeito diretamente pela memória de trabalho, nem inferido através de uma regra, é abandonado. Quando o objetivo inicial é satisfeito, ou não há mais objetivos, o processamento termina.

O tipo de encadeamento normalmente é definido de acordo com o tipo de problema a ser resolvido. Problemas de planejamento, projeto e classificação tipicamente utilizam encadeamento progressivo, enquanto problemas de diagnóstico, onde existem apenas algumas saídas possíveis mas um grande número de estados iniciais, utilizam encadeamento regressivo.

Em geral, os ASE's adotam apenas um modo de raciocínio, no entanto existem alguns que permitem ambos os modos, mas de maneira independente, e ainda outros que permitem um encadeamento misto, onde os encadeamentos progressivo e regressivo se alternam de acordo com o desenvolvimento da solução do problema e com a disponibilidade de dados.

Uma característica importante do modo de raciocínio se refere à monotonicidade ou não do método de inferência. Sistemas monotônicos não permitem a revisão de fatos, isto é, uma vez um fato declarado verdadeiro, ele não pode mais tornar-se falso. Sistemas não monotônicos, por outro lado, permitem a alteração dinâmica dos fatos. O preço desta capacidade é a necessidade de um mecanismo de revisão de crenças, pois uma vez que um fato, antes verdadeiro, torna-se falso, todas as conclusões baseadas neste fato também devem tornar-se falsas.

Estratégia de busca:

Uma vez definido o tipo de encadeamento, o motor de inferência necessita ainda de uma estratégia de busca para guiar a pesquisa na memória de trabalho e na base de regras. Este tipo de problema é conhecido como busca em espaço de estados. Este tópico foi um dos primeiros estudados em IA, no contexto de solução de problemas (do tipo quebra-cabeças) e jogos por computador (damas, xadrez, go, etc.).

Resolução de conflito:

Ao terminar o processo de busca, o motor de inferência dispõe de um conjunto de regras que satisfazem à situação atual do problema, o chamado conjunto de conflito. Se esse conjunto for vazio, a execução é terminada, caso contrário, é necessário escolher que regras serão realmente executadas e em que ordem. Os métodos de resolução de conflito mais utilizados ordenam as regras de acordo com os seguintes critérios: prioridades atribuídas estaticamente, características da estrutura das regras como complexidade, simplicidade e especificidade, características dos dados associados às regras como o tempo decorrido desde sua obtenção, sua confiabilidade ou seu grau de importância; e, finalmente, seleção ao acaso.

Em geral, a utilização de um desses critérios é insuficiente para resolver os conflitos. Neste caso, o ASE pode combinar mais de um método na forma de método primário, secundário, etc. Os melhores ASE's dispõem de diversos métodos de resolução de conflito e permitem ao usuário a especificação de quais métodos utilizar e em que ordem.

Representação de incerteza:

O tratamento de incerteza é uma ativa área de pesquisa em SE's, pois os domínios adequados à implementação de SE's se caracterizam exatamente por não serem modelados por nenhuma teoria geral, o que implica descrições incompletas, inexatas ou incertas. Diversos métodos foram propostos para tratar este problema, por exemplo, método Bayesiano, fatores de certeza (conforme o modelo adotado no MYCIN), teoria de Dempster-Shafer, teoria dos conjuntos nebulosos, teoria de probabilidades subjetivas e teoria de possibilidades.

De maneira geral, estes métodos atribuem aos fatos e regras uma medida numérica que represente de alguma forma a "confiança" do especialista. Os métodos utilizados não são necessariamente coerentes uns com os outros e cada método adapta-se melhor a determinados tipos de problemas. Diversos ASE's dispõem de mais de um método de tratamento de incerteza, deixando ao usuário a escolha do mais adequado ao seu problema. Uma característica freqüente desses métodos é a existência de um limite mínimo para a medida de incerteza, abaixo do qual o fato ou regra é desconsiderado. Este limite pode, em geral, ser fixado pelo usuário.

Representação de conhecimento:

Em geral, os ASE's se limitam a oferecer um único tipo de representação de conhecimento. Alguns sistemas dispõem de mais de um formalismo, os quais, no entanto, devem ser utilizados de maneira isolada. Alguns poucos ASE's possuem os chamados sistemas híbridos de representação de conhecimento que, além de possuir diversos formalismos de representação, dispõem também de algoritmos de acesso que integram os conhecimentos representados nos diversos formalismos para permitir sua utilização de maneira integrada.

BIBLIOGRAFIA:

Universidade Federal de Santa Catarina. DAS - Inteligência Artificial. Florianópolis, 2005. Disponível em: <<http://www.das.ufsc.br/gia/softcomp/softcomp.html>>. Acesso em: 10 maio 2005.

Universidade Estadual de Maringá. Departamento de Informática - Inteligência Artificial. Maringá, 2005. Disponível em: <<http://www.din.uem.br/~ia/especialistas/>>. Acesso em: 11 maio 2005.