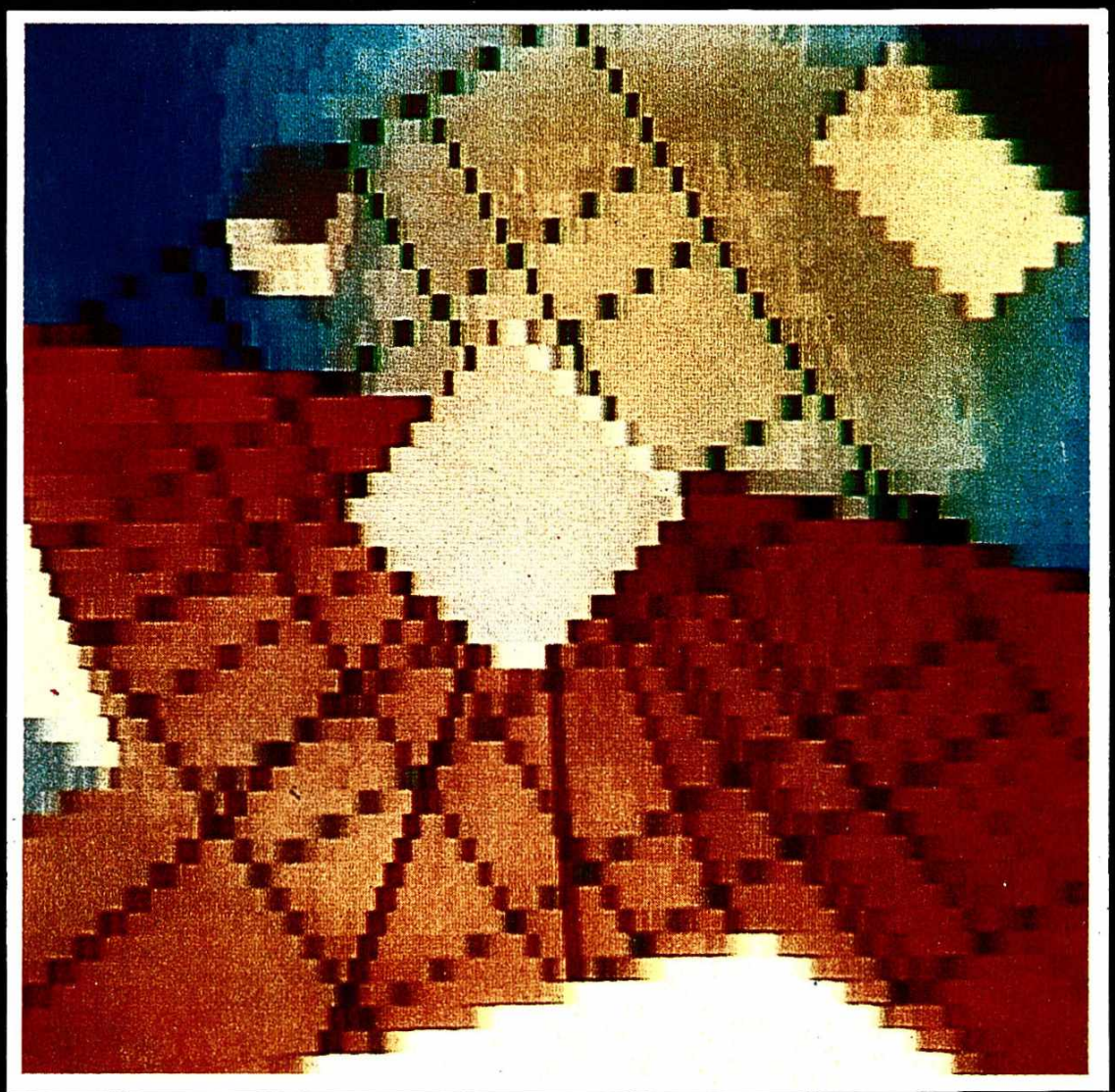
BIBLIOTECA BÁSICA

INFORMÁTICA

INTELIGÊNCIA ARTIFICIAL E SISTEMAS EXPERTOS

25



SÉCULO FUTURO

BIBLIOTECA BÁSICA

INFORMÁTICA

INTELIGÊNCIA ARTIFICIAL E SISTEMAS EXPERTOS

25

SECULO FUTURO

Diretor editor:
M. A. Nieto

Coordenação e supervisão técnica:
Engº Sergio Rocha Paggioli

Tradução:
Ideli Novo

Projetos especiais:
Rainer K.E. Ladewig

Editor Gráfico:
Auro Pereira da Silva

Revisão:
Susana M. Amaral Couto

Produção Gráfica:
Luiz Carlos Siqueira Lago

Arte-Final:
Rubens Tadeu Benedito (coordenador)
Luiz Antonio de Andrade
Vadinho de Oliveira

Fotocomposição, fotolito:
Omnicolor Gráfica e Propaganda Ltda - Rua Dr. Virgílio de Carvalho Pinto, 619
Pinheiros - CEP 05415 - São Paulo.

Impressão:
Editora Antártica S.A. - Av. Ramon Freire, 6920 (Pajaritos) - Santiago - Chile

ÍNDICE

PREFÁCIO

Desde há poucos meses cresceu exponencialmente o interesse do mundo científico e do campo da informática por um conjunto de técnicas e teorias conhecidas como Inteligência Artificial. Quais são as causas que motivaram o enorme auge desta nova disciplina?

Imediatamente, a tentativa de construir máquinas que apresentem as capacidades que associamos com a inteligência humana não é nada recente: já no século XVIII foram construídos os primeiros autômatos que tentavam emular aspectos parciais. A causa do relançamento da I.A. é proveniente da falta de algorítmos matemáticos capazes de enfrentar situações aparentemente simples, tais como o reconhecimento da linguagem, os problemas de diagnoses (enfermidades, geológicos) ou o reconhecimento visual de objetos...

O termo Inteligência Artificial se presta em certa medida a interpretações errôneas e, tal e como ocorreu com os "Cérebros Eletrônicos"; talvez dentro de uns poucos anos esta denominação tenha desaparecido e surjam outros termos mais de acordo com a realidade das pesquisas.

Hoje em dia os programas mais avançados são mais sistemas baseados no conhecimento de "experts" que sistemas inteligentes, já que suas capacidades são, ainda que de uma certa potência, muito limitados em suas aplicações, restringindo-se a temas muito concretos. Tais são os já muito famosos sistemas de diag-

noses de enfermidades infecciosas ou de prospecções geológicas, que estão sendo aplicados com muito êxito por inúmeros hospitais e pelas companhias petrolíferas.

O resultado de unir o conhecimento de um experto com os computadores mais potentes e rápidos está criando um novo ambiente que mudará completamente as estratégias de resolução de problemas antigos e além disso permitirá enfrentar a outros intratáveis na atualidade. Em todos aqueles problemas onde partindo de algumas combinações iniciais o sistema se expanda de uma forma exponencial, o computador, por rápido e potente que fosse, nunca iria por diante da crescente complexidade do modelo e, como resultado, as soluções sempre seriam parciais.

Incorporar um conhecimento, uma heurística, que permita ao programa eliminar aqueles caminhos cuja possibilidade de ser corretos é mais baixa, facilitará a resolução destas situações e impulsionará o esclarecimento de numerosas questões.

A Inteligência Artificial está sendo considerada por inúmeros países um dos aspectos chaves do progresso nos próximos anos. Dentro destes países temos que citar, por sua importância, o Japão, que incluiu de maneira preferencial os três campos da I.A. (sistemas expertos, robótica e linguagem natural) no conjunto de temas a pesquisar dentro dos planos da quinta geração de computadores.

É, portanto, importante que o leitor comece a ter uma idéia geral sobre as perspectivas e limitações que nascem com a irrupção da Inteligência Artificial no setor informático em geral e possa iniciar-se em outro dos ramos de conhecimento que são abertos a nossos olhos.

CAPÍTULO I

INTRODUÇÃO À INTELIGÊNCIA ARTIFICIAL

Desde o início da era dos computadores, os especialistas em informatica trataram de desenvolver técnicas que permitem aos computadores atuar como o faz o ser humano. Uma das bases de apoio desta nova forma de desenhar um programa é a Inteligência Artificial.

É conhecida como Inteligência Artificial uma nova forma de resolver problemas dentro dos quais são incluídos os sistemas expertos, o manejo e controle dos robôs e os processadores de linguagem natural.

Um conjunto das técnicas da I.A. está sendo aplicado recentemente na construção de sistemas expertos que permitem aos computadores ajudar ao homem a resolver problemas e na tomada de decisões.

Esta nova forma de enfrentar aos problemas produzirá nos próximos anos uma revolução dentro da nascente relação entre o usuário e o computador, comparável à introdução dos microcomputadores em todos os escritórios e lares. Por um lado ajudará na relação de situações que antes, por sua complexidade, eram quase impossíveis de tratar mediante a programação tradicional, e por outro, racionalizará e esclarecerá a informação acessível ao usuário, isto é, resumirá e simplificará a informação, destacando somente aquela que é realmente importante para as necessidades do usuário.

Considere, por exemplo, o problema de desenhar um progra-

ma de treinamento para o pessoal de vendas de uma empresa, onde os produtos e serviços para o cliente estão mudando constantemente. A técnica de vendas em geral pode ser ensinada, porém os aspectos específicos de cada produto seguramente terão mudado o tempo de terminar o curso. Imagine em lugar disto um programa capaz de interacionar com o vendedor fazendo-lhe perguntas e recomendando-lhe as opções apropriadas: seria igual que ter para cada vendedor um especialista sempre atualizado sobre os novos produtos. O programa contará com um sistema exequível e simples que pemita modificar a informação que maneja e renová-la constantemente; além disso, a mudança do programa será realizada pelos especialistas do produto na empresa e não será necessária a intervenção de um programador de computadores. Este é um exemplo possível do que chegará nos próximos anos.

Logo aparecerão também postos de informação inteligentes de cada departamento das fábricas, capazes de manejar mais dados e de forma mais complexa que os atuais computadores. A utilização destes terminais por parte dos chefes de departamento ou seção lhes permitirá controlar mais atividades e de uma forma mais ordenada, o que redundará no incremento da quantidade e qualidade das decisões que possa tomar.

Estas possibilidades que são vislumbradas estão levando a maioria das grandes empresas de todo mundo a criar departamentos de pesquisa no campo da I.A. e a resgatar das universidades o conhecimento sobre os sistemas expertos que estas possuem e que vêm elaborando desde a década dos anos sessenta. O salto das universidades à indústria leva consigo que o mercado das aplicações de I.A. esteja começando a expandir-se; os primeiros resultados práticos já estão sendo comercializados nos países mais avançados (Estados Unidos, Japão, França e Inglaterra).

A gênese da Inteligência Artificial

Nos finais da primeira guerra mundial apareceram em cenas os primeiros grupos de pesquisa que tentavam desenvolver uma máquina capaz de manejar de forma automática dados e números e efetuar complexas operações matemáticas. Dentre eles podiam ser distinguidas duas tendências: por um lado, aqueles que pensavam que as instruções fundamentais de funcionamento de uma máquina de tais características deviam ser os operadores lógicos "AND", "OR", "NOT" e, por outro, aqueles que viam muito mais vantajoso (ainda que menos potente em suas realizações) utilizar os operadores numéricos " + ", " × ", " ",...

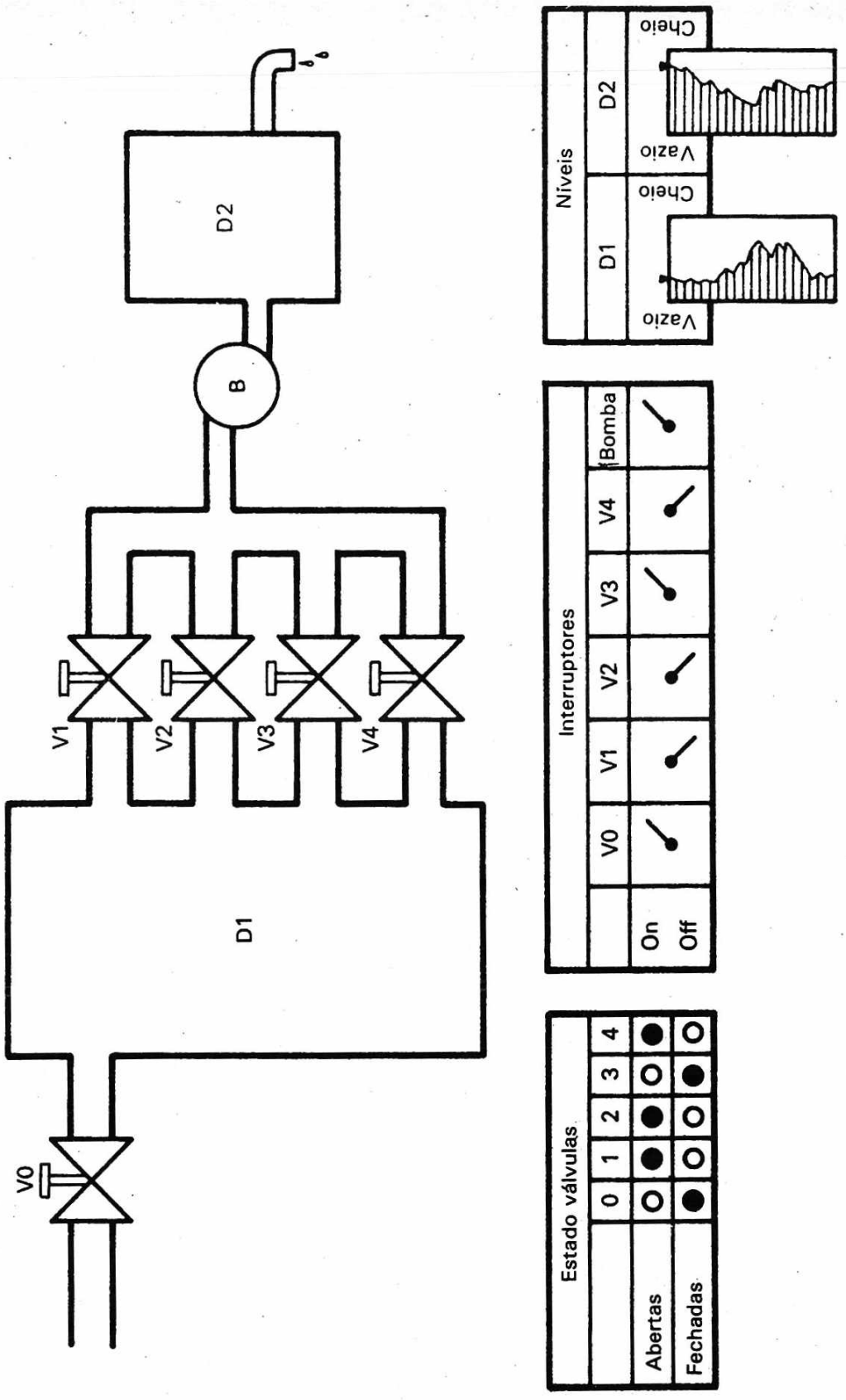


Figura 1. — Console interativo de um processo industrial. O operário pode modificar e intervir desde o console no processo que controla.

Naqueles tempos a lentidão e capacidade dos primitivos computadores fez com que fossem desenhados e construídos em base aos operadores matemáticos, que possuem uma capacidade de manejar números muito boa, ainda que falhassem bastante no manejo da lógica.

Com o passar dos anos os grupos de pesquisa baseados nos operadores lógicos foram abrindo caminho dentro das universidades e dando os primeiros passos no que hoje chamamos I.A. O caminho foi longo e custoso, já que os resultados obtidos consistiam em sistemas lentos na execução e o produto final não era o suficientemente prático. Finalmente, e devido a um fator externo como é o desenvolvimento da microeletrônica, que impulsionou a construção de uma nova geração de computadores de maior capacidade, e a sua rapidez e menor custo, foi provido à Inteligência Artificial do hardware apropriado a suas necessidades.

Tradicionalmente a Inteligência Artificial é dividida em três grandes aplicações: os processos de linguagem natural que facilitam a comunicação do computador com o usuário, a robótica e tudo o associado com a visão e manipulação de objetos e os sistemas expertos, baseados no armazenamento do conhecimento de um "expert".

Pode ser apreciada na figura 2 a cronologia e os distintos fatores que impulsionaram as técnicas de I.A. e os resultados que são esperados nos próximos anos.

Todos estes fatores que foram comentados anteriormente não teriam conseguido, no entanto, atrair a atenção das grandes empresas e do mercado informático em geral se não fosse pelo aparecimento de novos desafios aos quais as empresas do setor informático estão enfrentando. Um destes desafios, e talvez o mais importante, é a simulação de circunstâncias e problemas reais.

A simulação em um computador de uma situação real, por pequena e simples que pareça, somente é possível quando as soluções ou estruturas nas quais é expandido o problema são poucas; este tipo de problema é conhecido como de explosão combinatória. Naqueles casos onde as combinações são muitas, o computador, por grande e rápido que seja, nunca superará a complexidade do modelo.

A solução não é, como se intui, fazer mais potente o computador. Ao invés de gerar todas as soluções ou estruturas possíveis, o programa somente gerará aquelas cuja possibilidade de ser certas é mais alta. Consideremos o caso de um problema de xadrez. Como é possível que com todos os adiantamentos produzidos, um programa de xadrez não seja capaz de ganhar sempre a um gran-

Figura 2. — Evolução dos sistemas expertos, partindo da lógica e do conhecimento até chegar aos sistemas inteligentes.

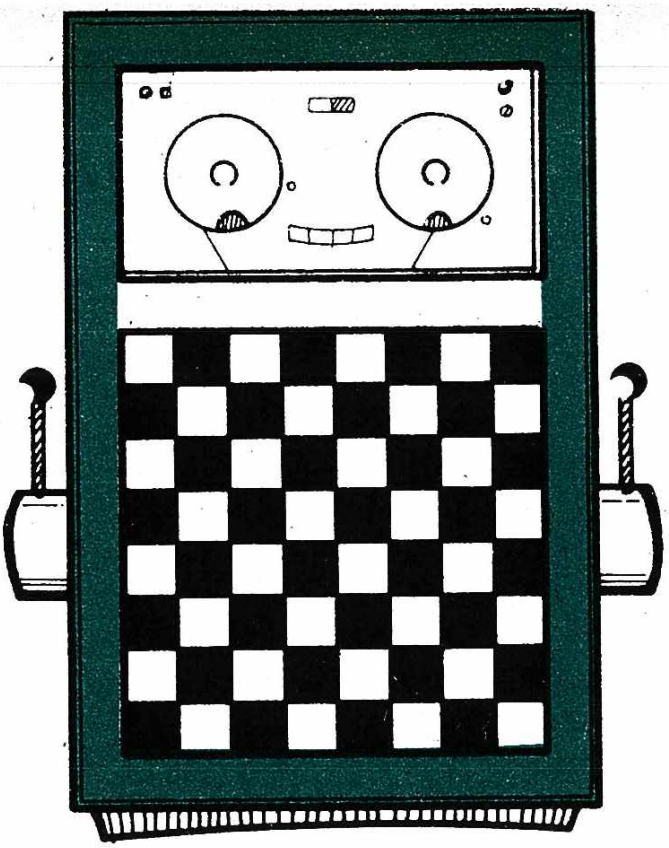


Figura 3. — Uma das últimas aplicações da I.A. são os jogos. Não lhes dão mais capacidade, e sim mais conhecimento.

de mestre? A solução é bem simples: o homem, que não possui nem a rapidez nem a memória de um computador, possui, no entanto, uma capacidade que lhe destaca; o enxadrista parte de antemão com uma heurística, tem uma experiência, um conjunto de regras, que lhe permitem eliminar das milhares de jogadas possíveis aquelas cuja possibilidade de serem corretas é mais baixa, estudando em detalhe somente um conjunto reduzido das jogadas iniciais.

Este tipo de conhecimento, a experiência de um experto em um tema determinado, a possibilidade de incorporar um conhecimento ao programa é o que levou ao ser humano a pesquisar no campo da Inteligência Artificial, cujo desenvolvimento é observado no quadro da figura 4.

A I.A. trata do desenho de sistemas informáticos inteligentes, isto é, sistemas que apresentam as características associadas com a inteligência humana: entendimento da linguagem, aprendizagem, raciocínio, resolução de problemas, etc.

Período	Sucessos Chaves
Antes segunda guerra mundial	Lógica formal Psicologia do conhecimento
O pós-guerra 1945-1954, início da I.A.	Desenvolvimento do computador Conferências sobre cibernética
Os anos de formação, 1955-1960	Aumenta a disponibilidade de computadores Linguagem de processamento de informação (IPL-I) Psicologia do processamento de informação
Os anos do desenvolvimento, 1961-1970	LISP Heurística Robótica Solução de problemas de xadrez DENTRAL (Stanford)
A especialização e os sistemas expertos, 1970-1980	MYCIN (Stanford) MACSYMA (MIT) Engenharia do conhecimento EMYCIN (Stanford) PROLOG
A comercialização, 1981	PROSPECTOR (SRI) Projeto japonês da quinta geração INTELLECT (A.I.C.) Sistemas inteligentes de recuperação de informação Diversas companhias comercializam ferramentas para a construção de sistemas expertos.

Figura 4. — Aspectos chaves na história da Inteligência Artificial.

Dentro deste conceito apareceram duas escolas. Uma, chamada de síntese, que trata de construir sistemas que apresentem inteligência independente de se usam ou não internamente os mesmos métodos que o homem, e outra, conhecida como de simula-

ção, que tenta emular a forma na qual o ser humano apresenta a faculdade de ser inteligente.

Desafios enfrentados pela inteligência artificial

De todas as faculdades que fazem ao ser humano inteligente, os pesquisadores em I.A. estão centralizando seus estudos em três grandes grupos.

- Comunicação e percepção.
  - Linguagem natural.
  - Visão.
  - Manipulação.
- Raciocínio simbólico.
- Engenharia do conhecimento.

1. Comunicação

Desde o início da década dos cinquenta, quando pela primeira vez se tentou construir um sistema de tradução automática mediante a utilização do computador, os programas que entendem uma linguagem humana foram convertidos em um dos grandes desafios da informática. Logo foi vista a forte relação entre este campo da I.A. e a lingüística, o que está ajudando inclusive a compreender a natureza da linguagem humana.

Os japoneses, em concreto em seu plano de quinta geração de computadores, incluiram à comunicação dentro dos temas cujo avanço é necessário para o desenvolvimento da ciência, e não somente os japoneses, mas também os países da Comunidade Econômica Européia e os americanos consideram este campo fundamental como ferramenta de progresso. Dentro destes planos são incluídos três grandes projetos:

— Tradutor automático multilingüe, com uma capacidade de 100.000 palavras.
— Sistemas de consulta sobre diversos temas, com mais de 5.000 palavras e 10.000 regras de inferência.
— Sistema capaz de falar e entender a linguagem natural, com cerca de 10.000 palavras de vocabulário.

No entanto, e apesar destes projetos, apareceram autores que se opõem à utilização direta da linguagem natural baseando-se em que dificultará a comunicação ao invés de torná-la mais simples. Sur-

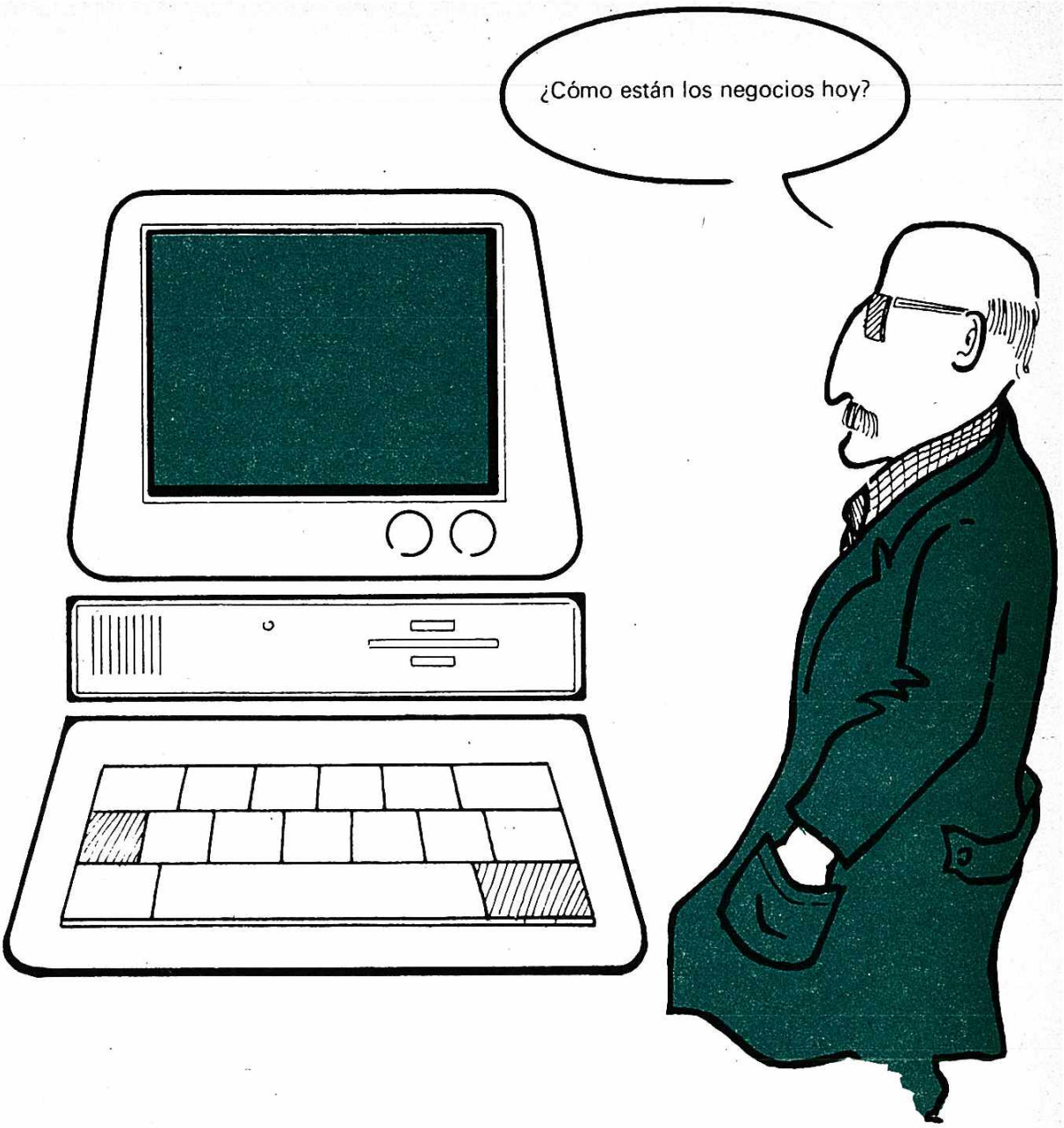


Figura 5. — A comunicação entre o homem e o computador em nossa própria linguagem é um dos aspectos chave para os japoneses em seu projeto de quinta geração.

giram problemas tais como análise sintática e semântica das orações, ambiguidade de palavras com múltiplos significados, a determinação do significado de palavras simples onde o contexto geral é o que as condiciona, etc. A linguagem humana, ainda que boa para comunicar-nos, é demasiadamente ambigüa e inexata, não muito apropriada para transmitir ao computador instruções precisas sobre o que deve fazer.

2. Raciocínio Simbólico

Na informática tradicional os computadores trabalham com programas nos quais os dados acedem diretamente ao computador, porém as decisões sobre como processar ditos dados estão impressas na linguagem do programa e armazenadas na memória durante o processamento de execução. Por exemplo, o programa que gestiona as contas correntes em um banco recebe todas as noites uns dados diferentes, porém a forma de processá-los é invariavelmente a mesma todos os dias.

A Inteligência Artificial tenta integrar o conhecimento no sistema; em outras palavras, um sistema inteligente que escreve seu próprio programa.

Os sistemas inteligentes são baseados em regras heurísticas, em contraste com os programas de cálculo numéricos, baseados no uso de equações analíticas. A heurística faz pé-firme, dentro do programa, nos aspectos do problema que parecem mais críticos e nas partes da base de conhecimento que parecem mais relevantes, e guia ao programa nos casos particulares excluindo certos caminhos e centralizando-se em outros. O resultado é que o programa segue uma linha de raciocínio ao invés de seguir uma seqüência de passos fixos no cálculo.

3. Engenharia do conhecimento

Sob a denominação de engenharia do conhecimento são agrupadas todas as áreas que intervêm no desenvolvimento dos sistemas expertos e bancos de conhecimentos. Dentro destas áreas aparecem os seguintes campos:

- Representação do conhecimento. Estão sendo desenvolvidas na atualidade uma grande variedade de métodos, todos eles para facilitar o raciocínio simbólico e permitir a codificação e aplicação do sentido comum. Os esquemas mais habituais de representação são as regras de produção, as estruturas lógicas e o cálculo de predicados.
- Aquisição do conhecimento. Obter o conhecimento necessário para a criação de um sistema experto não é uma tarefa simples. Em alguns temas o sistema pode aprender através da experiência, mas normalmente o experto e o programador do sistema devem trabalhar unidos para conseguir condensar o saber em algumas certas regras lógicas. Atualmente se está trabalhando sobre certos programas que

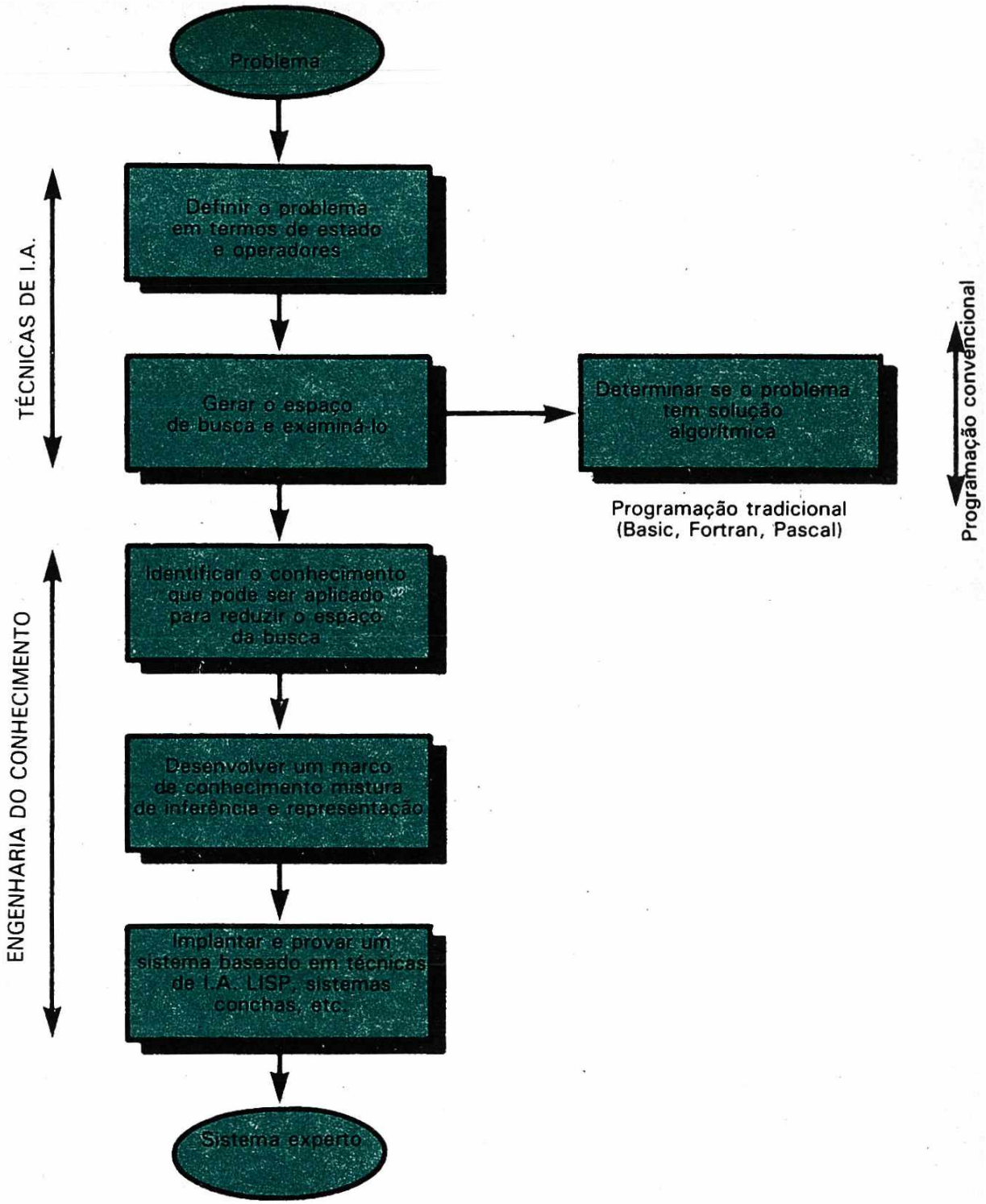


Figura 6. — A definição do problema é um dos aspectos chave no desenvolvimento dos sistemas expertos.

recebem a sabedoria do perito mediante sessões de ensino. O computador vai perguntando, analisando as respostas e incorporando-as ao banco em forma de regras lógicas.

- Métodos de inferência. São os métodos que traçam uma linha de raciocínio a uma pergunta dada. Estas técnicas de geração e análise de hipóteses são técnicas de concentração, de igual forma que as heurísticas já mencionadas, isto é, centralizam a atenção em determinadas regras e registros, marcando a linha de ação. Um problema associado a estes métodos é a quantificação e manejo de dados indeterminados, para o qual, ainda que às vezes é utilizado o teorema de Bayes ou a análise de decisão, se requer geralmente o uso da lógica difusa (Zadeh, 1965).

Fronteiras da Inteligência artificial

Ainda que seja difícil na atualidade predizer o que serão capazes de fazer os sistemas inteligentes dentro de suas múltiplas aplicações, de certa forma, a partir da análise do que é possível e o que não dos sistemas atuais, podem ser extrapoladas algumas idéias ao que será o futuro da I.A. nos próximos anos.

Atualmente, em que a tecnologia não fez nada mais que começar, as aplicações são muito restringidas; os sistemas baseados no conhecimento somente são capazes de enfrentar e resolver problemas dentro de um campo muito delimitado. Não são capazes de raciocinar a partir de axiomas ou teorias gerais e somente aceitam um tipo de fato e heurística determinados. Não têm desenvolvida a faculdade de aprender nem podem raciocinar por analogia.

A conseqüência de todos estes fatores é resumida em duas características:

a) A falta de sentido comum.
b) Seus raciocínios são deteriorados rapidamente quando o problema sai fora da tarefa específica para a qual foi desenhado.

Por outro lado, os sistemas baseados no conhecimento não emitem juízos incoerentes, não saltam nenhuma possibilidade nem são empenhados em manter uma postura contra dos fatos reais. Tampouco têm maus dias, sempre lembram todos os detalhes e sistematicamente consideram todas as possíveis alternativas. Os melhores sistemas expertos, que contêm milhares de fatos e regras e sobretudo estão desenhados para uma função apropriada (como, por exemplo, toda a classe de diagnósticos médicos, geológicos, etc), realizam as mesmas funções que o melhor experto e seus juízos e conclusões, são equiparáveis e inclusive melhores que a média de especialistas em dito tema.

CAPÍTULO II

CONCEITOS BÁSICOS

Da Inteligência artificial aos sistemas expertos

Quando nasceu a ciência da informática se pensou que as novas máquinas, capazes de realizar operações aritméticas por si só e com uma certa capacidade de operação, iam resolver absolutamente todos os problemas que tinha o homem.

Começou-se a pensar então em computadores que falassem em todos os idiomas da terra, robôs "amas de casa", grandes "andróides" (metade homem, metade robô) capazes de trabalhar nas piores circunstâncias, etc. No entanto, esta euforia inicial, causada principalmente pelo desconhecimento real do novo ramo da ciência, acabou muito rápido: Um computador tão grande como uma casa (o ENIAC, por exemplo) necessitava "uma eternidade de tempo" para fazer uma simples multiplicação.

Posteriormente, passados alguns anos de estudos sobre o caminho que deveria tomar esta ciência em sua evolução foram resolvidos os primeiros problemas físicos (como a passagem das válvulas aos transistores e destes últimos aos circuitos integrados), começou novamente outra euforia sobre a capacidade limite dos novos computadores. Poderia ser conseguido um computador que fosse capaz de armazenar toda a informação que existe na Biblioteca Nacional de forma que o leitor não tivesse que deslocar-se ao edifício da Biblioteca, mas que de sua casa poderia consultar qualquer livro que estivesse armazenado na memória do "maravilho-

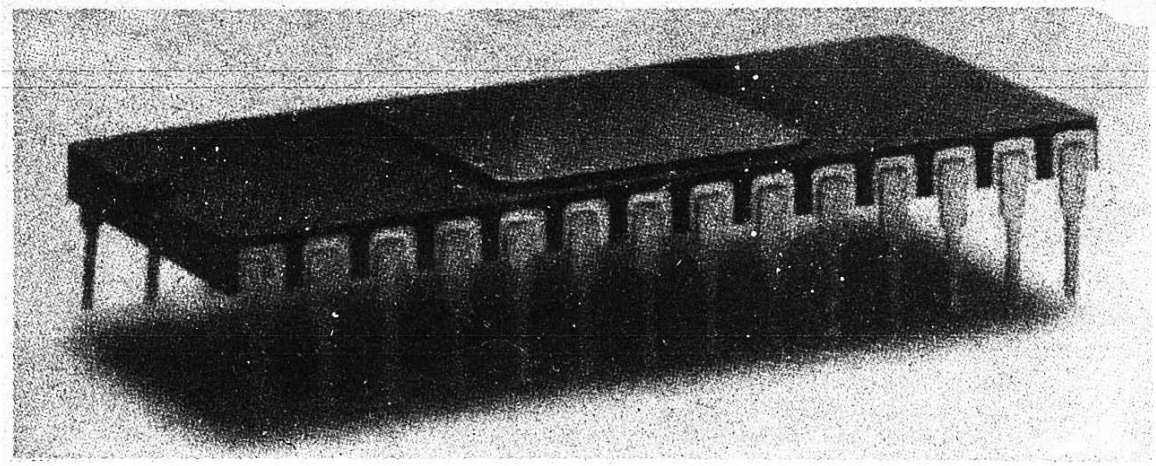


Figura 1. — Já nos mais famosos "chips" se encontra o presente (talvez o futuro) da informática. Sua capacidade de operação está crescendo dia a dia.

so computador". E não somente se pensou nestas tarefas: os computadores seriam capazes de saber tudo. Se alguém necessitasse saber quantos gols fez um time de futebol em um campeonato qualquer e quantos recebeu, bastaria perguntar ao computador.

Novamente o tempo voltou a demonstrar aos desmesuradamente otimistas que o que eles desejavam todavia ia demorar muitos anos para poder ser realizado.

No entanto, uma "parcela" destas magníficas idéias conseguiu propagar-se em alguns profissionais da informática.

Eles pensaram: "de imediato é muito difícil que um computador saiba tudo sobre tudo o que acontece no mundo. No entan-

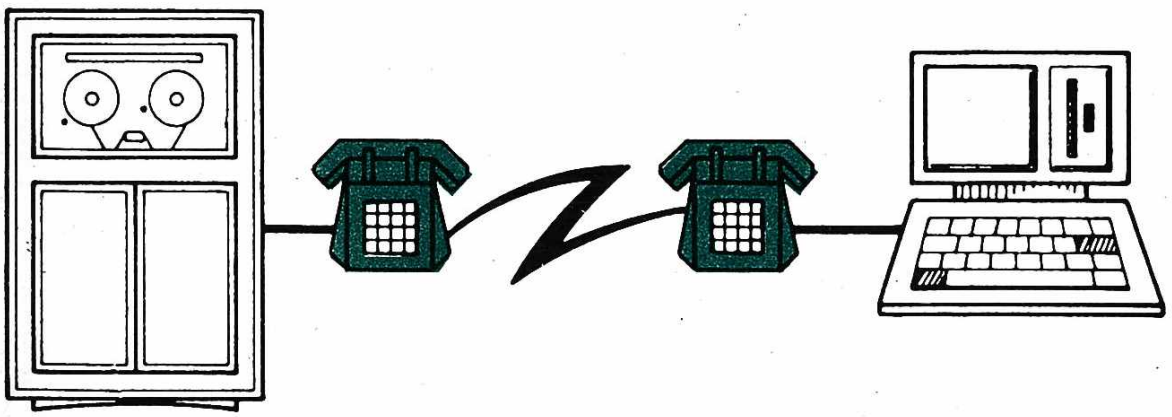


Figura 2. — Dos métodos antigos de manejo da informação aos que são utilizados na atualidade existe uma grande diferença. A comunicação de computadores situados em distintos lugares permitiu um grande avanço.

to, poderia ocorrer que um computador fosse programado para que soubesse tudo sobre um só tema".

Assim, pois, poderíamos dispor de um computador que somente soubesse sobre as partidas que são jogadas na Copa do Mundo, e nada mais do que isso. Se ao computador perguntássemos quem vai ser o primeiro na Copa, ele não o saberia, posto que não estuda o tema "partidas da Copa".

Este tipo de tarefas foi realizado rapidamente, posto que a única coisa que se necessitava era memória suficiente no computador para armazenar toda a informação (sobre as partidas da Copa jogadas, por exemplo). Porém alguém pensou se não seria possível que o computador fosse capaz de fornecer informação mais inteligente sobre o tema tratado: Tentar-se-ia que o computador desse sua opinião sobre quem ganharia a Copa. Para conseguir isto havia necessidade que a máquina trabalhasse de forma inteligente com os dados que possui (que são os resultados das partidas da Copa) para obter uma informação que não possuia previamente.

Pois bem, na tentativa de criar um sistema baseado em computadores que se ocupe de um determinado tema e que seja "medianamente inteligente" foi chamado sistema experto sobre esse tema.

O de "expert" pode referir-se a que o computador seja convertido em um perito sobre o tema tratado devido à grande quantidade de informação sobre essa matéria que possui.

O que é um sistema experto

Um sistema experto pode ser definido, mais concretamente, como uma estrutura de programação capaz de armazenar e utilizar, com menos restrições das presentes em um programa clássico, algum tipo de conhecimento sobre uma determinada área.

Expliquemos um pouco esta definição. Como qualquer outro pacote de trabalho sobre um computador, o sistema experto é uma estrutura de programação, ou melhor, é um programa. No entanto, não é um programa de computador normal como o que fazemos no computador que temos em casa, posto que sua estrutura é diferente: não tem as mesmas partes que têm os programas clássicos.

Esta estrutura de programação é capaz, basicamente, de duas coisas:

- pode armazenar informação, como se fosse um banco de dados, sobre o tema considerado;

- tem a faculdade de utilizar essa informação para obter alguns resultados que não existiam previamente no computador. Isto o realiza mediante uma técnica que é a que realmente diferencia aos sistemas expertos dos programas clássicos.

Outras características específicas dos sistemas expertos são encontradas, além das já explicadas de uso geral (armazenar e utilizar), na capacidade de aprendizagem e o poder de inferência.

A capacidade de aprendizagem de um sistema experto, ainda que pareca uma qualidade futurista e extravagante, resulta fácil de implantar; se converte em uma ferramenta muito útil na hora em que o sistema melhore com relação ao momento de sua criação. Em qualquer caso, a capacidade de aprendizagem é limitada e, além disso, deve estar sujeita ao controle do engenheiro que gestiona o sistema experto.

É importante diferenciar ao usuário do "engenheiro do conhecimento". O usuário é a pessoa que utiliza o sistema, como se fosse o dono de um casa. O engenheiro do conhecimento se ocupa de que o sistema funcione como quer o usuário (seria o arquiteto, o

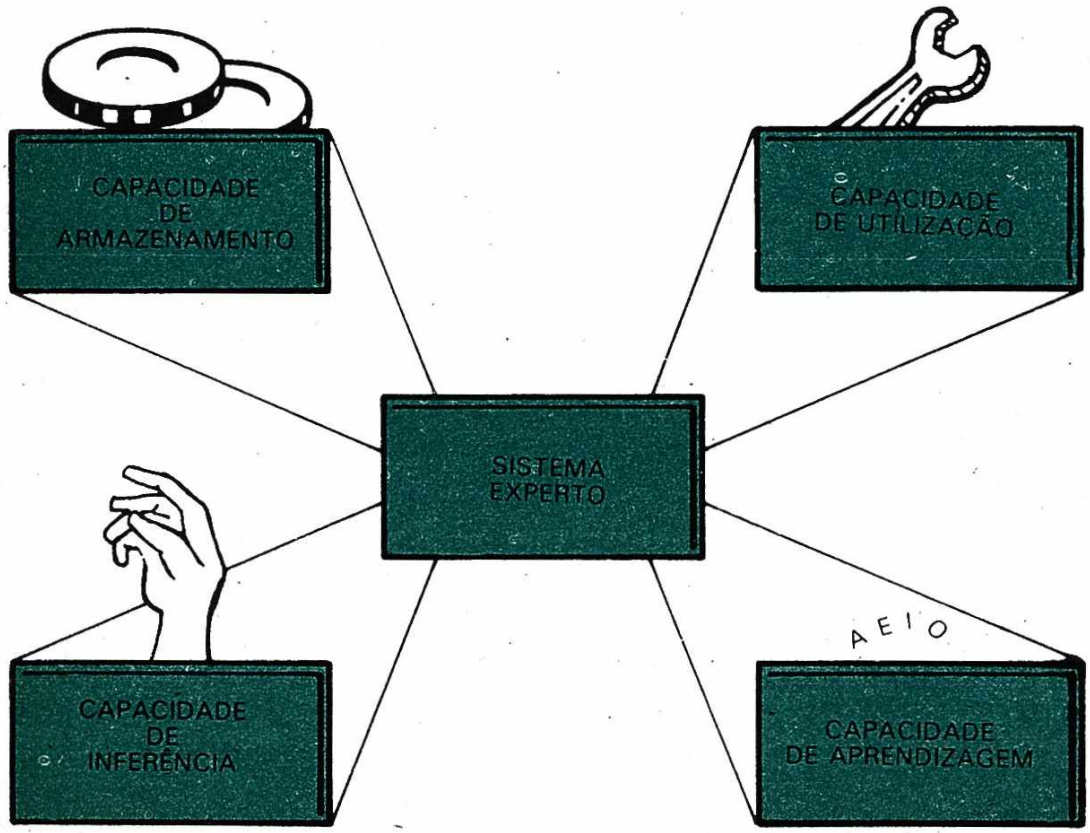


Figura 3. — Capacidades dos sistemas expertos.

encanador, o carpinteiro, o pedreiro... todos os que criam a casa e a arrumam para que o dono se sinta cômodo).

De momento, e talvez por muitos anos, os computadores não podem tirar informação *completamente* nova em relação da que possuem armazenada. Se ao nosso famoso computador, o que armazena todas as partidas que foram jogadas da Copa do Mundo, lhe perguntarmos quem acredita que irá ganhar a Copa do ano que vem, poderá contestar qualquer equipe que participou da competição, porém não poderá responder com uma equipe que não existe. Isto é muito importante. Assim, pois a aprendizagem será, precisamente, o que permita ao computador realizar as seguintes tarefas:

- completar a informação armazenada;
- utilizar nova informação que necessite para processos de inferência.

A nova informação que aprende o sistema pode vir por dois caminhos diferentes:

- Do próprio usuário
  Em alguns processos o computador pode ser consciente de que nova informação pode mudar o resultado. O computador "copeiro" poderia perguntar ao usuário, antes de dar alguma solução: Haverá alguma nova equipe em Primeira Divisão o ano que vem? Assim, no caso de que exista alguma, esta será introduzida pelo próprio usuário e o sistema a incorporará a seus conhecimentos e o terá em conta como possível solução.
- De um processo interno do computador
  Neste caso não se consegue informação completamente nova, mas resultante de interrelação de informação previamente acumulada.

Para oferecer um exemplo deste segundo método vamos criar outro computador que se ocupe de resolver problemas médicos.

O usuário, neste caso um médico, introduz ao sistema os sintomas do paciente e o próprio sistema expressa sua opinião indicando uma enfermidade. Este "médico eletrônico" poderia aprender da seguinte maneira: Suponhamos que a informação que possui o sistema seja

O frio produz catarros
$\longrightarrow$

Os catarros mal curados produzem bronquite

Se um paciente diz que há uma semana passou muito frio e que posteriormente ficou mal, o computador pode resolver que o paciente possui bronquite. Esta informação, que induziu o próprio computador, poderia aprendê-la se criasse um novo fato de informação que fosse:

O frio produz bronquite.

Este "médico eletrônico" nos serve além disso para explicar brevemente a segunda grande qualidade específica dos sistemas expertos: o poder de indução. Podemos apreciar como a partir de uma informação que introduziu o usuário (há uma semana passou muito frio) e de algumas relações entre fatos de informação que possui, o computador consegue uma solução, combinação de tudo. À ação de combinar a informação para conseguir novos fatos é chamada indução.

Estrutura de um sistema experto

Uma vez explicada a teoria dos sistemas expertos e descritas as possibilidades e características mais importantes, resta por definir como está formado um sistema experto. Conhecer a estrutura geral de um sistema deste tipo permite esclarecer a compreensão de seu funcionamento.

O sistema experto está formado basicamente por dois elementos:

- um banco de conhecimento,
- um motor inferencial.

Analisemos cada um deles:

- Banco de conhecimentos.
  Nele é armazenada a informação referente ao tema tratado em cada caso pelo sistema. A pergunta poderia ser: Por que é chamado banco de conhecimentos e não banco de dados, se ao fim e ao cabo o que é armazenado em dito banco é informação? Com respeito a este tema todavia não foram esclarecidos os cientistas. Inclusive não foram colocados de acordo ainda com respeito aos elementos simples que engloba o próprio banco de conhecimento. Para apro-

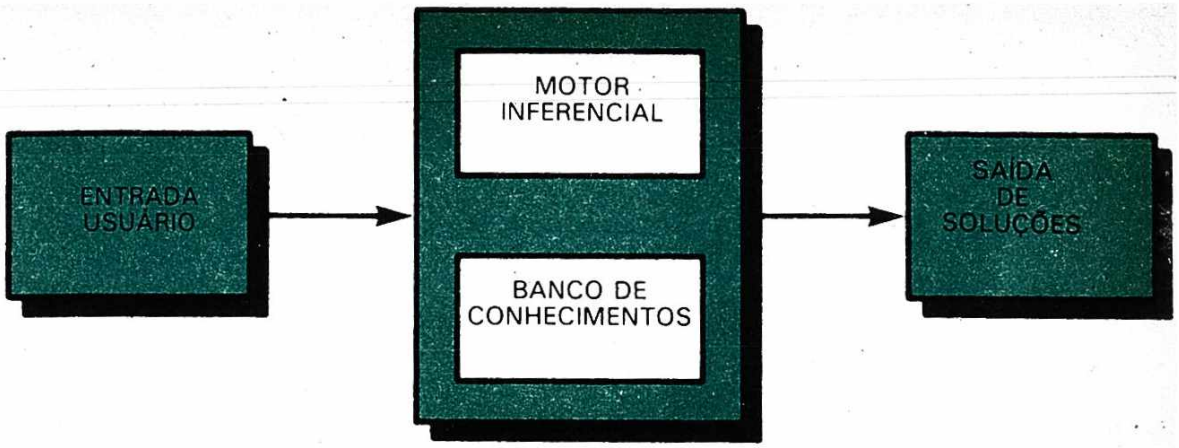


Figura 4. — O esquema geral de um sistema experto apresenta um diagrama de blocos bastante simples.

fundar neste problema temos que conhecer os níveis do conhecimento diferenciados em inteligência artificial, como faremos posteriormente.

- Motor inferencial

É ocupado de gestionar o banco de conhecimentos de modo que ante perguntas do usuário o sistema ofereça respostas aceitáveis. Isto é, se ocupa de tomar as regras e realizar inferências para que tenhamos uma solução. Mais adiante serão explicadas estas funções com mais detalhe.

Níveis de conhecimento

Existem três níveis de conhecimento que estruturam a informação que maneja um sistema experto.

- **Fatos básicos de informação**

É o nível mais baixo dentro da hierarquia de conhecimentos. Representam simplesmente fatos básicos, isto é, os elementos lingüísticos mais simples que possuem sentido por eles mesmos. São a base da informação.

Exemplos de fatos básicos são:

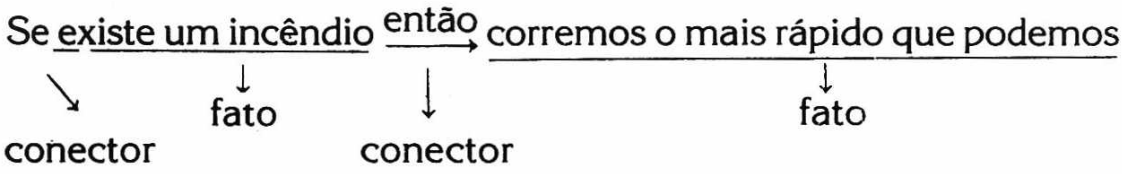
"Nuria está como uma foca"
"Nico tem óculos"
"Marcos é diplomático"

• **Regras de conhecimento**

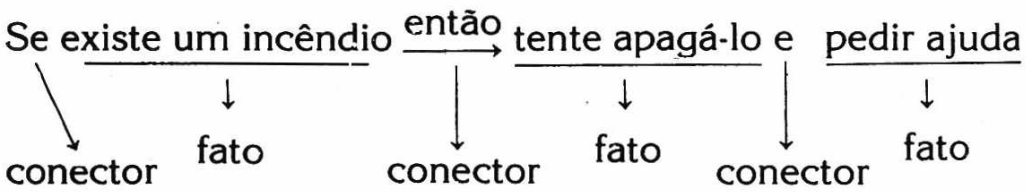
São blocos lingüísticos que representam um determinado tipo de informação em função da forma em que são construídos. Estão formados por fatos básicos, ainda que possa ocorrer (de fato ocorre em alguns sistemas expertos) que as regras de conhecimento podem por elas mesmas representar a unidade mínima de conhecimento com a qual trabalha um determinado sistema experto. Uma regra de conhecimento pode representar o que temos que fazer em caso de incêndio; e somente representará isso. No caso de que ocorresse sua inundação não poderíamos aplicar esta regra.

Para criá-la teremos que utilizar como mínimo dois fatos básicos: a primeira representará o que fazer realmente no caso de incêndio; a segunda servirá para ter claro que a regra criada somente deve ser utilizada quando existe um incêndio e não quando existe um temporal, por exemplo.

Assim poderíamos criar a seguinte regra:



Ainda que poderíamos construir outra regra como:



onde a própria regra está formada por mais de dois fatos.

Vemos como mediante a informação das regras de conhecimento é constituído o conhecimento do sistema sobre um tema escolhido (no caso anterior, seria como tratar um incêndio).

Ainda que será explicado posteriormente o significado dos conectores, já vemos que sua função é unir diferentes fatos para criar uma estrutura (a regra de conhecimento) que possua sentido em seu conjunto e que além deste deve ser o que pretendemos que tenha.

Pode ser visto que a estrutura básica das regras de conhecimento é:

Se [fatos-1] então [fatos-2]

onde [fatos-1] é o primeiro conjunto de fatos, que recebe o nome de antecedente, e [fatos-2] é o segundo, que é chamado conseqüente.

- **Regras de controle**

Representam o último nível de informação na estrutura do conhecimento aqui tratada. Uma vez constituídas as regras de conhecimento sobre um tema, são armazenadas todas elas na memória do computador. No entanto, ante uma pergunta do usuário terá que utilizar, para responder-lhe, somente algumas quantas das regras que são possuídas, não todas.

Que regras devem ser utilizadas para responder ao usuário? O sistema necessita um instrumento capaz de escolher em cada caso a regra que corresponda.

Segundo o exemplo do incêndio, suponhamos que no computador temos duas regras de conhecimento:

Regra A; "Se existe um incêndio então corra muito depressa"

Regra B: "Se existe uma inundação então aprenda rapidamente a nadar"

Em cada caso o computador deveria analisar as regras e escolher a que realmente tem que aplicar. Seria muito problemático que alguém em um incêndio tentasse aprender a nadar (ainda que provavelmente não seria queimado).

A escolha das regras adequadas para cada caso é realizada no computador por meio de outras regras, chamadas regras de controle.

As regras de controle, como seu nome indica, controlam a escolha das regras de conhecimento. Por exemplo, uma regra de controle poderia ser:

'Quando ocorrer um fato «A» então aplicar todas as regras em que apareça como antecedente o próprio fato «A».

Assim, no caso de que ocorresse um incêndio seriam aplicadas todas as regras onde aparecesse:
"Se existe um incêndio..."

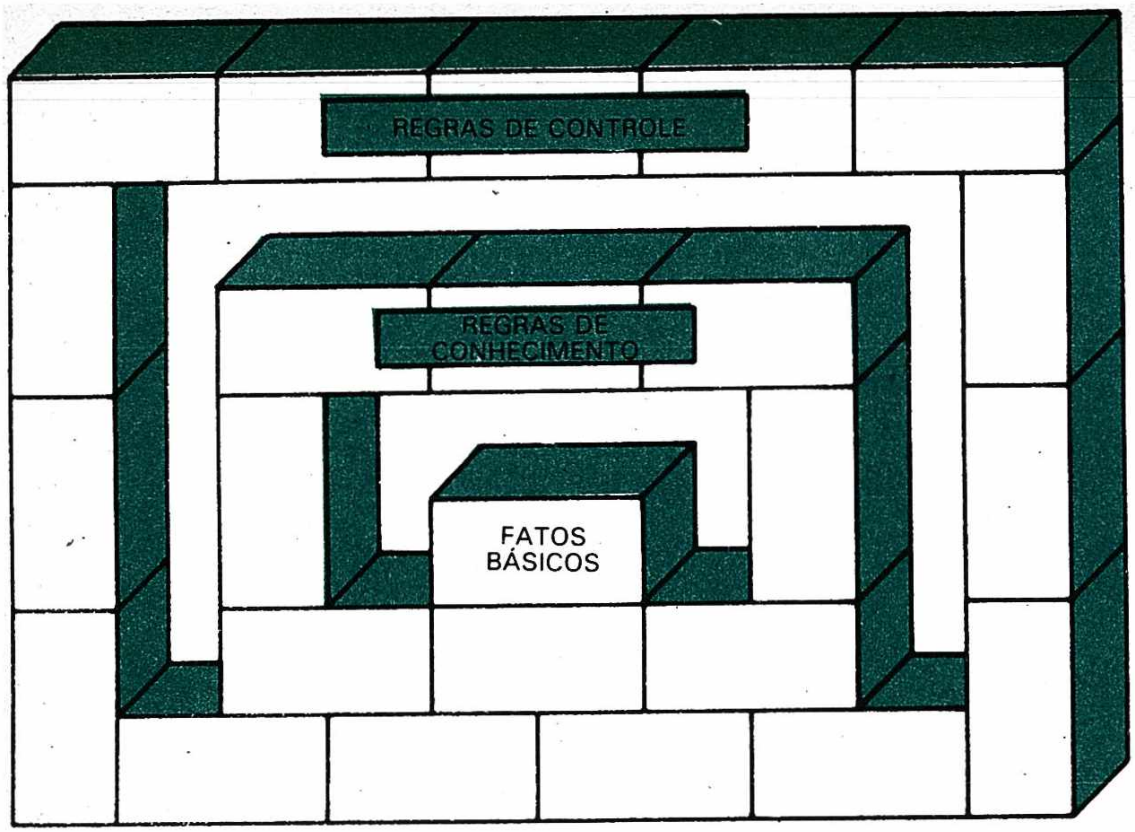


Figura 5. — A estrutura que possuem os níveis de conhecimentos de um sistema experto tem uma forma envolvente.

A estrutura de conhecimentos utilizada é englobada na forma que ilustra a figura 5.

Atribuição de níveis

Uma vez definidos os níveis de conhecimento, o seguinte passo é atribuir-lhes um lugar na estrutura do Sistema Experto.

Aqui começam as divergências de opinião entre os cientistas. Para alguns dita atribuição é:

Fatos básicos formam o banco de dados associado ao sistema

Regras de conhecimento formam o banco de conhecimentos

Regras de controle formam o motor de inferência

Isto é, criam um banco de conhecimentos formado por regras e, além disso, o sistema possui um banço de dados associado.

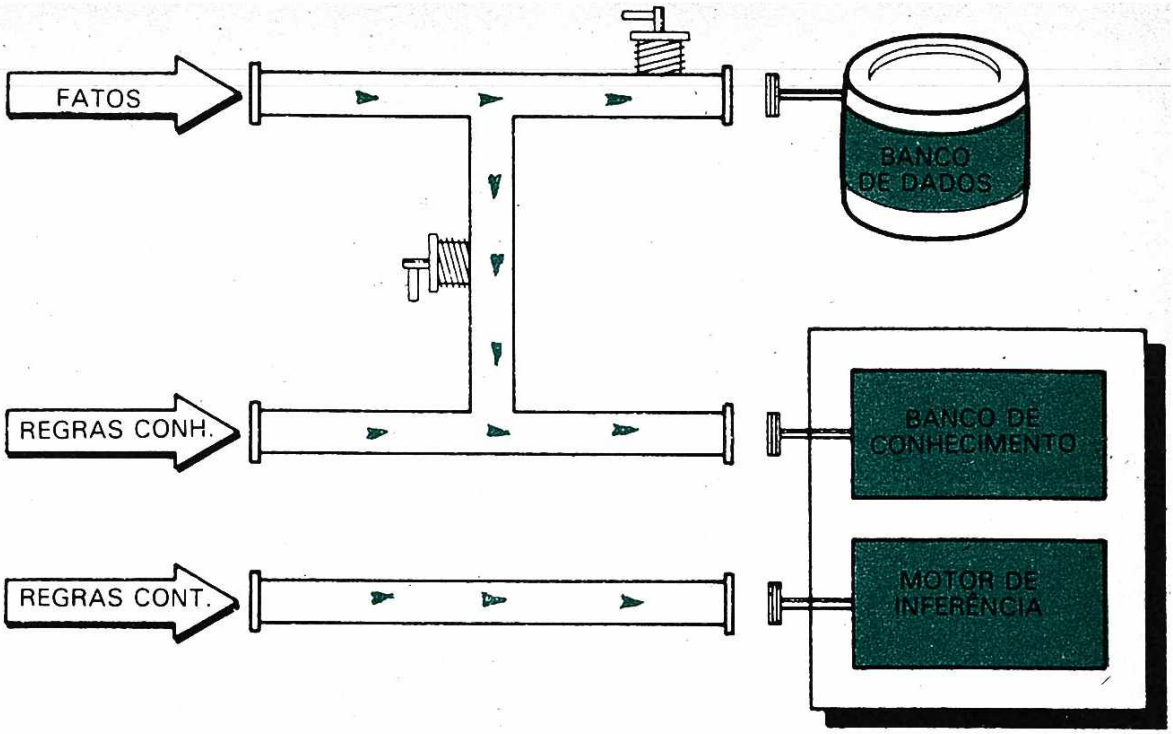


Figura 6. — A atribuição dos níveis de conhecimento a cada elemento do sistema experto pode ser representada mediante uma canalização de tubulações.

Para outro grupo de cientistas a atribuição é realizada a seguinte maneira:

Fatos básicos
Regras de conhecimento | formam o → banco de conhecimentos
Regras de controle formam o → motor de inferência

Nesta atribuição o banco de conhecimentos está formado pelos fatos básicos e pelas regras de controle.

Para manejar corretamente a inferência existem grandes quantidades de ferramentas lógicas e matemáticas estudadas neste século. Entre elas são encontradas a dedução, a indução, etc. Todos estes métodos matemáticos podem ser implantados nos sistemas para que trabalhem com as regras e produzam soluções.

Uma vez estudadas a estrutura e qualidades mais importantes dos sistemas expertos podemos realizar um gráfico que aglutine perfeitamente o trabalho do sistema, tal como o mostrado na figura 7. Se nos fazemos a famosa a pergunta: por que ao banco

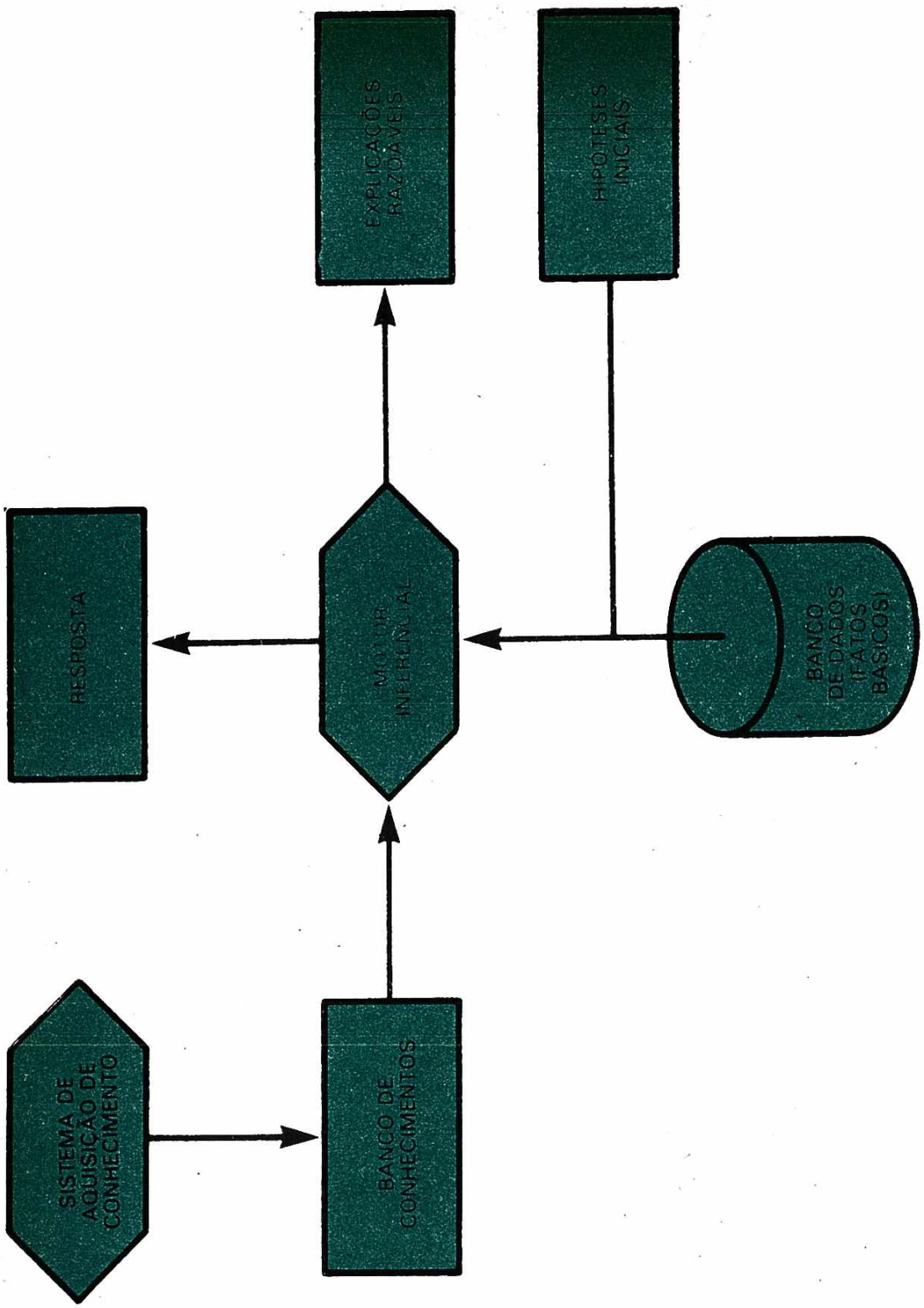


Figura 7. — A figura expressa a estrutura detalhada de um sistema expert em função das facilidades que apresenta em seu funcionamento.

de conhecimentos não chamamos banco de dados? compreenderíamos agora que nos sistemas expertos o banco de conhecimen-

tos significa algo mais que informação armazenada, posto que já inclui uma certa "inteligência" em como e quando utilizar a informação. Por isso é chamada precisamente *Banco de Conhecimentos.*

Diferenças estruturais com a programação clássica

Os sistemas expertos representam um passo adiante na ciência da programação por sua grande diferença com os programas clássicos.

Na programação normal para resolver um problema é criado um programa. Neste programa, que está constituído por linhas em BASIC, FORTRAN ou outra linguagem, são encontrados formando uma unidade tanto a informação sobre o método para resolver o problema como os próprios dados particulares do problema.

No entanto, nos programas de inteligência artificial (e mais concretamente nos que constituem os sistemas expertos se consegue diferenciar o método de resolução de problemas (uma es-

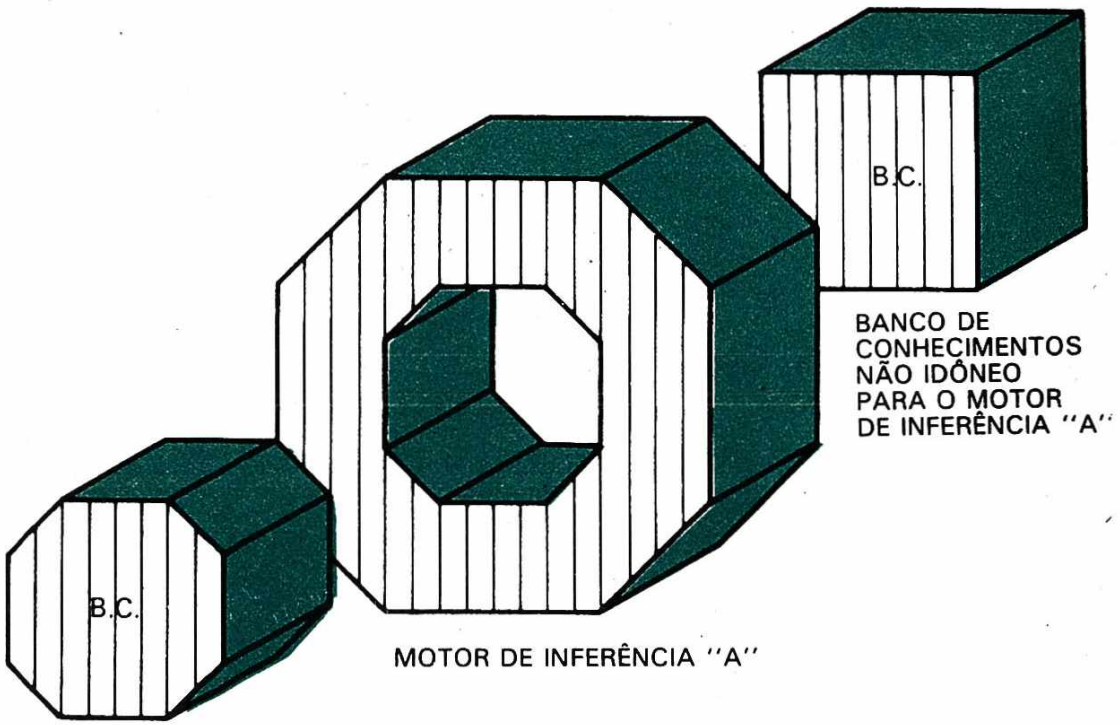


Figura 8. — O banco de conhecimentos é o elemento diferenciador dos múltiplos sistemas expertos. Não são adaptados igualmente a um mesmo motor inferencial.

pécie de "solucionador de problemas" porém sem problema a resolver) da informação característica do problema. Isto é muito importante, posto que facilita enormemente a evolução do sistema ao ser modulada sua estrutura.

Assim pois, o motor de inferência é construído como um ente isolado e independente do banco de conhecimentos. Isto permite criá-lo "fazendo a vez" de outras partes do sistema e assim é agilizado o processo necessário.

No entanto, temos que reconhecer que não existe 100% de independência entre o banco de conhecimentos e o motor de inferência, já que, como ocorre com tudo nesta vida, alguns motores inferenciais funcionam melhor com um determinado tipo de banco de conhecimento que com outro e, portanto, são criadas algumas pequenas diferenças entre os motores em função da forma que tomam o conhecimento sobre o tema.

Já temos definidas as diferenças de estrutura entre os programas clássicos e os sistemas expertos. No entanto, não está muito clara a grande vantagem destes sistemas. Podem pensar vocês que muito definir e estruturar mas pouco diferenciar realmente seu funcionamento; pelo visto até agora poderia ter sido feito um programa normal estudando por um lado a mecânica e armazenando por outro os dados do programa. No entanto, "sim" existem grandes diferenças de funcionamento.

Estrutura sol

Suponhamos que queremos realizar um sistema experto em diagnósticos médicos; voltamos a trabalhar com nosso Médico Eletrônico. O sistema experto que vamos realizar, como somos muito "chulos", poderá resolver qualquer caso sobre qualquer tipo de enfermidade (imaginem o que pedimos!); ainda que não se tenha conseguido, na realidade, nós já o temos.

Para ensinar ao sistema a primeira enfermidade que deve conhecer teríamos que recolher todos os sintomas que conhecemos sobre ela para em seguida introduzí-los no computador. Tomemos, exemplo, a gripe;

- febre não muito alta;
- mal-estar geral;
- dores em todo o corpo;
- tosse, etc.

Mas, quem nos impede de realizar um programa clássico que

Figura 9. — Estrutura de um programa clássico. O bloco básico de inferência corresponde a um programa simples que realiza uma única função.

comprove se temos gripe? Poderia tratar-se de um programa que funcionasse de forma que: se entre os dados de entrada que dê o usuário se encontra algum sintoma que defina a gripe então diria simplesmente que temos gripe. Em caso contrário concluiria que não temos gripe.

Esta estrutura (Fig. 9) é a normal em um programa clássico. O que acontece se além de analisar se o paciente sofre de gripe, o sistema estuda a possibilidade de que tenha artrite? Novamente a primeira coisa a fazer é recopiar os sintomas da artrite (dor de articulações e ossos, perda da mobilidade, etc.); então poderia ser criado outro "programa" para reconhecer se o paciente possui artrite, que poderia ser juntada como o que tínhamos. Assim poderíamos saber se o paciente possui gripe ou artrite.

Porém existe um problema: o programa global deve analisar *em primeiro lugar uma das duas enfermidades*. Isto é, deverá ver

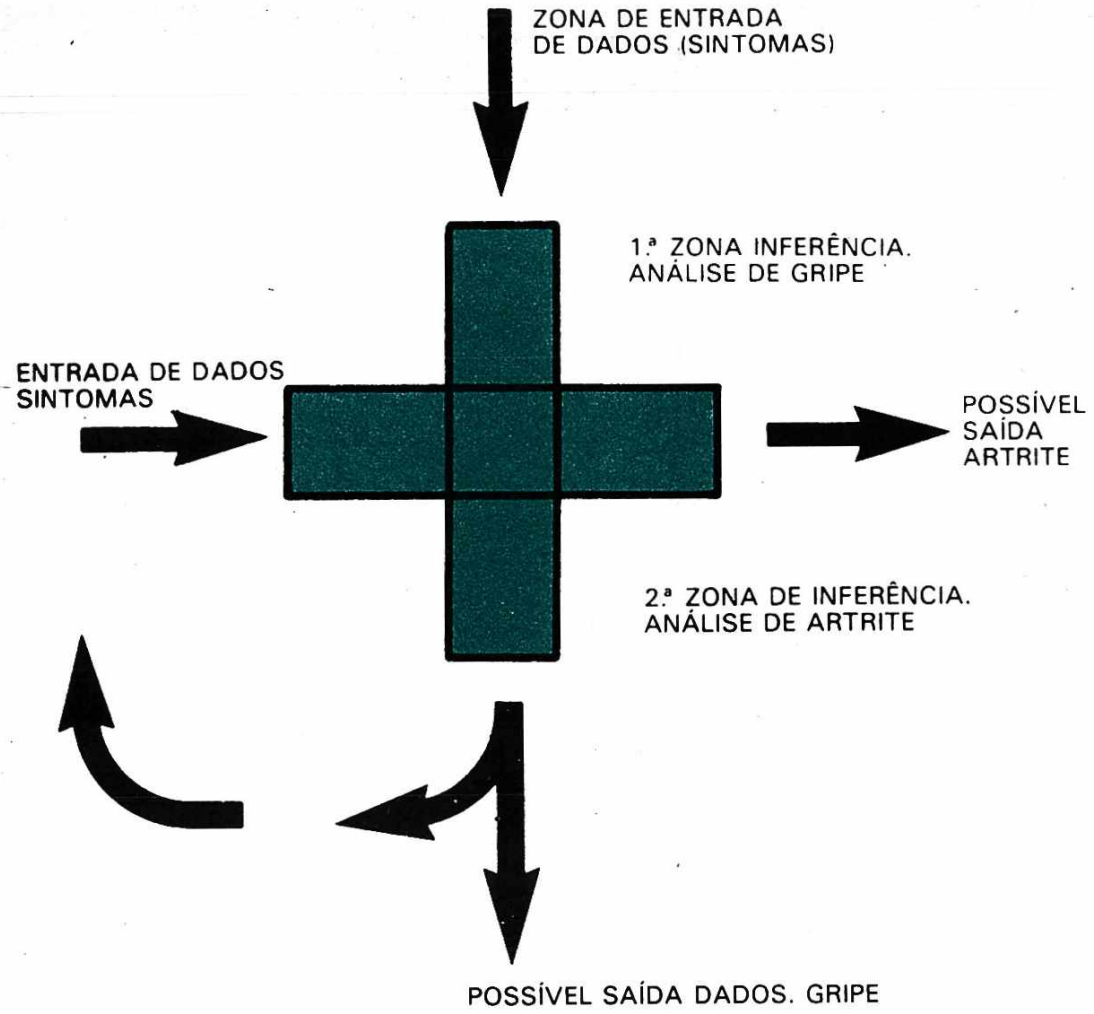


Figura 10. — O bloco de inferência corresponde a um equivalente de dois programas clássicos funcionando um depois do outro. Primeiro é decidida uma coisa e depois outra.

primeiro se o paciente tem gripe e em seguida analisar a possibilidade de que sofra de artrite (Fig. 10), ou ao contrário.

Assim seria possível continuar com todas as enfermidades que conhecemos. Isto dá a impressão de que qualquer programa realizado de forma clássica poderia conseguir os mesmos resultados finais que um com técnicas de inteligência artificial.

No entanto, a solução final conseguida mediante a programação clássica resultaria muito pouco compacta e seu funcionamento seria muito lento ao ter que recorrer de uma forma pré-definida to-

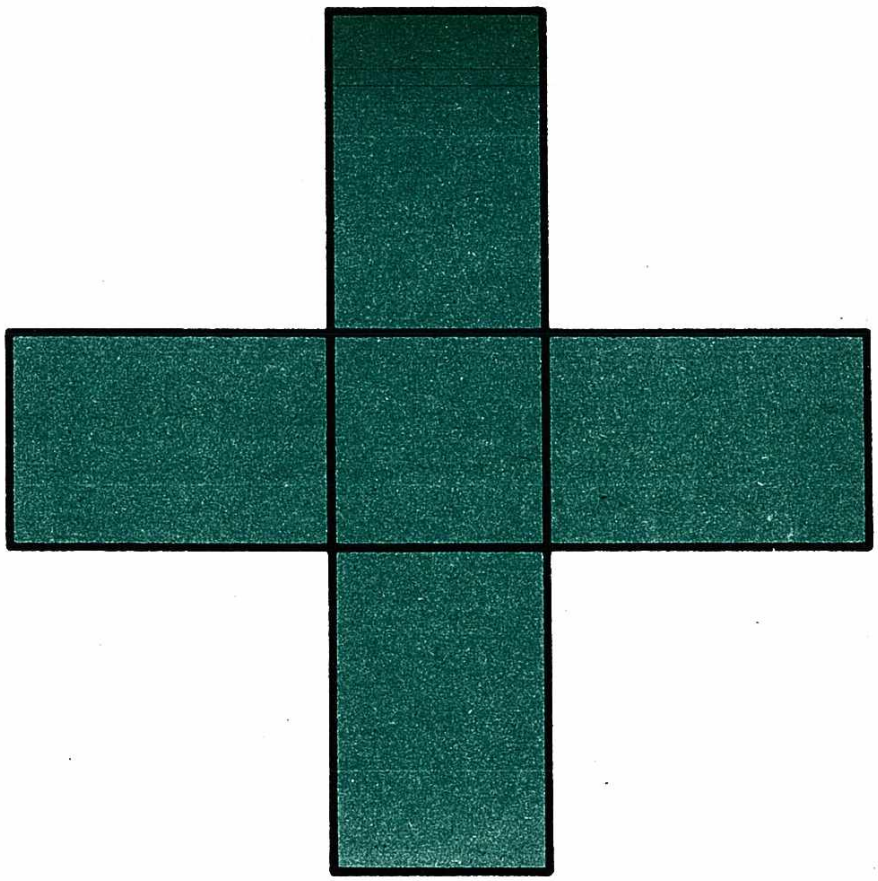


Figura 11. — A figura representa a zona de inferência utilizada e sua representação particular em um programa clássico com duas possibilidades.

do o espectro de enfermidades que o programa global tivesse armazenado.

Analisemos a forma que vai tomando o bloco global de inferência. Para o caso da dupla possibilidade, anteriormente analisada, se tem o bloco da figura 11.

No caso de que se tivesse um programa com mais enfermidades, ou melhor, com mais blocos de inferência, seria conseguida uma zona de inferência como a da figura 12.

Os sistemas expertos são muito mais eficazes que os programas clássicos porque podem ativar o bloco de inferência que é necessário em cada caso sem necessidade de ter que analisar todos os precedentes.

Isto é: se o paciente tem hepatite, o sistema experto analisará os dados de entrada que são introduzidos e deduzirá que tem hepatite sem deduzir previamente que não tem gripe e que não tem artrite. Neste ponto é estabelecida a grande vantagem dos sis-

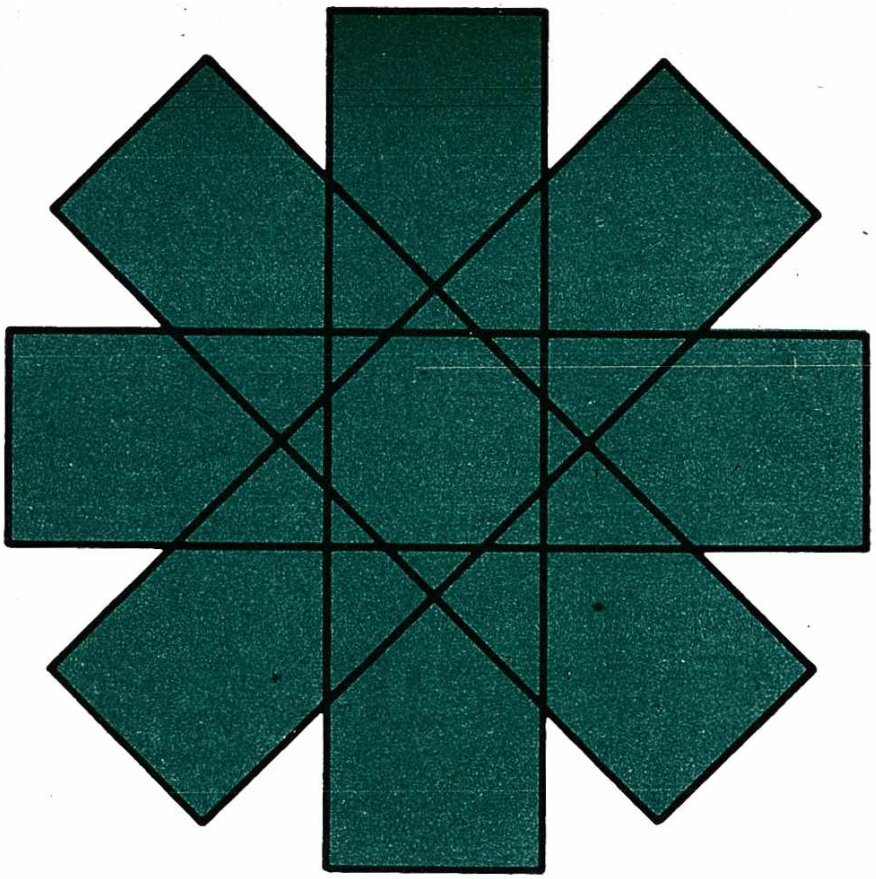


Figura 12. — Conforme aumentam as possibilidades em um programa tradicional, as zonas de inferência são multiplicadas.

temas expertos. De um banco de conhecimentos formado por muita informação somente é ativada a que corresponde a cada caso em função da entrada de dados. Isto é o que se conhece como estrutura sol (Fig. 13).

A partir da estrutura sol do sistema experto pode ser compreendido perfeitamente o papel fundamental do motor de inferência. Ante cada problema delineado pelo usuário e tendo em conta a estrutura autônoma do sistema, podem ser ativadas grande quantidade de blocos de inferência para resolvê-lo.

Estes blocos não têm por que ser os anteriormente definidos, mas que simplesmente são formados por regras independentes uma das outras, porém encadeadas para resolver o problema específico que é estabelecido em cada caso.

O banco de conhecimentos é apresentado como um labirinto (Fig. 14). Em função da porta pela qual se entre e em função das normas básicas que utilizemos para atravessá-la sairemos em um lugar ou em outro, obtendo resultados distintos em cada caso.

No entanto, entre todos os caminhos possíveis para resolver

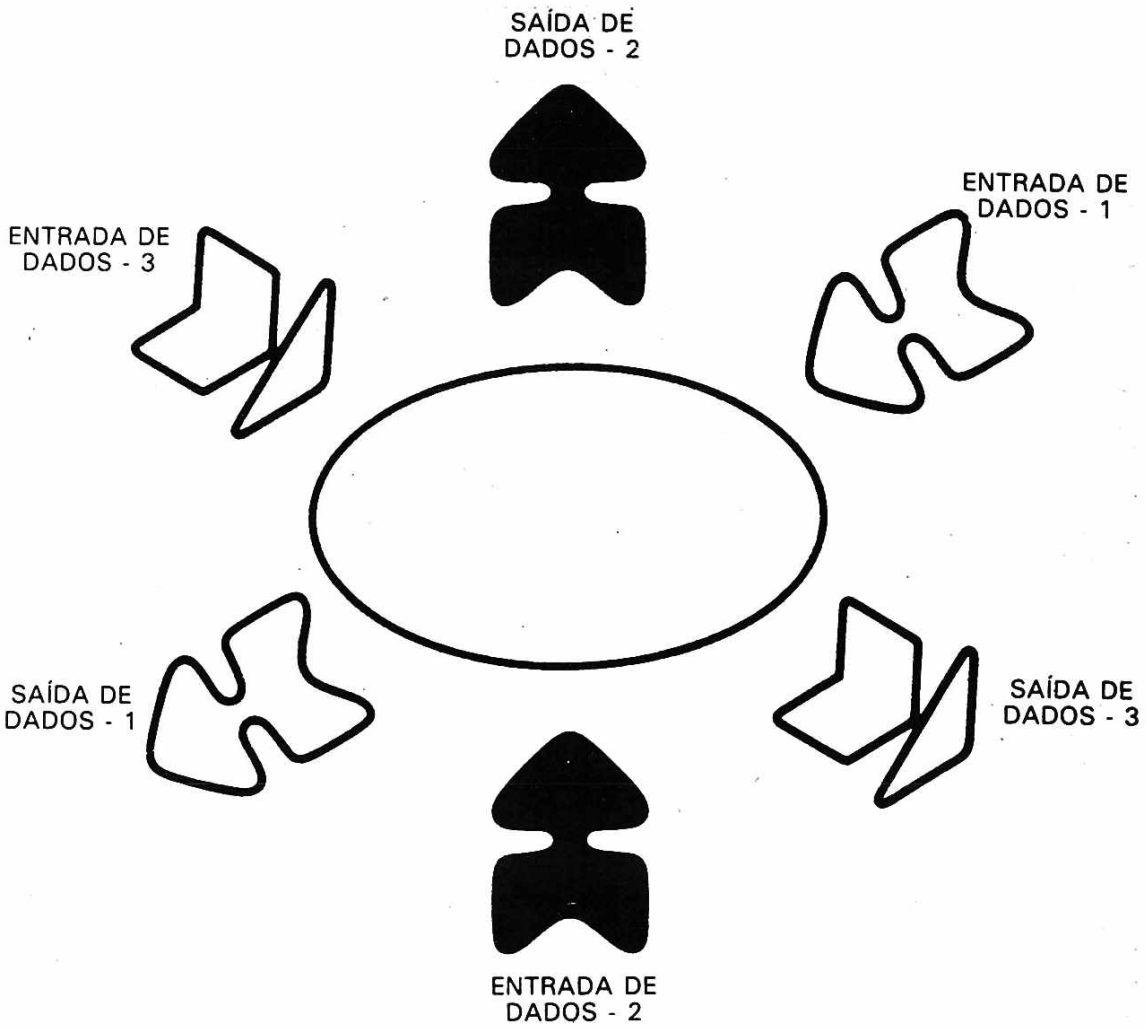
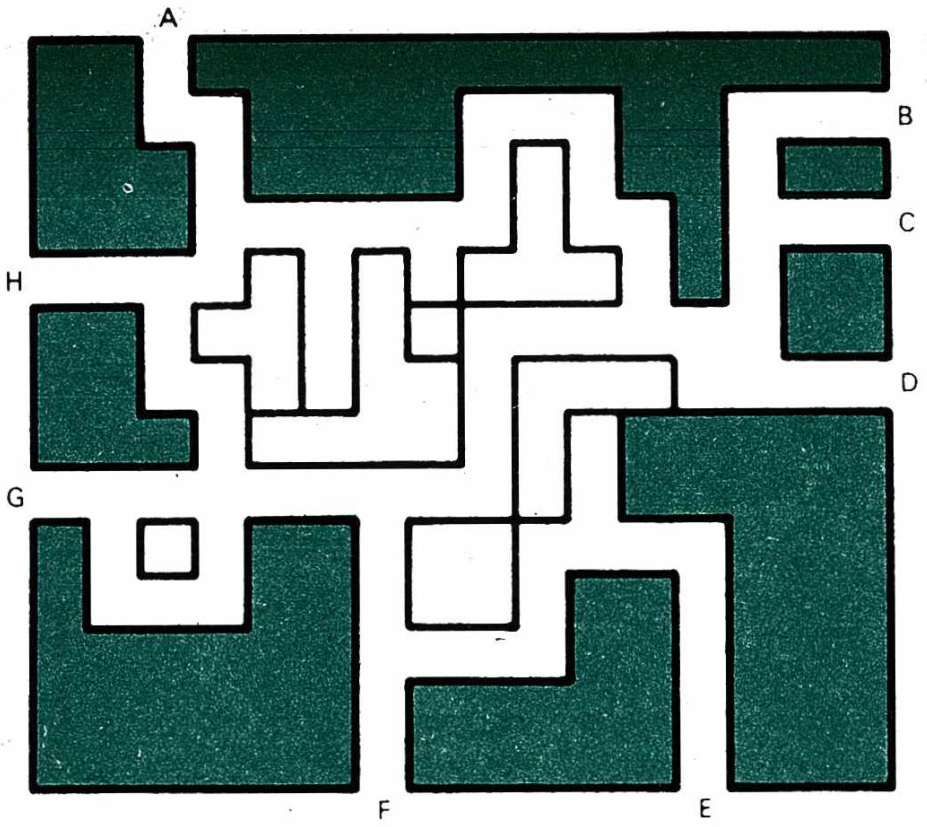


Figura 13. — A evolução do banco de conhecimentos visto como conjunto de blocos de inferência dá como resultado uma estrutura em sol do sistema experto.

um problema existe um que resultará o melhor. Para conseguí-lo teremos que utilizar as regras ótimas do banco de conhecimentos: o motor de inferência deverá escolhê-las. Esta tarefa é fundamental e é precisamente neste ponto do problema onde são delineadas soluções inovadoras.

Para escolher as regras ótimas podem ser aplicados muitos critérios: simplesmente mecânicos, isto é, sem que exista nenhuma "partícula de inteligência" ou então mais heurísticos, baseados na experiência. A palavra anteriormente citada, heurística, representa o eixo sobre o qual gira a inteligência artificial e será estudado mais profundamente posteriormente.

Existe na atualidade uma tendência a crer que todo proble-



A, B, C, D, E, F, G, H → Possíveis entradas
ou saídas do banco de conhecimento

Figura 14. — Uma qualidade importante do banco de conhecimentos é que possui muitas "entradas" e "saídas". Em cada caso a saída final dependerá da entrada escolhida e das regras usadas. Neste sentido é comparável a um labirinto.

ma informático *deve* ser resolvido mediante a criação de um sistema experto específico e isto é falso. É certo que muitos problemas poderiam ter uma solução desde o ângulo da inteligência artificial, porém esta pode chegar a ser em ocasiões muito menos eficiente que um programa clássico. Isto é assim porque existe um determinado tipo de problemas que pertencem ao ambiente da inteligência artificial e outro que não encaixa tão bem. Portanto, antes de realizar um projeto informático convém fazer um estudo da viabilidade e escolha das ferramentas adequadas para sua execução. Não custa muito e pode economizar muito tempo e dinheiro.

Particularidades do sistema experto

Devido a grande quantidade de possibilidades na hora de criar o "caminho de inferência" para a resolução de um problema pode acontecer (e de fato se espera que aconteça) que ante os pro-

blemas estabelecidos sejam conseguidas soluções não programadas.

Isto é precisamente o que se pretende, que o sistema possua "mobilidade", que seja algo mais que um simples banco de dados de soluções.

Graças à sua estrutura o sistema é capaz de explicar, passo a passo, em sua evolução desde as condições iniciais até a solução final e, além disso, descrever por que tomou cada decisão na hora das sucessivas opções dentro do "caminho da inferência".

Como o banco de conhecimentos é encontrado separado do resto do sistema este pode ser renovado, atualizado e melhorado em sua informação sem necessidade de variar sua estrutura operacional. Isto resulta muito importante, posto que pode ser utilizado inclusive para resolver tarefas completamente distintas (dentro da modalidade de problemas aptos para serem resolvidos com a inteligência artificial).

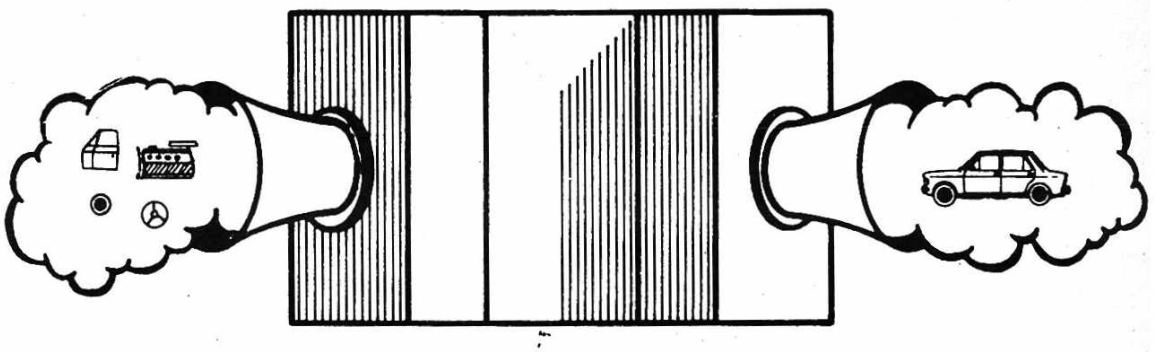


Figura 15. — O sistema experto gerativo "cria" soluções a partir dos dados de entrada.

Áreas de aplicação dos Sistemas Expertos

Temos falado de problemas que se prestam perfeitamente a sua resolução mediante técnicas de inteligência artificial, de motores de inferência que funcionam melhor com um tipo de banco de conhecimento que com outros e de um longo número de diferenças substanciais.

Isto nos faz pensar que existirão diversas zonas de aplicação para sistemas expertos em função do tipo de desenho a que tenham sido submetidos.

Cada uma destas zonas é coberta por um tipo de sistema experto que possui as condições ótimas para cobrí-la. São considerados dois tipos básicos de sistemas expertos:

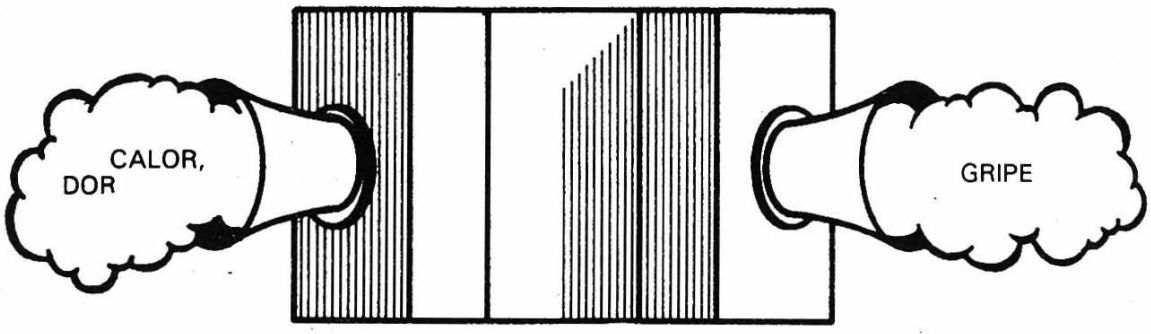


Figura 16. — O sistema experto interpretativo "interpreta" os dados de entrada até conseguir dar com uma solução convincente.

• SISTEMAS EXPERTOS GERATIVOS

Como seu nome indica realizam tarefas de tipo "criativo" (na concepção mais limitada da palavra). Se tenta introduzir-lhes os máximos níveis de inteligência para que dependam o menos possível das diretrizes fundamentais de cada caso introduzidas pelo "homem". A informação que utilizam está composta pela que é introduzida pelo usuário para cada problema e pela armazenada no seu próprio banco de conhecimentos.

Sua principal função é o desenho (por exemplo, industrial, de peças mecânicas, etc.), a indução de problemas matemáticos, etc.

• SISTEMAS EXPERTOS OPERACIONAIS OU INTERPRETATIVOS

Sua forma de trabalho é menos original que a dos sistemas gerativos. A informação que manejam provém basicamente da que o computador possui armazenada; o peso específico da informação que fornece o usuário é menor para avaliar a atuação do sistema. São sistemas de tipo consultivo.

Manejam problemas que não são apresentados nos outros sistemas, como são o manejo de grande quantidade de informação, a correta gestão desta e problemas de precisão e certeza na informação que o sistema apresenta como solução ao problema estabelecido.

Os principais campos de operação para este tipo de sistemas são a manutenção de máquinas, a consulta médica especializada, etc. Por exemplo, nosso sistema "Médico Eletrônico" formaria parte deste grupo, posto que o que realiza é um diagnóstico médico do paciente.

Vantagens dos sistemas expertos

A principal vantagem dos sistemas expertos é que, quando realmente servem para o que foram construídos, otimizam qualquer tarefa encarregada em relação a um programa convencional em um tempo semelhante.

Classificando estas vantagens teríamos:

AJUDAR

O sistema experto pode responder a perguntas que lhe são formuladas de uma forma direta quando possui as soluções armazenadas em sua memória. Isto é, pode servir como uma ajuda tipo agenda.

Sua memória é utilizada como a do usuário, posto que os dados que este pede são encontrados armazenados no computador.

Por exemplo, criemos um terceiro sistema experto, ao que chamaremos "Cuida-automóveis eletrônico". Será ocupado de manter nosso automóvel em perfeito estado para a condução mediante a informação de "como se encontra o automóvel em cada instante", olha os quilômetros, o óleo, a água, os pneus, etc., e deduz se ao automóvel temos que fazer algum reparo, trocar alguma peça ou funciona bem. Além disso, como nosso "cuida-automóveis eletrônico" é um sistema experto estupendo, podemos perguntar-lhe qualquer coisa sobre o funcionamento do veículo.

Uma das regras que teria o sistema seria:

REGRA I:
SE O NÍVEL DE ÓLEO DO AUTOMÓVEL É BAIXO $\xrightarrow{\text{então}}$ O MOTOR FUNDE

A capacidade de ajuda do sistema experto será evidente quando lhe perguntarmos:

O que acontece se o nível de óleo é baixo?

posto que responderá:

"O motor funde"

Não existe de fato nenhum tipo de inferência, posto que a informação é encontrada "tal qual" no computador.

INFERIR

Representa um passo mais evoluído. O sistema é capaz de responder a perguntas com soluções que não são encontradas diretamente armazenadas na memória do computador. Isto é conseguido mediante um processo de inferência, de modo que a partir das premissas iniciais são conseguidos fatos que solucionam o problema estabelecido.

Suponhamos que nosso sistema possui, além da regra anteriormente citada, outra regra que diz:

REGRA 2
SE O MOTOR FUNDE $\xrightarrow{\text{então}}$ O CARRO QUEBRA

Um usuário poderia perguntar:

O que aconteceria se o nível de óleo é muito baixo?

e o computador responderia:

"O carro quebraria"

Podemos ver como a informação resultante da utilização do sistema ("se o nível de óleo é muito baixo pode acontecer que o carro quebre") não era encontrada previamente armazenada no computador, mas que foi induzida em seu funcionamento normal.

JUSTIFICAR

O poder de justificação do sistema é entendido como a facilidade de expor todos os passos do raciocínio realizado pelo computador sobre o banco de conhecimentos incorporados e a modificação de fatos básicos para ter uma idéia exata do por que da solução escolhida.

Em nosso sistema, para continuar com o exemplo, ante a resposta que nos proporcionou o sistema:

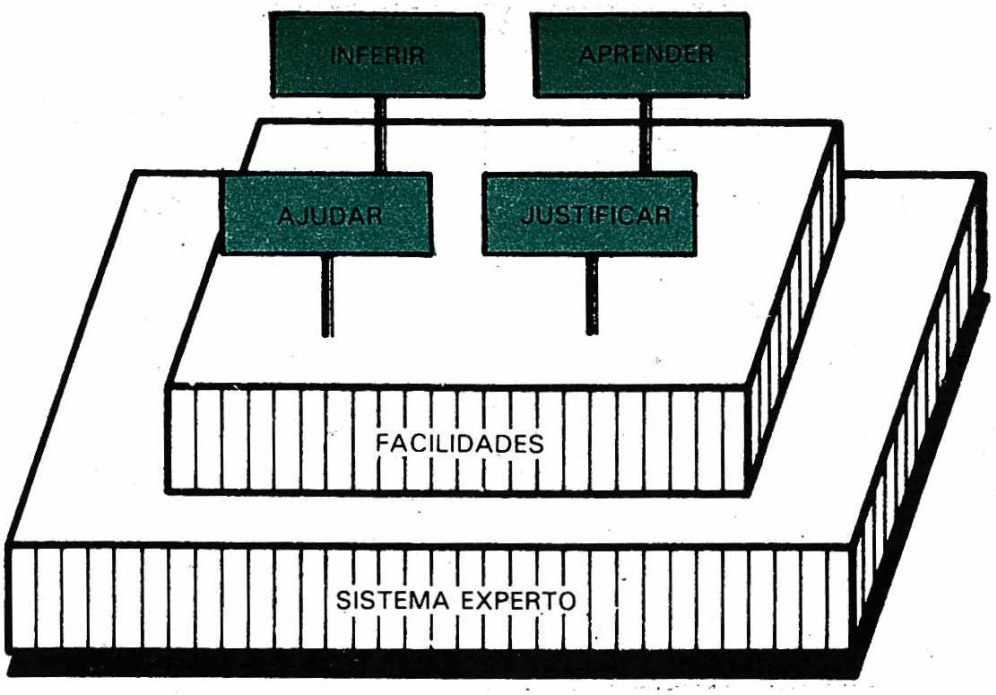


Figura 17. — As vantagens do sistema experto são as bandeiras a favor da Inteligência Artificial.

"Ante um nível de óleo muito baixo o carro quebraria"

poderíamos pedir que a justificasse

"Se o nível de óleo é baixo o motor fundiria e se o motor fundisse então o carro quebraria"

Com isto a conclusão estaria completamente justificada.

APRENDER

Este é um aspecto pouco explicado. As pessoas que desconhecem o tema sentem um certo temor (não falta de fundamento "no fundo") ante a possibilidade de que um sistema artificial seja capaz de aprender "algo". No entanto, a capacidade de aprendizagem destes sistemas expertos é certamente limitada e, além disso, muito artificial.

Um aspecto que não foi comentado todavia dos sistemas expertos é o da aprendizagem inicial que tem o "programa" antes de que seu funcionamento possa ser considerado como ótimo. Esta aprendizagem inicial, fundamentalmente realizada pelo engenheiro do conhecimento, consiste em ensinar ao sistema o necessário

para que este possa trabalhar com conhecimentos sobre o tema tratado. Na realidade o que se faz é encher o banco de conhecimentos com a informação básica sobre o tema. Com isto será evitado que o sistema experto se comporte como um verdadeiro "inútil" antes de ter aprendido o suficiente. Esta primeira aprendizagem representa, portanto, a tomada de nível.

No entanto, existe outra aprendizagem que serve para melhorar esse nível inicial e conseguir que o sistema evolua, o que tem lugar durante a fase de utilização: em qualquer inferência pode ser armazenada, com o que é ampliado seu conhecimento.

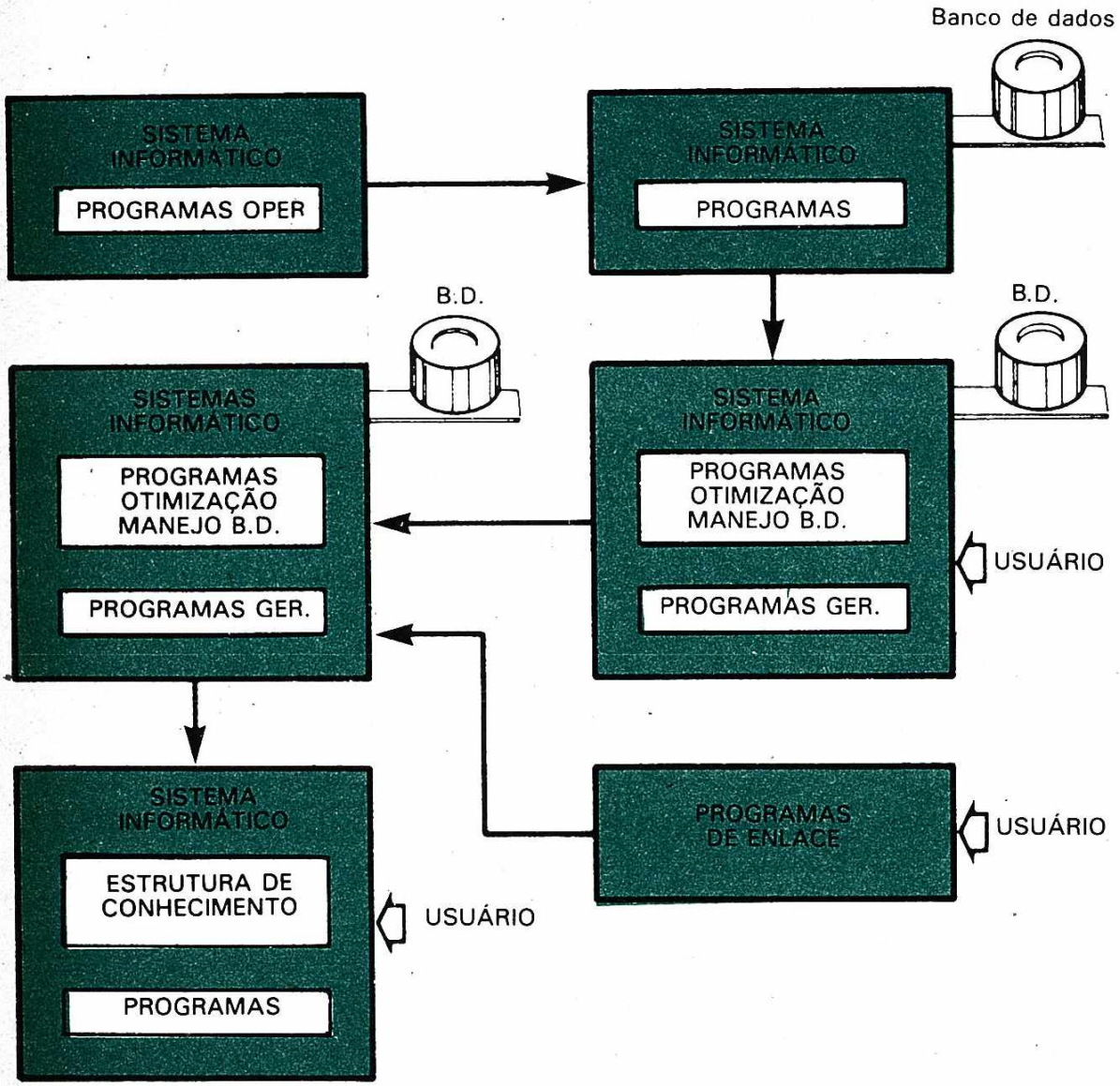


Figura 18. — A evolução final dos sistemas expertos de recuperação de informação permitirá um manejo ótimo do conhecimento.

Outra forma de aprendizagem, muito mais artificial todavia, consiste em que quando o sistema não sabe algo o pergunta ao usuário.

Seguindo com o exemplo de nosso "cuida-automóveis eletrônico" a inferência que foi conseguida:

"Se o nível do óleo está muito baixo então o carro quebra"

poderia ser armazenado como uma nova regra do banco de conhecimentos correspondente. O sistema teria aprendido.

Uma aplicação dos S.E.: A recuperação da informação

Uma aplicação muito importante dos sistemas expertos é encontrada em um campo que possui um enorme futuro como foco desenvolvimento: Se trata da recuperação da informação, campo que surgiu como continuação das tentativas que foram realizadas para aglutinar a informação de todo tipo em bancos de dados. Consiste no estudo dos métodos presentes e futuros cujo fim é otimizar o recolhimento de informação, armazenada em bancos de dados, com destino ao usuário final.

Atualmente existem muitos problemas, não somente com a pouca capacidade dos próprios bancos, mas com sua especificidade, sua forma de armazenar a informação e a maneira de aceder a ela desde o exterior. Justamente para tentar melhorar tudo isto está sendo aplicada a I.A.

Os sistemas expertos têm muita utilidade neste campo, posto que podem servir de intermediários entre os bancos de dados e o usuário final. Assim, por exemplo, hoje em dia é muito difícil consultar um banco de dados com uma linguagem "natural", ou melhor, uma linguagem normal (falada ou escrita), mas temos que fazê-lo mediante uma linguagem de comandos mais ou menos complicado.

Poderia ser construído um sistema experto que tivesse como finalidade o poder entender o que o usuário lhe pergunte (em seu próprio idioma) e em seguida traduzí-lo à linguagem de comandos que o banco de dados necessita. Com isto conseguir-se-ia facilitar as coisas na hora em que uma pessoa sem conhecimentos de informática pudesse utilizar o computador como se fosse uma simples máquina de escrever.

Outra tarefa deste sistema experto intermediário poderia consistir em executar o processo contrário, isto é, que ante algumas soluções (informação) enviadas pelo banco de dados ao usuário

em forma de comandos se conseguisse que o sistema os transformasse de forma que fossem possíveis de ler como um livro. Com isto seria evitado novamente o ter que conhecer aspectos internos dos bancos de dados para fazer uso deles.

Ante um futuro que cada vez se vê mais próximo, é estabelecida a dúvida de para onde irão evoluir os sistemas de recuperação de informação. Parece claro que os atuais bancos de dados desaparecerão, deixando passagem a sistemas expertos nos quais a informação seja armazenada no banco de conhecimentos. Com isto será dado um grande passo, não somente na otimização de busca que seria conseguida, mas porque a informação poderia ser muito mais rentabilizada. Em um banco de dados esta deve ser estruturada mediante procedimentos associativos, isto é, ou mediante relação de alguns fatos com outros, ou de forma hierárquica, etc. No entanto, o banco de conhecimentos permite armazenar "conhecimentos" independentes uns dos outros, o qual representa uma grande vantagem.

A evolução dos sistemas expertos de recuperação de informação seria correspondente com o gráfico da figura 18.

Lembrando que na figura está a evolução dos sistemas informáticos visto de menor a maior complexidade e não através do tempo, podemos observar como o fundamental do sistema é encontrado em sua capacidade interna de operação mediante alguns programas. Em um passo seguinte, é "presenteado" ao sistema um banco de dados para que seu trabalho não seja puramente interno, mas que possua conexão com o mundo exterior. O seguinte que se tenta é, por conseguinte, otimizar essa conexão para que o sistema tire o máximo rendimento.

Posteriormente, uma vez melhorada a comunicação com o mundo exterior, tenta-se conseguir otimizar a conexão do sistema com o usuário para que este não tenha que conhecer nenhum tipo de linguagem especial para comunicar-se com o computador.

Por fim o passo do futuro representa o momento em que todos os blocos anteriormente citados são integrados em uma só estrutura.

CAPÍTULO III

REPRESENTAÇÃO DO CONHECIMENTO

Um dos pontos fracos de toda a teoria de programação que estivemos vendo até agora fundamenta-se na forma de representar o conhecimento. Temos que buscar uma forma capaz de representá-lo o suficientemente bem para que resulte efetiva e aproveitável. Parece uma "tonteira", porém não o é; assim, suponhamos que queremos representar a seguinte afirmação:

"Hoje chove muito, porém ontem choveu mais do que choverá amanhã".

De imediato, até a uma pessoa seria difícil *quantificar* uma informação como a anterior, assim que se tentarmos introduzí-la em um computador os problemas serão ainda maiores. De entrada parece impossível que o computador seja capaz de saber quanto significa realmente a palavra "muito". O que fazemos indiretamente quando nos comentam "hoje chove muito" é recorrer a uma informação prévia que temos armazenada de antemão e que adquirimos com a experiência. Sabemos que quando chove muito o céu está muito fechado e a água que cai encharca o corpo muito rapidamente. Isto é, temos um conhecimento aproximado da água que cai quando chove muito; quando dizemos "chove muito" sabemos que cai mais de uma gota cada hora e, obviamente, menos que se nos caisse todo o mar Mediterrâneo em cima de uma vez.

Pois o mesmo deve ter um computador para ser capaz de ma-

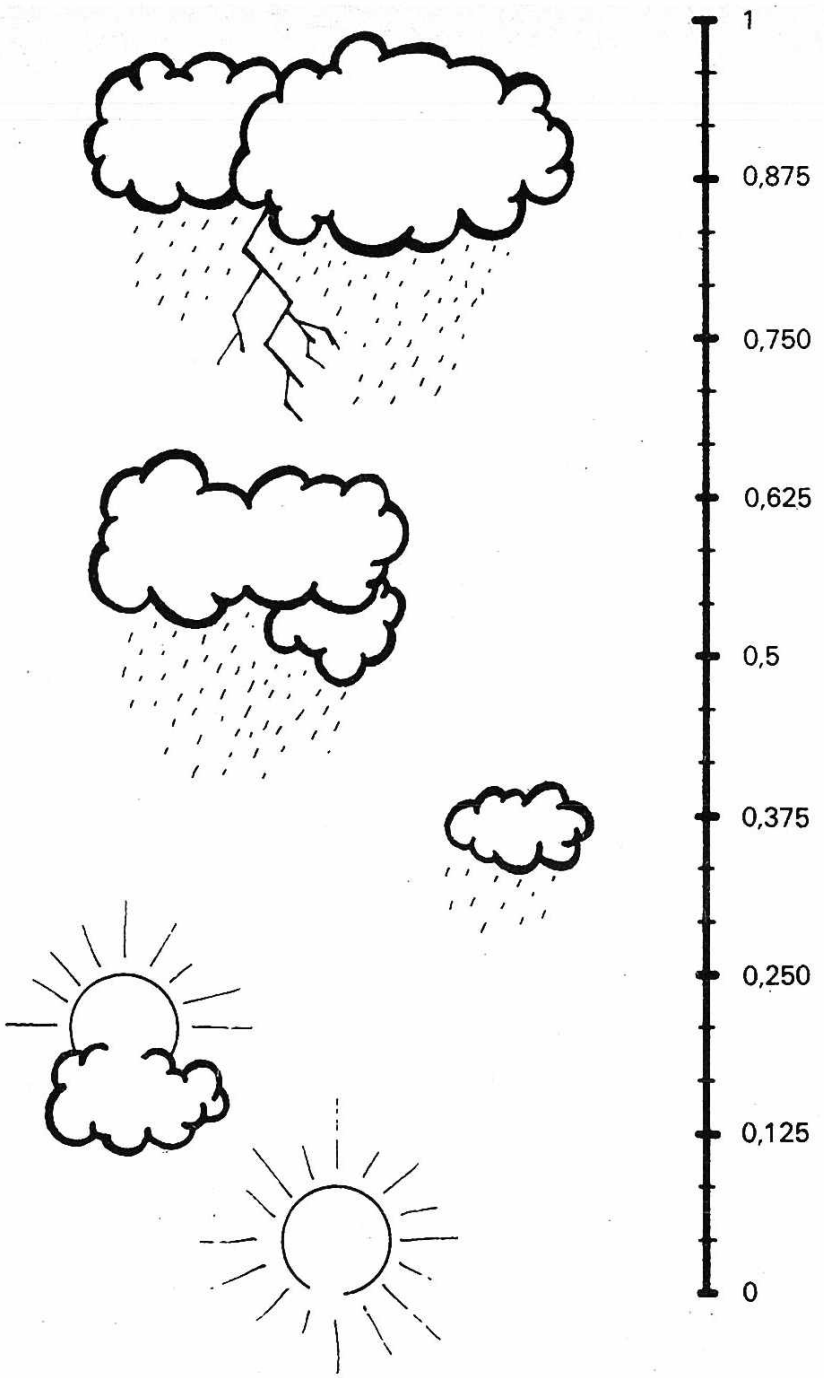


Figura 1. — O "fato" de chover pode ser quantificado matematicamente mediante uma escala graduada.

nejar este tipo de conceito. No entanto, é muito difícil de conseguir, posto que o homem vai aprendendo desde seu nascimento, de tal modo que faz parte de sua vida.

Por isso todos os métodos atuais de representação do conhecimento são aproximados e não encaixam exatamente com o que o homem deseja realmente transmitir.

Assim, para representar a frase anterior poderíamos introduzir no computador algo parecido ao seguinte.

Hoje chove 0.6 porém ontem choveu 0.5 e amanhã choverá 0.3.

Criamos uma tabela para a chuva onde o volume está compreendido em um intervalo (0-1). O 0 significa que não existe um sol explêndido e o 1 que está caindo um aguaceiro que produz inundações.

Geramos, pois, uma relação numérica entre a informação que queremos transmitir e sua quantia (Fig. 1).

Podemos comprovar como ao utilizar este tipo de representação do conhecimento, perdemos o significado inicial da frase, trocando-o por outro aproximado. Se tivéssemos utilizado outra forma de representação, o significado final da frase representada teria sido distinto.

Este problema e outros muitos são estabelecidos na hora de escolher que tipo de representação do conhecimento utilizar para criar o Sistema Experto.

Figura 2. — A ambigüidade é o pior inimigo da representação do conhecimento.

Antes de definir os distintos tipos de representação que são manejados é necessário nominar as condições comuns que devem cumprir todos eles para que sejam válidos.

É fundamental eliminar a possibilidade de que em algum caso possa ser representado qualquer tipo de ambigüidade, posto que é este o principal inconveniente para que uma representação seja considerada aceitável.

Parece claro: se queremos uma boa representação do conhecimento não pode ocorrer que estamos representando de uma mesma maneira vários feitos de informação distintos. Temos que evitá-lo a todo custo. Para isso vamos analisar os dois tipos básicos de ambigüidade que existem:

- **AMBIGÜIDADE REFERENCIAL**

Por exemplo

"ele veio"

pode ter muitos significados distintos em função do contexto em que se encontre

— primeiro significado:

"Nico ordenou a Luís que viesse e ele veio"

Neste caso sabemos, por toda a frase, que "ele veio" significa que Luis veio.

— segundo significado:

"Nico disse a Luis: vamos. No entanto, Luis foi embora e somente ele veio".

Neste caso é Nico o que vem. Vemos como a mesma frase pode significar coisas distintas em função do contexto no qual se encontra.

AMBIGÜIDADE PELO SENTIDO DAS PALAVRAS

Por exemplo:

"Marcos pegou uma bola"

"Marcos pegou um resfriado"

"Marcos pegou uma bebedeira"

Aqui podemos comprovar como é a palavra a que cria ambigüidade, posto que o significado da frase depende do que "pegou" Marcos. Esta ambigüidade é muito difícil de eliminar e talvez seja, precisamente, a que atrase em muitos anos o que os computadores sejam capazes de *entender* tudo o que o homem possa contar-lhe em sua própria linguagem, sem nenhum tipo de restrições.

Tipos de representação do conhecimento

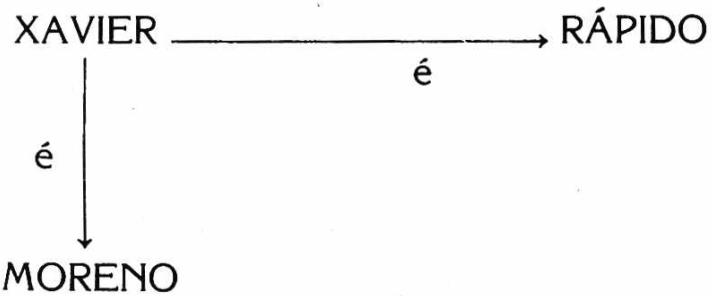
• REPRESENTAÇÃO DO TIPO REDES DE RELAÇÃO

A representação do tipo redes de relação é aquela que utiliza estruturas tipo rede (malha) para relacionar os conceitos entre si

Por exemplo para representar as frases:

"Xavier é rápido" e "Xavier é moreno"

seria feito:



pode ser visto como a forma final de representação tem uma estrutura tipo rede onde os nós são conceitos e os arcos são a relação entre os conceitos.

• REPRESENTAÇÃO DO TIPO LINEAR

A representação do tipo linear não utiliza redes de relação, mas que sua representação é estrutural.

Por exemplo, para representar as frases anteriores faria algo do tipo:

para "Xavier é rápido"
(inst Xavier rápido)
e para "Xavier é moreno"
(inst Xavier moreno)

Vemos como a representação utilizada tem uma forma linear em seu desenvolvimento (Atenção, linear de linha!!). Ainda que existam vários tipos distintos dentro da forma linear, todos eles têm em comum que sua estrutura não é de tipo árvore de relação, dai sua definição.

Aprofundaremos agora nos dois tipos de representação começando pela linear.

Representação em forma linear

A representação em forma linear utiliza cadeias de palavras com uma estrutura perfeitamente definida mediante algumas regras de formação determinadas. Cada cadeia recebe o nome de fórmula.

As palavras que formam a fórmula são os dois tipos distintos predicados e argumentos.

Existe um predicado para cada bloco mínimo de informação. O predicado une os argumentos para dar-lhe sentido em cada fórmula. Além disso, dá o sentido que deve ter a fórmula.

Por exemplo, a frase:

"Maria é alta"

seria representada,

(inst. Maria alta)

onde "inst" (que provém do inglês instance) é um predicado que une os argumentos Maria e alta. Sua função é definir que Maria é alta, isto é, relacionar "Maria" com "alta".

A vantagem desta estrutura é que uma fórmula já composta, como por exemplo (inst. Maria alta), pode fazer parte de outra fórmula mais complicada como argumento. Assim é conseguida uma grande flexibilidade e é permitida a construção de estruturas complicadas a partir de outras mais simples.

Em relação com toda esta definição, vamos analisar um sis-

tema muito utilizado para trabalhar com o computador que particulariza as idéias gerais que expusemos até agora: é o "cálculo de predicados". Possui suas próprias normas particulares, ainda que a base nos resultará conhecida.

Cálculo de predicados

É um método muito utilizado para a representação de proposições e para o posterior manejo que delas pode ser feito para realizar inferências.

Basicamente consiste em criar células básicas de informação (fórmulas), constituída cada uma delas por predicado e argumentos.

O exemplo anterior serve por sua vez como definitivo do cálculo de predicados:

(inst Maria alta)

inst → Predicado
Maria → Argumento
Alta → Argumento

Para evitar a ambigüidade ao definir os elementos são atribuídos coeficientes numéricos para conseguir que cada argumento tenha um único significado.

Por exemplo:

(inst Cadeira-1 Cadeira)	Cadeira-1 é uma cadeira
(inst Cadeira-2 Cadeira)	Cadeira-2 é uma cadeira

Cadeira-1 e Cadeira-2 são definidas como cadeiras, porém pode acontecer que uma seja clássica e a outra tipo tamborete de bar; precisamente para diferenciá-las são atribuídos coeficientes numéricos distintos.

• ARGUMENTOS

Existem três tipos distintos de argumentos que podem ser utilizados nas fórmulas:

— Simbolos constantes
— Variáveis
— Funções de aplicação

Os símbolos constantes podem ser palavras que tenha que relacionar.

Por exemplo, 'João', 'Homem' ou qualquer outro.

As variáveis têm o mesmo papel que em matemática, isto é, podem tomar qualquer valor dentro do intervalo de definição que possuam.

Por exemplo, se temos:

(inst X Homem)

o que se consegue é que a variável "X" possa ter o valor "homem".

As funções de aplicação são, como já comentamos anteriormente, as fórmulas complexas que servem de argumento em outras.

Por exemplo, uma fórmula complexa que pode ser utilizada como argumento poderia ser:

(inst Pedro coxo)

- **PREDICADOS**

Como o cálculo de predicados não é uma linguagem na qual as palavras "chave" estejam definidas previamente, e não temos que utilizá-las forçosamente, é permitida uma grande riqueza na hora da representação.

Isto significa que podemos utilizar como predicados os que nos interessem para representar o que queremos.

Por exemplo, podemos ter:

(filho-de Marta Maria)

queremos representar que Marta é filha de Maria e utilizamos o predicado "filho-de".

Na realidade o cálculo de predicados serve de suporte para programar nas linguagens que maneja o computador (posto que todavia estamos no século XX). Estas linguagens, das quais a mais comum em I.A. é o LISP, estão estruturadas para trabalhar com informação alfanumérica, e o cálculo de predicados é amoldado muito a eles.

Como subclasse do papel que realizam os predicados, temos os elementos conectores e relacionadores de fórmulas.

Existem dois basicamente:

CONECTORES (if, and, or, not)

têm a faculdade de permitir conhecer a certeza ou falsidade da fórmula que formam conhecendo a do argumento sobre o qual atuam.

Por exemplo.

(not (inst carro vermelho))

O significado dessa representação é "o carro não é vermelho".

Se analisamos o valor podemos ver se o carro que queremos representar *realmente* é vermelho, em cujo caso

(inst carro vermelho) é certo,

e sabendo isto sabemos também que

(not (inst carro vermelho)) é falso, posto que o carro é realmente vermelho.

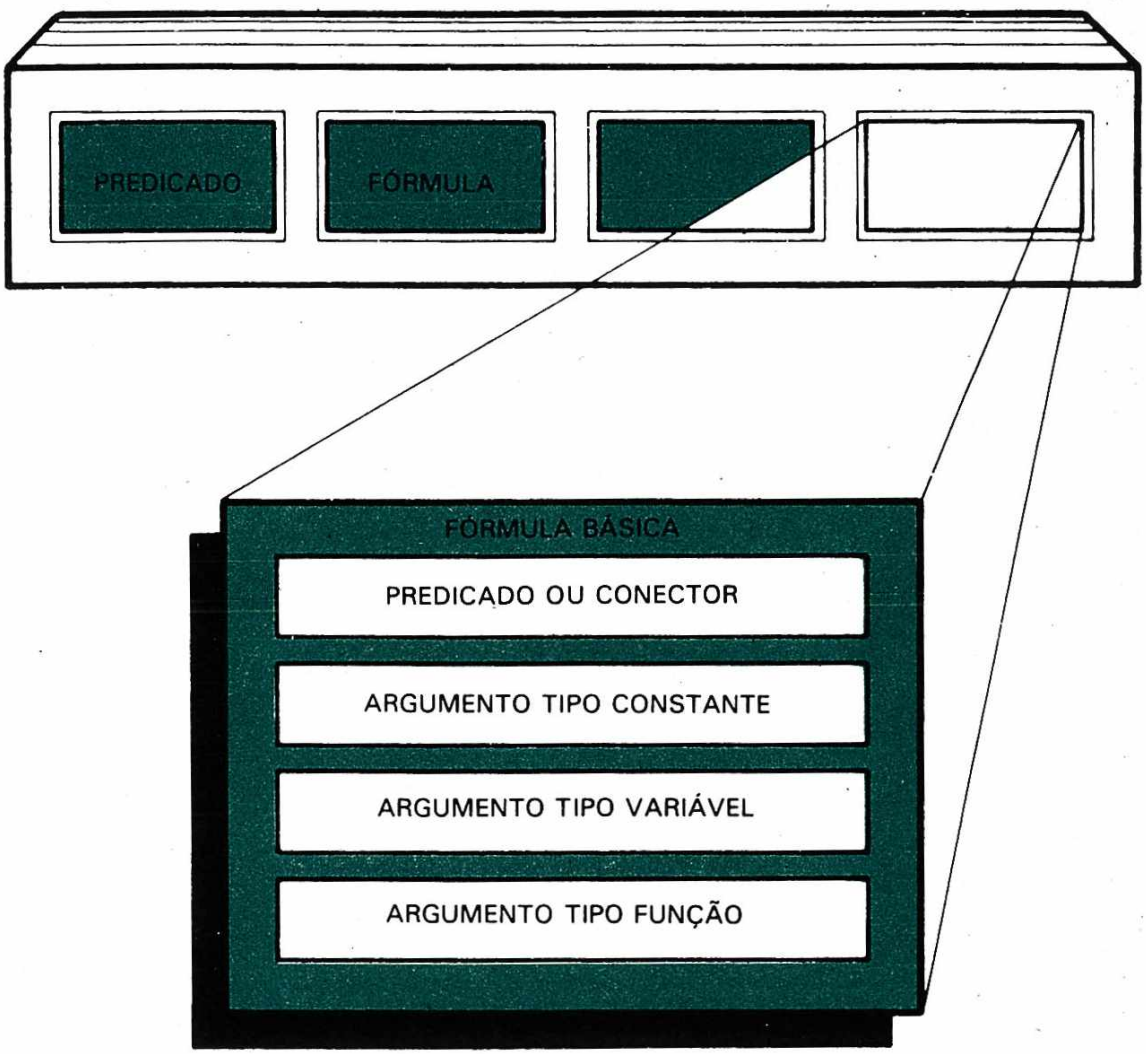


Figura 3. — A figura esquematiza a estrutura de toda informação representada mediante cálculo de predicados.

Existem dois

— Existencial, simbolizado como ∃ (exists)

tem no cálculo de predicados um sentido similar ao que possuia em matemática (existe...)

— Universal, simbolizado como V (forall)

é representado da seguinte forma:

(forall (X)A)

e significa que a variável X está quantificada de forma universal na fórmula A. Por exemplo: (forall(homens)nasceram) indica que todos os homens nasceram.

Regras do conhecimento

Quando estudamos a hierarquia que existia dentro dos níveis de informação comentamos que o conhecimento era conseguido precisamente mediante as "regras do conhecimento". Pois bem, estas regras devem poder ser representadas no cálculo de predicados posto que, em caso contrário, não serviria de nada esta teoria. E, de fato, podem ser representados.

Como existem muitas formas de fazê-lo nos centralizamos aqui nas regras de produção. Sua estrutura é a típica de qualquer regra:

IF (antecedente) THEN (conseqüente)

onde IF e THEN são os conectores ingleses que corresponderiam a:

SE (antecedente) ENTÃO (conseqüente)

Por exemplo:

"se chove então me molho"

A forma de representá-la mediante o cálculo de predicados teria o formato seguinte:

(if P Q)

o significado desta expressão seria: 'se é certa a proposição P então é certa a proposição Q'.

Já vimos as regras de produção. Agora a única coisa que falta é saber como será realizada a inferência e o que será conseguido induzir a partir de algumas regras do conhecimento iniciais.

Isto é conseguido definindo para cada sistema as regras de controle que o governam, também conhecidas como regras de inferência. Trabalham com os fatos básicos e as regras de produção para poder deduzir, induzir, etc.

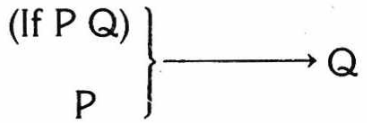
Vamos analisar algumas das que existem em matemática lógica.

• MODUS PONENS (DEDUÇÃO)

A regra de inferência diz assim: dados como informação certa
(if P Q)
e
P

se consegue ter a certeza de Q.

O podemos representar como



Por exemplo: se temos como informação inicial
(se chove então me molho)

e

(hoje chove)

podemos ter a segurança de que "hoje me molho".

Existe uma regra que pode ser concluída da "modus ponens": É a "modus tollens".

Diz assim:

dados: (if P Q)
e
Q

é obtido P. A barra situada sobre o P e o Q significam NOT (NOT

P e NOT Q).

• **ABDUÇÃO**

É uma regra de inferência muito boa para gerar o processo de explicação que é pedido ao sistema experto.

Funciona assim:

Dados (if A B)
e
B

É tirado A.

Tem associado um problema fundamental, e é que pode levar a conclusões falsas. Por exemplo de:

"Nico tem dores"
e
"Se sofre de câncer então tem dores"

se conclui que "Nico tem câncer", o qual é como mínimo muito aventurado.

No entanto, tem um grande valor por seu avançado nível de risco.

• **INDUÇÃO**

Esta regra de inferência é muito boa nos processos de aprendizagem do sistema experto. Seu significado é o mesmo que possui na matemática clássica, isto é, é baseado nos processos de generalização matemática a partir de uns certos elementos individuais. Por exemplo, se temos a certeza de:

"João respira"
e
"Maria respira"

Podemos tentar generalizar que
"Todos os homens respiram"

É também uma regra arriscada, posto que facilmente incide em erros importantes ao generalizar características que pertencem somente a elementos isolados.

Poderia ser esquematizado da seguinte forma:
se definimos

(p A)

(cujo significado seria que 'A' cumpre a propriedade 'p'), ao ter a certeza de (p A) e (p B) podemos inferir:

(forall (X) (p X))

que significa que para todo X é cumprida a propriedade 'p'. Além destas regras de inferência existem outras que também são utilizadas nos sistemas expertos. Com todas elas pode ser conseguido um bom motor inferencial; em função das que são escolhidas para criá-lo, o motor terá umas e outras características.

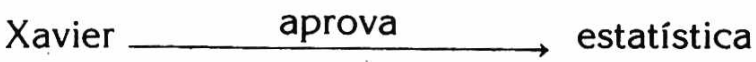
Representação tipo redes de relação

É a segunda grande tendência para a representação do conhecimento.

São utilizadas as estruturas em rede como método de relacionar conceitos e informação.

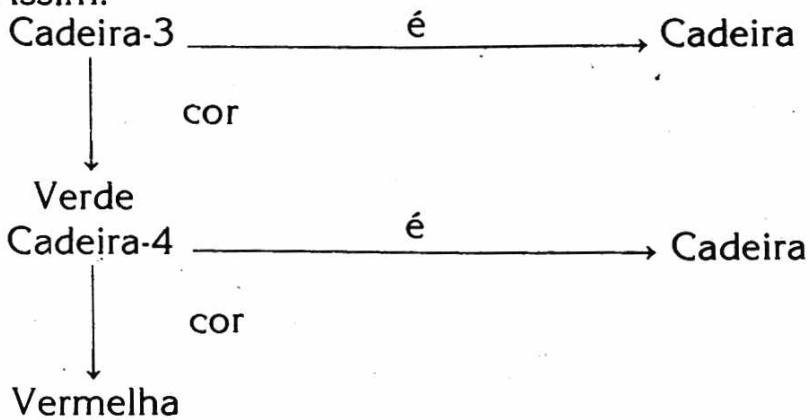
A rede é composta de nós e arcos. Os nós representam conceitos e os arcos são os laços da união entre eles:

Por exemplo:



Nestes tipos de representação também temos que evitar a ambigüidade. Para isso são utilizados coeficientes numéricos diferenciadores.

Assim:



Podemos observar duas coisas nos exemplos anteriores: que é necessário que existam coeficientes anti-ambigüidade (neste caso diferenciam duas cadeiras distintas) e que cada nó pode ser ponto de encontro de várias áreas; com isto se consegue criar uma estrutura tão complexa como se deseja e permite representar con-

ceitos complicados.

Agora teríamos que analisar todo o conjunto de ferramentas que, atuando sobre este tipo de representação do conhecimento, facilitam a criação de alguma estrutura que representa as famosas *regras de produção* da representação linear. Da mesma forma também é necessário neste caso controlar os processos de inferência que tenham lugar no sistema.

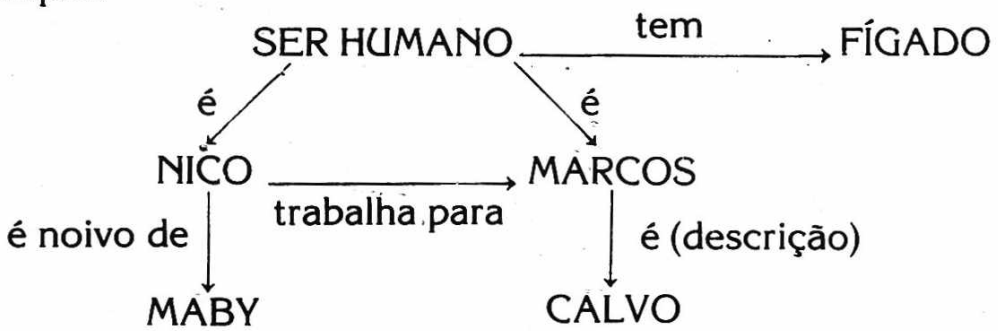
Para realizar todo este controle existem algumas técnicas de gestão de grafos e algumas propriedades (também de grafos) que permitem um manejo muito adequado da base de conhecimentos. Não as exporemos aqui em profundidade, apenas nos limitaremos a definí-las.

Algumas destas ferramentas para a gestão da rede são:

- **HERANÇA POR HIERARQUIAS**

As propriedades que possui um conceito as adquirem por "herança" todos os "filhos" de dito conceito.

Por exemplo:

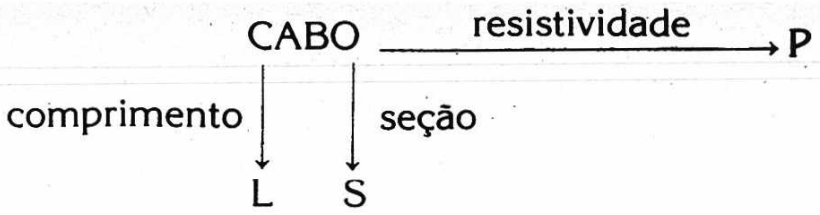


Todo ser humano tem fígado. Como Marcos é um ser humano, também tem fígado e não é necessário pendurar um arco diretamente do nó MARCOS ao nó FÍGADO. O mesmo acontece com NICO.

- **DEMONS**

É utilizado como procedimento para encontrar valores ou resultados partindo de informação já existente. Suponhamos que queremos encontrar a resistência elétrica de um cabo de alta tensão. Podemos fazê-lo se conhecemos o tipo de material que forma o cabo, ua resistividade, o comprimento do cabo e sua seção. Se temos tudo isso representado em uma rede poderemos calcular outra propriedade do cabo como é a resistência elétrica.

Assim teríamos:

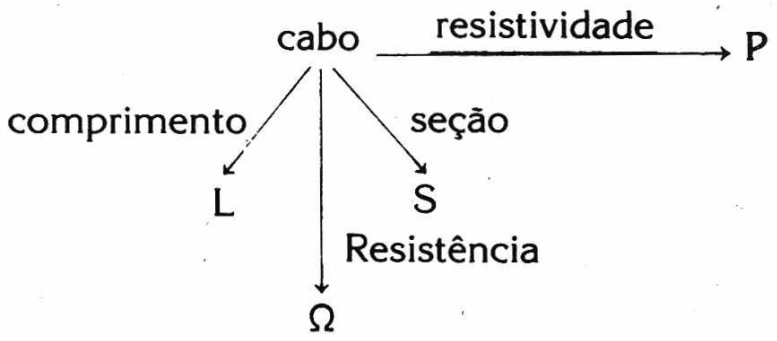


O processo a seguir deveria ser:

1) Ver se existem sobre o nó cabo as relações resistividade, seção e comprimento.
2) Em caso de que existam, criar uma nova relação onde o valor que lhe corresponde é conseguido mediante a multiplicação e divisão dos valores que possuam as anteriores, da forma seguinte:

$$\text{Resistência} = \text{Resistividade} \times \frac{\text{comprimento}}{\text{seção}}$$

O grafo ficaria da seguinte forma:



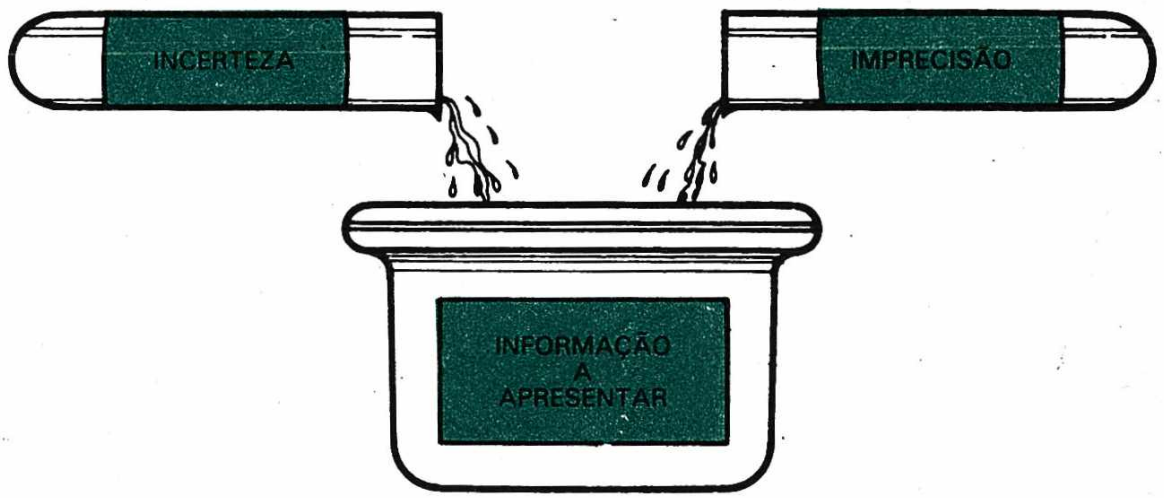


Figura 4. — Toda informação possui doses de imprecisão e incerteza que dificultam, e podem chegar a tornar impossível, sua utilização eficaz.

• **DEFAULT**

É utilizado para permitir, em certa medida, o trabalho da rede com um conceito novo: a probabilidade.

Até agora falamos do armazenamento e gestão do conhecimento sem analisar a possibilidade de que este não fosse de todo certo ou então fosse inexato.

Ainda que os problemas de certeza e precisão do conhecimento são muito importantes, aqui simplesmente faremos referência a eles e não os trataremos com profundidade. No caso de que o leitor pretenda aprofundar mais nesta área encontrará uma bibliografia ao final do livro onde encontrará informação a respeito.

O Default (palavra que provém do inglês e significa "por defeito") é utilizado para facilitar o manejo de informação que não possui um valor de veracidade absoluto.

Por exemplo:

João $\xrightarrow{\text{cor-cabelo(default)}}$ ruivo

quer dizer que o cabelo de João provavelmente seja ruivo.

Existem outros problemas à parte da probabilidade de verdade em uma informação; já citamos a certeza e a precisão. Devido a estes problemas a representação do conhecimento não pode evoluir tão rápido como em um princípio era previsto.

CAPÍTULO IV

ESTUDO DO MOTOR DE INFERÊNCIA

Já vimos a problemática básica da representação do conhecimento, por que se necessita um tipo determinado de representação e os múltiplos problemas que aparecem ao estudar o tema. Com relação ao motor de inferência a única coisa que comentamos é que sua principal missão era selecionar regras de produção que devem ser utilizadas para resolver cada problema proposto pelo usuário e como o motor de inferência realizava suas induções mediante algumas regras de controle determinadas.

Agora vamos analisar um pouco mais profundamente como funciona um motor de inferência "por dentro".

O primeiro será estudar o funcionamento do motor quando deve atuar, em suas duas fases: avaliação e execução.

Uma vez que é aprofundado o estudo do motor serão explicados os passos necessários para que seu funcionamento seja correto e ajustado ao que é desejado, que concretamente, são: a inferência, o conceito de busca, o controle da operação e o controle do espaço de busca.

Fases de operação do motor

Para resolver um problema mediante um sistema experto é necessário percorrer todo um caminho de inferência até chegar a uma solução final; ao longo do mesmo é necessário que o motor escolha muitas regras de conhecimento que estão armazenadas em seu banco de conhecimentos.

Cada vez que devemos escolher uma dessas regras deve ser colocada em funcionamento toda uma complexa estrutura que permita escolher a melhor, aplicá-la, etc. Isto o deve fazer o motor de inferência um grande número de vezes.

Para permitir otimizar este processo o motor está desenhado para que funcione por ciclos. Primeiro realiza uma tarefa e em seguida outra.

As principais fases do motor de inferência são:

— Avaliação
— Execução

• AVALIAÇÃO

Como seu nome indica, nesta fase o motor se ocupa de escolher as regras que deve aplicar para cada problema dentre todas as regras que existem em seu banco de conhecimentos.

Isto, que parece tão fácil a primeira vista, é muito complicado, com problemas de difícil solução, como, por exemplo, os conflitos entre regras, que analisaremos posteriormente.

A primeira etapa desta fase, como pode ser vista na figura 1, consiste na restrição. Se trata de recolher todas as regras que, por sua estrutura, possam ser aplicadas a um determinado programa estabelecido. Dentre todas as regras (R) as escolhidas formarão um subconjunto R_1.

Posteriormente temos que realizar uma filtragem das regras selecionadas escolhendo as que encaixam perfeitamente com o problema estabelecido. É conseguido o subconjunto R_2.

Uma vez conseguido o conjunto de regra R_2 temos que eliminar desse subconjunto as regras que provoquem conflitos. Este passo é importante. Suponhamos que estamos analisando nosso carro com o "maravilhoso" "cuida-automóveis eletrônico" e suponhamos também que em R_2 temos duas regras que dizem:

R_1 — "Se o carro faz ruído ao acelerar então temos que parar imediatamente".
R_2 — "Se o carro faz ruído ao acelerar então temos todavia que acelerar mais".

Podemos imaginar que o computador "formaria uma confusão" se tivesse estas duas regras para aplicar ao problema proposto. Por isso é necessário resolver todos os conflitos que apareçam em R_2.

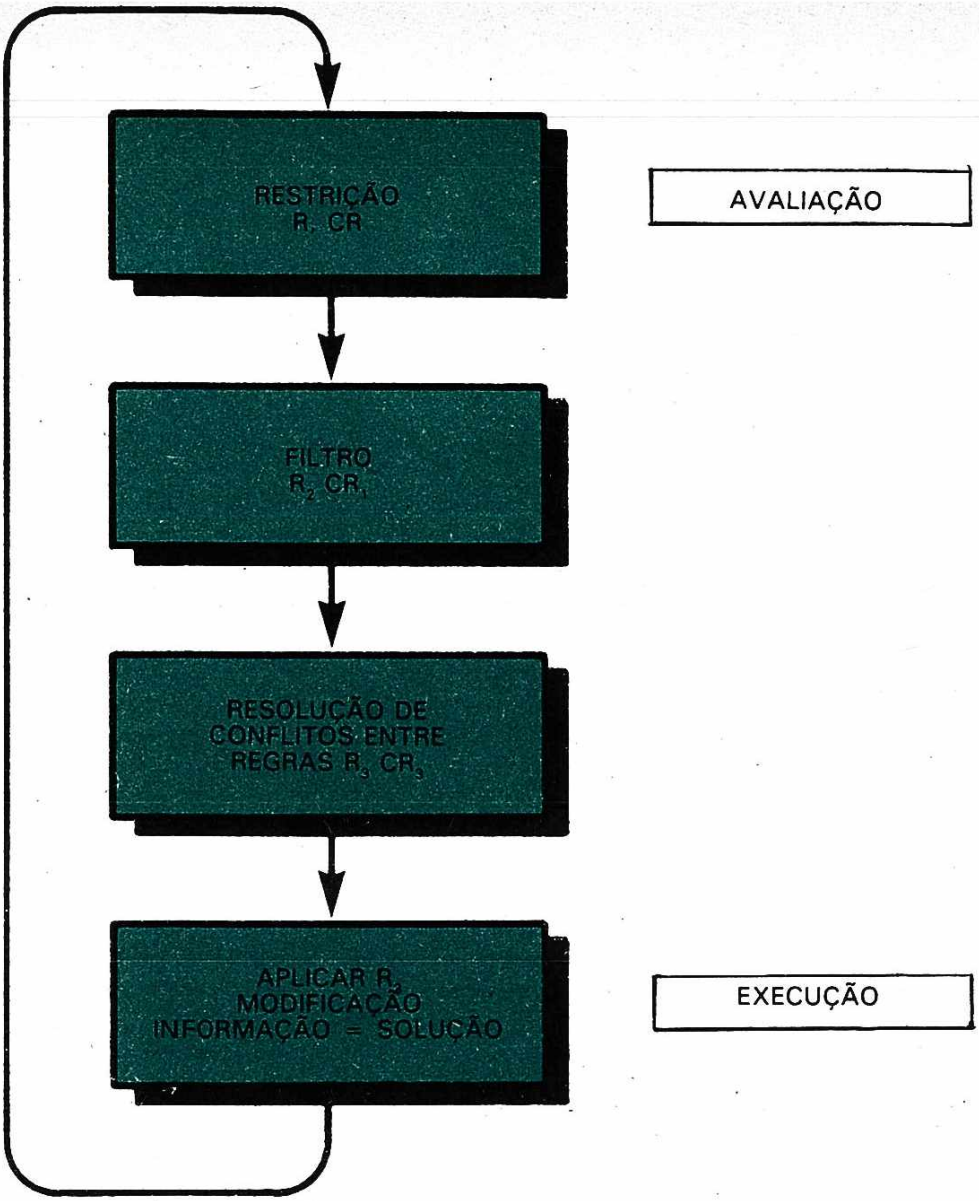


Figura 1. — O motor de inferência funciona mediante ciclos, com algumas fases perfeitamente definidas que vão estreitando o conjunto de regras adequadas para o problema.

As duas últimas etapas são as que estão menos desenvolvidas atualmente e é aqui onde mais fraquejam os sistemas expertos.

Os loops de inferência são laços fechados de indução: suponhamos que temos 4 regras A, B, C, D e que ao ativar "A", esta nos manda a "B", "B" a "C", "C" a "D" e, por fim "D" a "A". Teríamos o assinalado na figura 2.

Isto obrigaria ao computador a inferir sempre o mesmo, den-

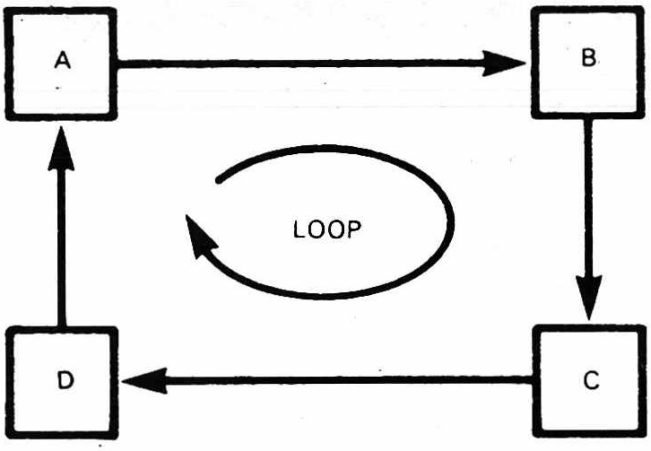


Figura 2. — Os loops de inferência são laços fechados de inferência nos sistemas expertos. Temos que evitá-los a todo custo.

tro de um loop sem fim. Este problema também deve ser detectado e eliminado para que o sistema funcione corretamente. Para evitar os "loops de inferência" é necessário uma ferramenta muito potente.

- **EXECUÇÃO**

Esta fase se ocupa de que as regras escolhidas na fase anterior (costuma ser uma só regra) sejam executadas, produzindo uma mudança na informação que será apresentada como solução do problema proposto.

Uma vez realizada a execução o sistema comprovará se existe uma condição de final e, em caso contrário, passará a uma nova fase de avaliação.

Funcionamento do motor de inferência

Já sabemos quais são as fases que leva a cabo o motor. Agora vamos analisar como é feito realmente tudo o que explicamos anteriormente.

Existem duas tarefas que deve realizar o motor.

- Inferência
  Deve ser aplicada a inferência ao executar uma das regras. Estas são executadas em função do tipo de inferência que possua o sistema experto.

- Controle
  O controle é ocupado de toda a gestão do sistema que analisamos anteriormente; tanto da forma em que são escolhidas as regras como da resolução de conflitos ou da eliminação de loops de inferência.

É importante o papel que joga o usuário nesta fase na maioria dos sistemas, pois muitas vezes o computador não conhece alguns dados importantes para gestionar o sistema e então os pedirá ao usuário. Desta forma pode ser suprida a falta de informação. Dentro da fase do controle temos que distinguir dois tipos:

— *Controle do espaço de busca*

É ocupada de gestionar "como são escolhidas as regras" em função do espaço de busca concreto (conjunto de regras que podem ser aplicadas dependendo de cada problema).

— *Controle da operação*

É ocupado de resolver conflitos e realizar todas as tarefas para gestionar a fase de avaliação.

- **INFERÊNCIA**

Para realizar as inferências temos que ter armazenadas umas determinadas regras de controle que ensinem ao sistema como levá-las a cabo.

As regras de controle (chamadas também regras de inferência) já foram estudadas; correspondem ao Modus Ponens, Abdução, etc. Aqui pode ser comprovado onde são aplicadas as regras que foram analisadas anteriormente.

- **CONTROLE**

Já definimos várias vezes as tarefas de controle de um sistema experto para que funcione bem. Este deve enfrentar-se com dois problemas básicos, consistentes e dispor de métodos para:

- decidir onde começar a analisar as regras que são armazenadas no banco de conhecimentos;
- resolver os conflitos entre regras e os loops de indução.

Além de resolver estes problemas, é necessário que o sistema trabalhe otimamente gestionando o espaço de busca de uma forma adequada.

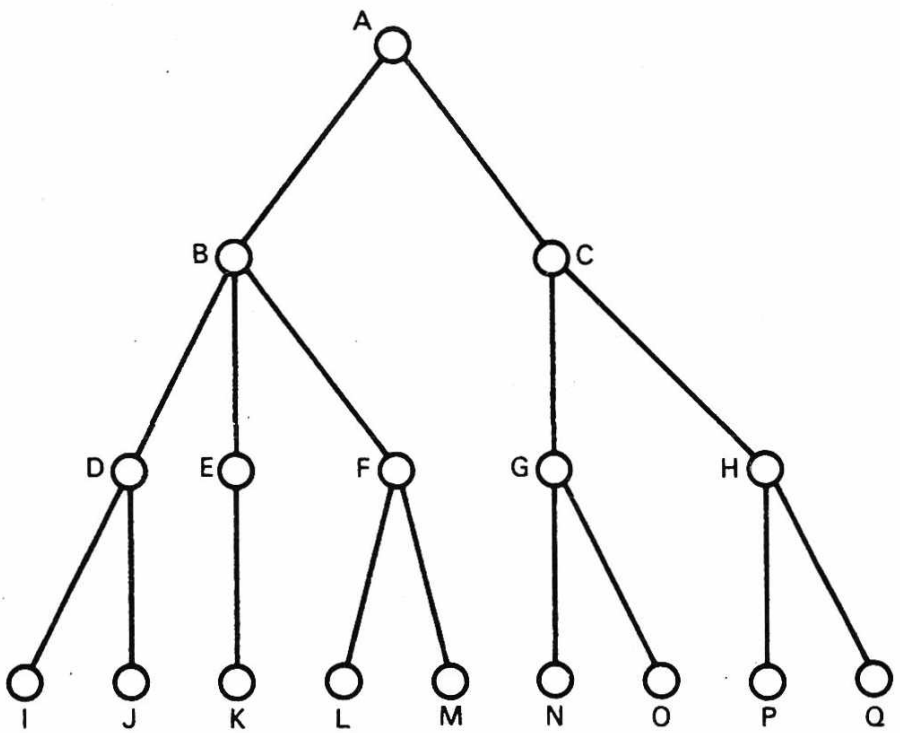
Quando em etapas anteriores definíamos o motor de inferência como o elemento do sistema que devia escolher as regras que tinham que ser aplicadas em cada momento, não foi comentado nada de como se conseguia esta escolha; o processo é realmente complicado.

Como sabemos, o computador não pode inventar nada. Tudo o que tem deve ter sido programado previamente.

Portanto, os métodos de escolha de regras não são "inteligentes" tal e como nós o entendemos.

Para evitar o problema que isto representa foi criada toda uma teoria de modelagem que tenta esquematizar ao máximo todo o problema.

Assim foi criado inicialmente o conceito de espaço de busca. Quando devido ao aparecimento de um problema é ativada a primeira regra para resolvê-lo, esta poderá permitir a aplicação de um



Grafo 1. — Árvore de busca genérico, com as regras e sub-regras situadas nos distintos modos.

determinado número extra de regras e cada uma delas permitirá a aplicação de outras.

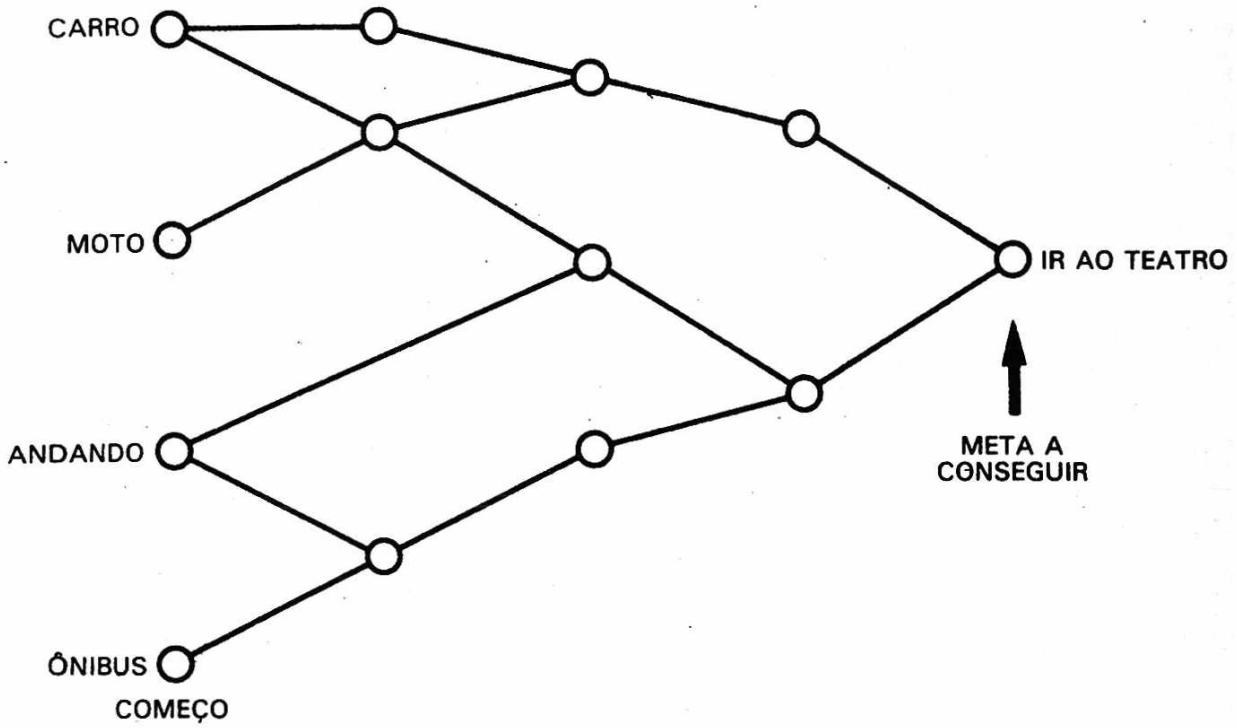
Assim é criada uma árvore (grafo 1), que representa o conjun-

to de possíveis aplicações de regras. O problema ficará resolvido quando ao longo de toda a árvore (árvore de busca) se consiga encontrar um caminho que nos leve à solução.

Em função da regra escolhida inicialmente se tem uma árvore diferente. A árvore completa para um problema determinado normalmente costuma ficar extensa.

Gestão das árvores de busca

Quando já se possui a árvore de busca para a resolução do problema aparecem muitas formas de gestioná-la. Para escolher entre elas costumam ser realizados estudos sobre a forma que possui dita árvore, distinguindo-se dois tipos básicos em função dos problemas que temos que resolver.



Grafo 2. — Possível árvore de busca para "ir ao teatro", pertencente ao grupo dos que vão reduzindo possibilidades ao avançar neles.

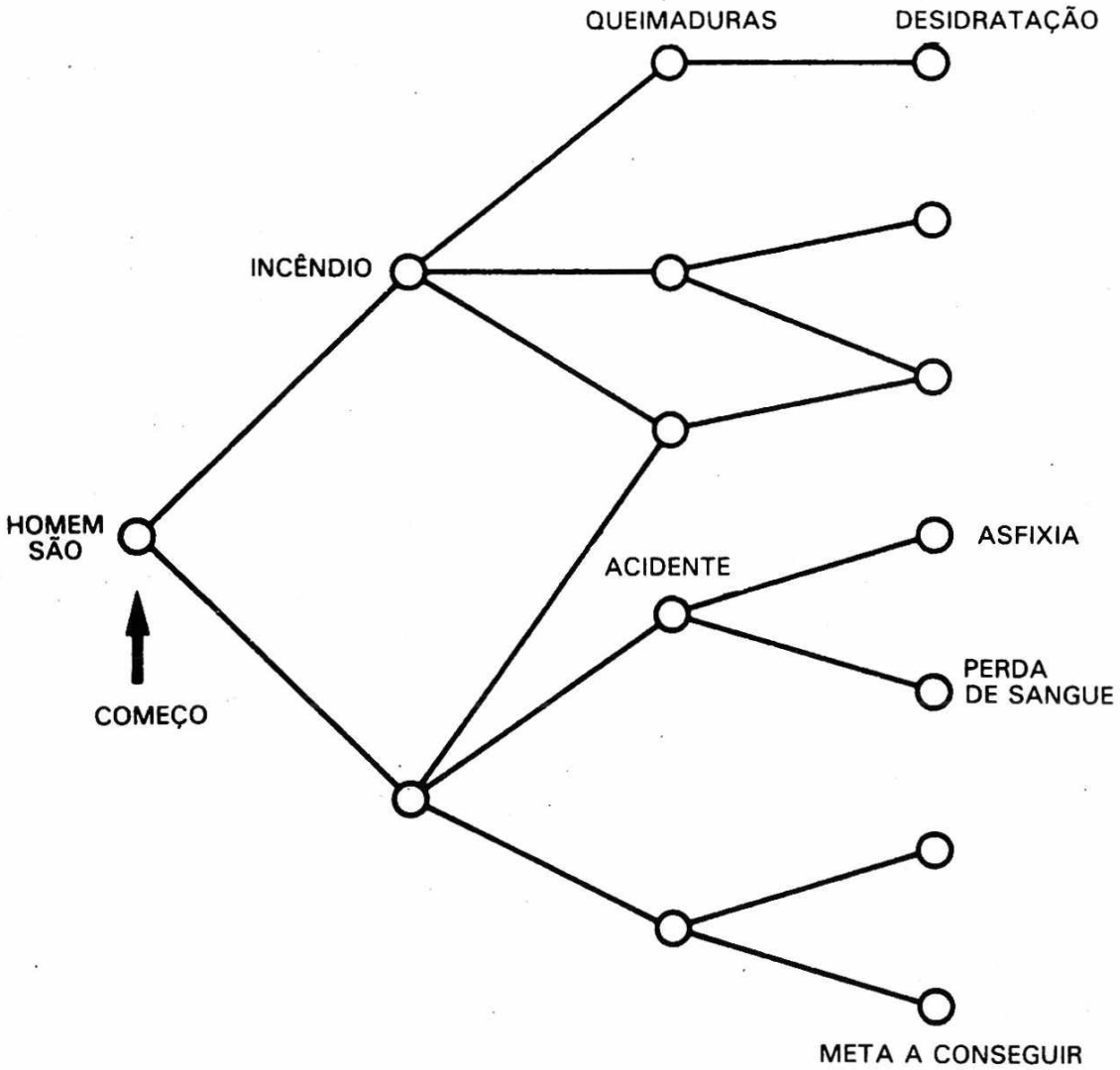
Suponhamos que o problema a resolver é que queremos ir ao teatro. Para ir ao teatro existem muitas formas possíveis: de carro, andando, de ônibus, etc.

O que devemos fazer é escolher entre as múltiplas formas a que mais nos convenha em função da distância, o clima, o preço, etc.

A árvore de busca que teremos será do tipo mostrado no grafo 2.

Existem outros problemas nos quais, contudo, a árvore de busca é completamente distinta. Por exemplo, quando queremos predizer algo porém não sabemos concretizar, posto que depende dos dados iniciais que se possuam.

Se trata de inferir possíveis conseqüências dos fatos. A árvore de busca seria algo parecido ao grafo 3.



Grafo 3. — Em algumas árvores o que se produz é uma maior "ramificação" conforme nos aprofundamos.

Estas duas árvores dão base a distintas maneiras de gestioná-las

Quando como na primeira árvore, se parte do objetivo a cumprir e é buscada a maneira de realizá-lo, ao método chamamos encadeamento desde atrás. Parte do final para chegar ao princípio.

No entanto, se, como no segundo método, se parte do começo e se deixa evoluir o sistema até que chega ao objetivo, chamamos encadeamento desde a frente.

Em função do tipo de problema que tenha de ser resolvido teremos que utilizar uma destas duas teorias para gestionar a escolha de regras correspondentes.

Isto significa que estes dois métodos são formas de delinear o problema e não políticas de escolha de regras.

Um caminho na árvore: escolha de regras

Uma vez escolhida a maneira de gestionar o problema será delineada a política de escolha de regras que se deseja. A escolha de regras representa encontrar um caminho sobre a árvore de busca. E, claro está, existem muitas formas de conseguir um caminho (escolher regras), cada uma delas diferentes.

Geralmente se diferenciam duas formas de realizar a escolha.

- **Escolher as regras de uma maneira mecânica**

É criada uma função pré-definida que escolhe as regras sempre desde um mesmo ponto de vista.

- **Escolher as regras de forma mais inteligente.**

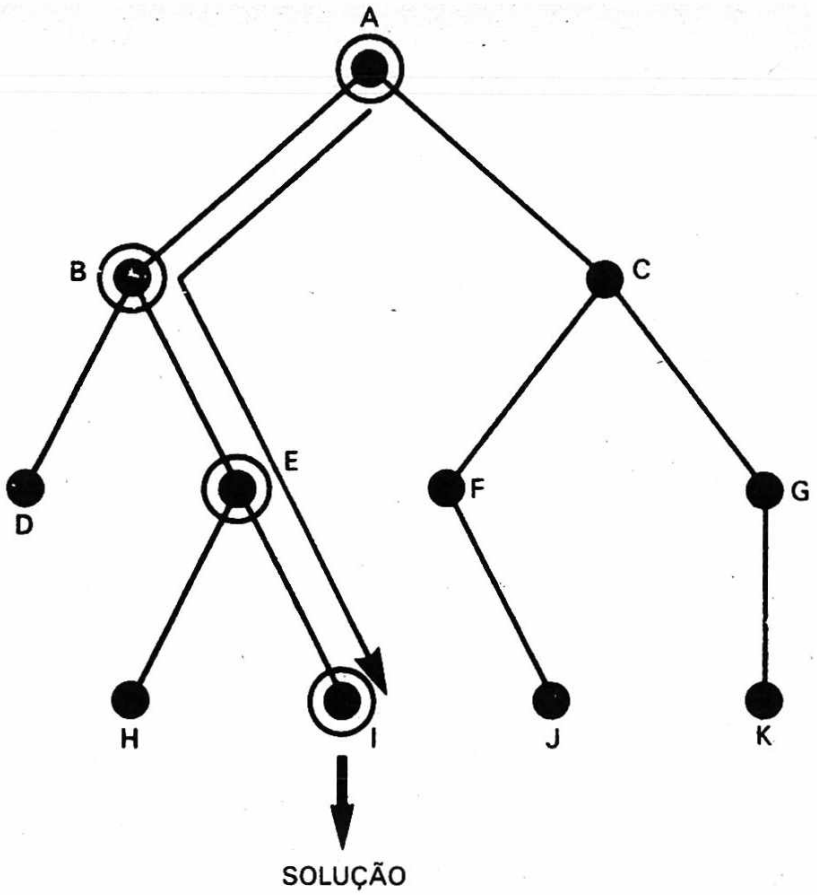
Geralmente são utilizados métodos de avaliação do estado em que se encontra o sistema de forma que em função da escolha de uma regra determinada seja alcançada mais rapidamente a solução. Muitas vezes é aplicada a experiência acumulada de outros processos. São os chamados métodos heurísticos.

ESCOLHA MECÂNICA

Para conseguir um caminho de forma mecânica na árvore de busca que representa ao problema delineado existem duas formas básicas.

BUSCA EM PROFUNDIDADE

Como seu nome indica, se trata de partir do nó inicial e buscar a solução ao problema aprofundando o máximo possível na árvore de busca. Quando se chega a um ponto onde não se pode con-



Grafo 4. — Caminho válido para alcançar a solução (e ótimo).

tinuar então se retrocede para buscar um novo caminho em profundidade.

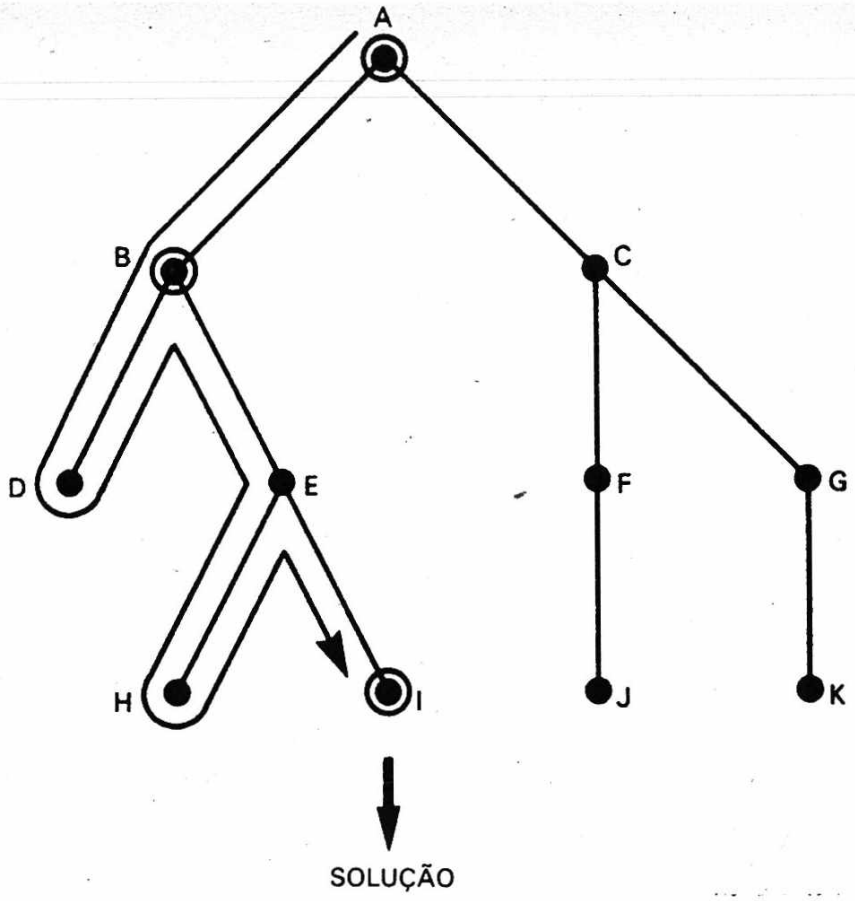
Vejamos um exemplo esclarecedor. Suponhamos que, vendo a árvore do grafo 4 resolvemos o caminho para conseguir a solução.

Da figura se deduz que se aplicamos as regras A, B, E e I (nessa ordem) chegamos a solucionar o problema.

Uma busca em profundidade buscaria a solução rastreando a árvore da seguinte forma (ver grafo 5): começaria por A, continuaria escolhendo um nó descendente do A, poderia ser o B ou o C. Pega o B de uma forma arbitrária.

Uma vez escolhido o B é analisado se chegou à solução, que é encontrada no nó I. Como não chegou se aprofunda até D. Ao chegar a D, que tampouco é solução, não se pode continuar; portanto volta a B para passar posteriormente a E. Depois a H e uma nova volta a E para finalmente chegar à solução.

Este é o método de busca em profundidade.



Grafo 5. — Processo seguido na busca mecânica em profundidade para alcançar a solução.

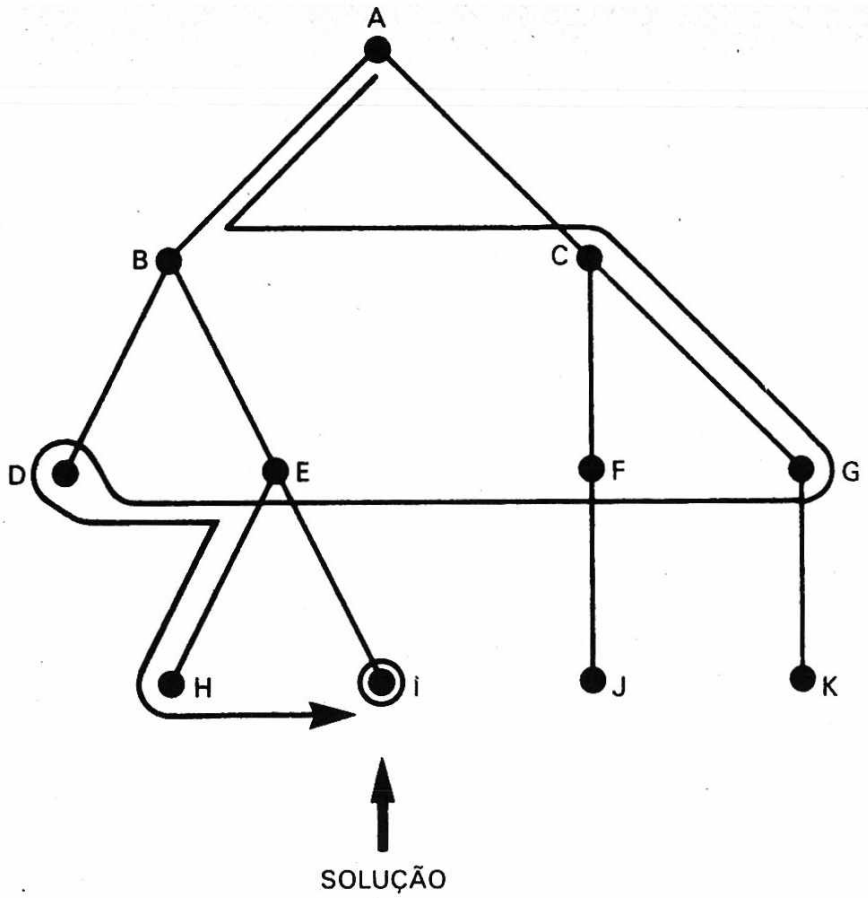
BUSCA EM EXTENSÃO

Vai percorrendo todos os nós de cada nível. Ante o mesmo problema, delineado na mesma árvore de inferência, uma busca em extensão atuaria da seguinte forma (representada no grafo 6).

Inicialmente partir-se-ia do nó A; como no caso anterior, passar-se-ia ao nó B. Uma vez analisado que não se chegou à solução, ao invés de continuar em profundidade se continua "em extensão", analisando o nó C. De C, como não existe outro nó "ao mesmo nível" se passa ao nó G. Assim vão sendo analisados todos os nós do mesmo nível até D. Baixa-se até H e de H se passa a I, a solução final.

O escolher um destes dois métodos depende, como no caso das estratégias de busca, do tipo de problema que se pretenda solucionar.

O que acontece normalmente é que os sistemas expertos são desenhados somente com um tipo de método, pelo qual adquirem



Grafo 6. — Sistema de busca mecânica por extensão.

peculiaridade que os tornam específicos para um tipo de problemas determinado.

ESCOLHA HEURÍSTICA

Ao fixar uma forma mecânica de escolher as regras (por exemplo, em profundidade) pode ser compreendido que o sistema não possui nenhum tipo de conhecimento sobre se durante sua operação está aproximando-se à solução ou então distanciando-se.

Simplesmente pega uma regra e comprova se chegou ao fim que se pretende; em caso negativo escolhe uma nova regra em função da norma que tem fixada.

O que pretendem os métodos heurísticos de escolha de regras é precisamente dar informação ao sistema acerca de como ficaria aplicando uma determinada regra. Uma vez que é possuída esta informação, o sistema pode escolher a regra que otimize sua operação.

Por exemplo, se conseguíssemos saber exatamente como se

comporta o sistema ante o problema que propusemos anteriormente, a escolha de regras seria precisamente a que desse o caminho direto desde o estado inicial até a solução, ou melhor, seriam aplicadas as regras A, B, E, I por essa ordem. Podemos comprovar como este método utiliza o mínimo número de regras. É o mais rápido e preciso.

Como realmente não existe um método capaz de conhecer perfeitamente o problema, são utilizadas aproximações que o que permitem é reduzir ao mínimo os erros na escolha de regras.

Estas aproximações são funções ou algorítmos que coloca o engenheiro do conhecimento e que ele deve escolher. Neste caminho são encontrados os últimos avanços em inteligência artificial que são realizados nos Estados Unidos, Japão e Europa.

CAPÍTULO V

PROCESSADORES DE LINGUAGEM NATURAL

As necessidades de informação e manejo informático aos quais estão submetidos a maioria dos profissionais nos mais diversos campos, está provocando uma mudança substancial na relação homem-computador. Não é o especialista em informática o que tem que contatar com a máquina, mas estudiosos de outros temas, normalmente usuários inexperientes em seu manejo.

A inteligência artificial permitiu desenhar uma série de ferramentas, os *processadores de linguagem natural*, que facilitam a inter-relação homem-computador e permitem uma comunicação muito mais fluída e menos estruturada que os sistemas tradicionais de menus.

Os processadores de linguagem natural estão compostos funcionalmente de um *analisador*, um *dicionário* e um *direcionador*. O analisador estuda, ou sintática ou semanticamente, a estrutura das orações com ajuda de regras incorporadas ao sistema. O dicionário expressa relações entre conceitos e armazena palavras, sendo o direcionador o encarregado de passar da linguagem natural analisada a uma linguagem formalizada, própria do sistema onde está sendo trabalhado.

Necessidade da linguagem natural na comunicação com o computador

Desde o momento em que o computador começou a ser in-

troduzido na indústria, nos serviços públicos e inclusive em nossos lares, surgiu a necessidade de facilitar ao usuário a comunicação com o mesmo. Desde o início da era do computador uma das grandes tendências foi a de facilitar o trabalho com as máquinas, sendo estas as que levam o maior peso possível em dita interrelação. É hora de desenvolver pacotes de software que permitam ao usuário consultar por si mesmo, sem muitos conhecimentos prévios e em sua linguagem natural, e que seja o computador quem realize a tarefa de traduzí-lo à linguagem formal da própria máquina.

Se bem que o campo da linguagem natural tem uma acidentada e triste história, a inteligência artificial e as ferramentas que utiliza podem ser a tecnologia que fazia falta para um funcionamento satisfatório dos interfaces. É necessário ressaltar desde o princípio que os processadores de linguagem natural (PLN) são todavia muito limitados em suas aplicações para não deixar-se inundar por um excessivo otimismo. Os problemas inerentes à natureza da linguagem, tais como sua *escassa lógica* ou suas *múltiplas estruturas*, não estão nem muito menos resolvidos e existem muitos difamadores de sua utilização.

A aplicação mais imediata dos processadores de linguagem natural é a recuperação de informação. Uma vez que são desenvolvidos sofisticados métodos de manejo de informação, tais como os sistemas inteligentes de recolhimento, catalogação e recuperação automática, se viu a necessidade de melhorar também o interface com ditos bancos de dados. Se for conseguido ensinar ao computador o suficiente da linguagem natural para que seja capaz de entender as perguntas estabelecidas, são eliminados dois grandes inconvenientes:

- O intermediário humano. É difícil que o intermediário compreenda realmente o que o usuário quer perguntar devido, fundamentalmente, ao distinto conhecimento do tema que possuem ambas as pessoas;
- Ter que ensinar ao usuário, profissional em outros campos, as linguagens próprias do computador.

Funções que realiza um processador de linguagem natural

Os processadores de linguagem natural são capazes de realizar uma variada série de funções:

- Permitem ao usuário ter um menor conhecimento do sis-

tema, devido a que evita a utilização de estruturas fixas.

- Corrigem erros de tipo léxico. Ao reconhecer uma palavra que contém uma falta, o programa tentará associá-la com alguma do dicionário. Se alguma destas encaixa o suficiente confirma a correção da falha e a frase é analisada. Ladeer, sistema experto sobre informação naval, nos proporciona um exemplo:

  *Quanto é distante Kitty Hw de Gibraltar?
  Correção Hawk
  Analisado.

- Permitir ao usuário construir suas frases de maneira incrementada, referindo-se a frases anteriores. Esta capacidade é denominada anáfora e existem dois tipos:

  * Substituição e elipse. O PLN pode substituir o significado de alguma palavra prévia (substituição) ou então entender uma pergunta na qual não aparece nenhuma referência prévia (elipse).

  * Quais carros correm a mais de 100 Km/h?
  * Quais deles entre 100 e 150 km/h? (substituição)
  * Quais entre 150 e 200 km/h? (elipse)

  * Referências a pronomes. Um pronome, diferentemente de uma substituição, costuma ser aplicado a valores anteriores mais que a significados.

  * Por que cortaste a madeira?
  * Para fazer uma cama
  * Como o fizeste?

- Responder a perguntas de forma inteligente.
  À pergunta:

  Está Berenice matriculada no oitavo curso?
  (E o oitavo curso não existe), a reposta pode ser:

  * Não

  * Não, não existe o oitavo curso (muito mais clara e precisa).

- Permitir a ampliação das regras gramaticais. É de muita utilidade que os PLN possam ser ampliados facilmente, pois assim permitem sua simples adaptação a todo tipo de usos particulares.

Problemas que apresentam os PLN

Do conjunto de problemas que afetam aos processadores de

linguagem natural vamos citar os mais importantes. Um conjunto de problemas surge da gramática da linguagem. O ideal seria um processador que tivesse toda a gramática incluída, porém é óbvio que, pelo momento, isto excede a capacidade dos computadores, ao existir um número muito importante de regras em uma sintaxe completa. A solução aponta para duas vias: desenvolver somente um subconjunto da língua, aplicando algumas quantas regras sintáticas, ou então obviar a falta de uma gramática completa construindo um sistema que tenha em conta primeiramente considerações de tipo semântico.

Uma dificuldade tão importante como a anterior são as perguntas gramaticalmente mal construídas. Certos pesquisadores se inclinam por construir processadores que se antecipem ao erro incluindo-o nas regras sintáticas, enquanto que outros, fazendo menos "pé-firme" em considerações sintáticas, conseguem sistemas menos sensíveis a ditos erros.

Outro problema a considerar, por sua ampla utilização, é o uso de pronomes. Ao aparecer um, o analisador deve resolver antes de poder enviar a pergunta ao sistema. Vejamos com um exemplo a importância deste fator.

* Todas as garotas têm moto?
Sim
*Quantas têm mais de 18 anos?

O usuário tem em mente que a idade de conduzir uma moto é ao redor aos 18 anos, pelo qual a segunda pergunta alude às garotas. O computador, por sua parte, pode não ser partícipe de dito conhecimento e não saber qual dos nomes — Garota, — Moto concorda com a pergunta.

Estes e outros problemas dão uma idéia da complexidade dos processadores de linguagem natural e do longo caminho de pesquisa que fica por percorrer. Se pensamos que a solução passa porque o usuário fala melhor é desvirtuado o propósito destes interfaces, que é permitir ao profano comunicar-se em seu próprio idioma. Toda tentativa de colocar restrições é desvirtuar sua finalidade.

Como resumo podemos dizer que os processadores de linguagem natural não chegarão ao mercado até dentro de alguns anos devido fundamentalmente a três fatores:

- São requeridos grandes computadores para sua implantação. Não somente para conter ao PLN, mas também o sistema ao qual vai associado deve ser de certa importância e capacidade.
- Padronizar toda a linguagem é uma árdua tarefa, por isso,

de momento, serão utilizadas linguagens parciais.
- A linguagem humana é pouco apta para transmitir informação clara, consisa e precisa.

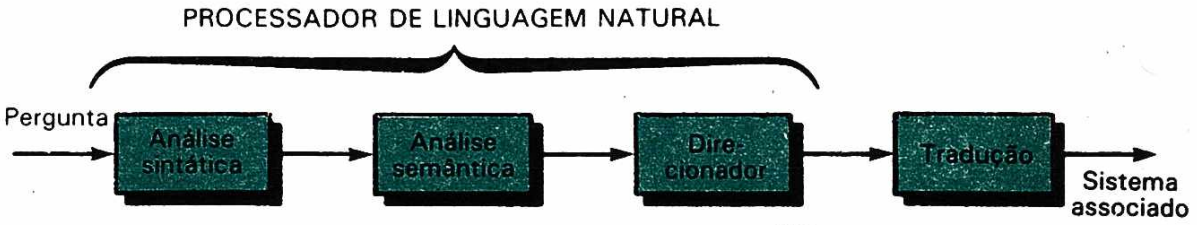


Figura 1. — Fluxo de trabalho de um processador de linguagem natural. Partindo de uma pergunta é criada uma estrutura única compreensível pelo sistema associado.

Arquitetura de um processador de linguagem natural

Um processador de linguagem natural é um dos módulos que compõem um sistema de recuperação de informação. Sua missão é analisar, tanto semântica como sintaticamente, as perguntas do usuário e criar uma estrutura formal que em seguida seja traduzida à linguagem própria do sistema associado.

O processo que segue uma pergunta pode ser resumido mediante um diagrama de blocos (Fig. 1); o processador de linguagem natural compreende unicamente os três primeiros módulos.

Quase todos os produtos que hoje em dia são comercializados no mercado apresentam esta distribuição. O intérprete francês SAPHIR, como pode ser apreciado na figura 2, tem uma grande semelhança com o esquema típico destes sistemas. Nele podem ser apreciados os módulos de correção de erros morfológicos e léxicos, assim como o que realiza a função de concretizar, estabelecendo um diálogo com o usuário, as perguntas ambíguas ou imprecisas.

Uma vez visto de forma geral onde é encravado um PLN, o seguinte passo é descrever em maior detalhe os distintos módulos que compõem dito sistema (Analisador, Dicionário e Direcionador) e sua implantação em diversos programas.

O analisador

O analisador é o programa que estuda a gramática da oração. Dita gramática definirá a função de cada palavra dentro da frase. A chave do êxito ou do fracasso de um processador de linguagem natural é a representação de dita linguagem.

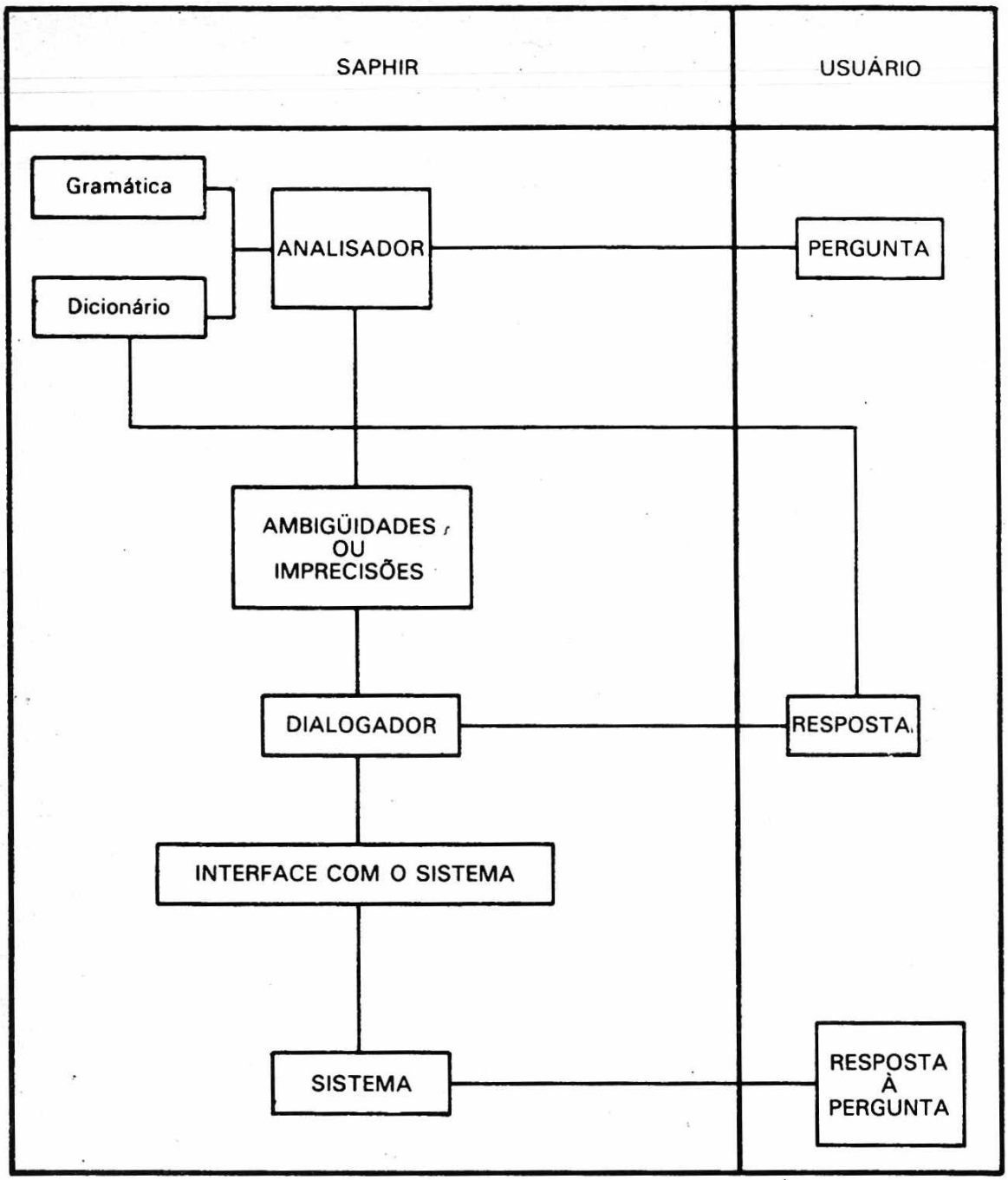


Figura 2. — Arquitetura real de um PLN, o SAPHIR, desenvolvido na França como interface para um banco de dados inteligente.

Entre os distintos tipos de representações as mais comumente usadas são as árvores funcionais e as estruturas temáticas, apoiando-se ambas na forma em que uma idéia é expressa em palavras e frases. Na primeira o analisador (ou parser) cria as árvores a partir das regras sintáticas de construção de orações, enquan-

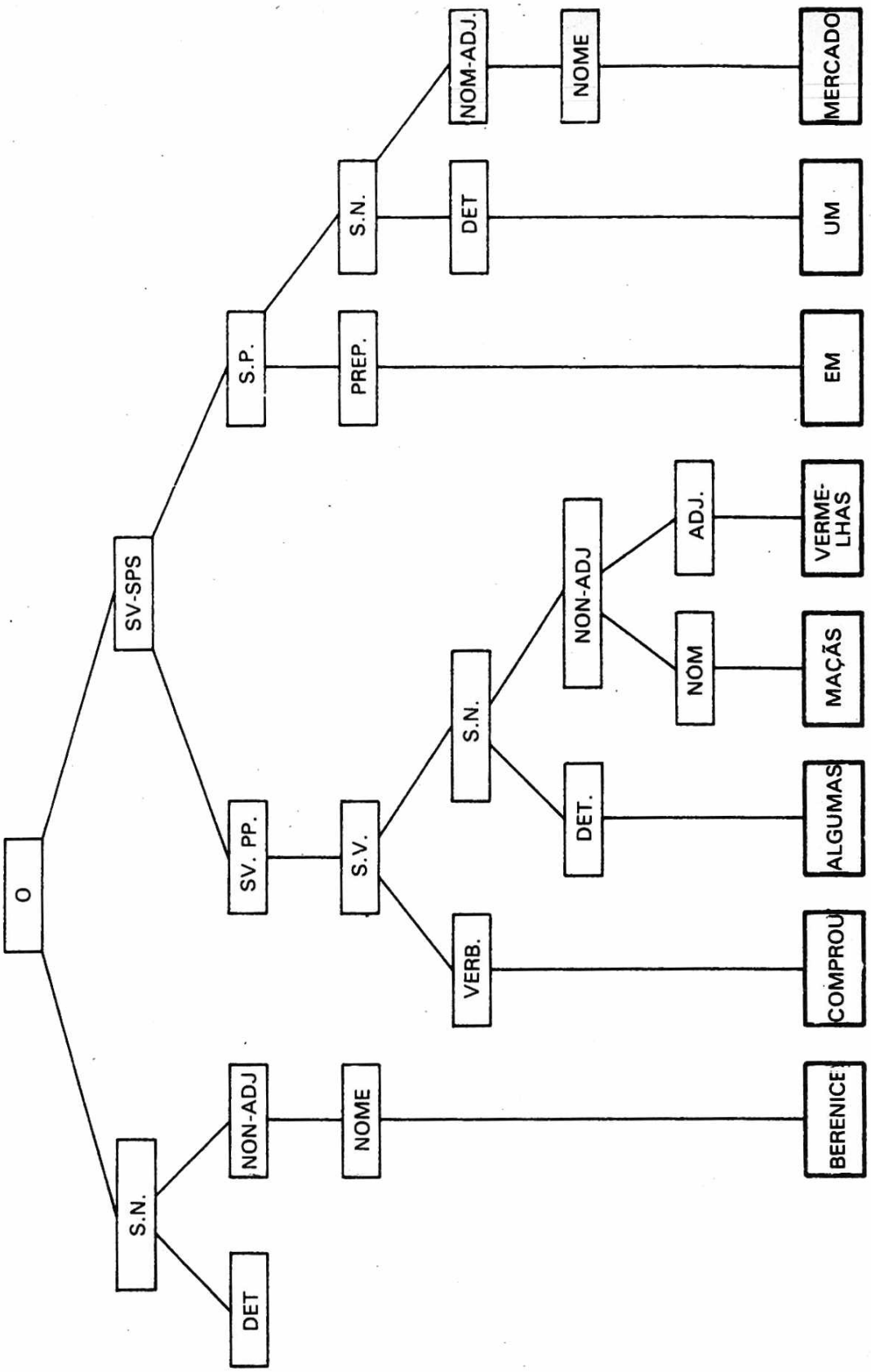


Figura 3. — Árvore funcional resultado de uma análise sintática mediante uma gramática de livre contexto.

to que o processador de estruturas determina como os distintos sintagmas nominais são relacionados com o verbo.

Se considerarmos a frase "Berenice comprou umas maçãs vermelhas em um mercado", a árvore funcional seria composta (tal e como é apreciado na figura 3) de um sintagma nominal, mais o sintagma verbal, composto por sua vez do verbo com seu sintagma nominal, e a frase preposicional com seu sintagma nominal correspondente.

Por outra parte, o processador de estruturas temáticas indicaria que o ato é "comprar", o agente "Berenice", o objeto temático "maçãs vermelhas" e o lugar ou procedência "em um mercado" (Fig. 4).

Como vemos, os métodos são completamente diferentes: nas árvores a finalidade do analisador é determinar a função de cada palavra dentro da oração, enquanto que nas estruturas temáticas é determinar como se relaciona cada sintagma nominal com o verbo. São duas teorias que em alguns casos podem ser complementadas, a *sintaxe* e a *semântica*.

Dentro das árvores funcionais foram definidos quatro méto-

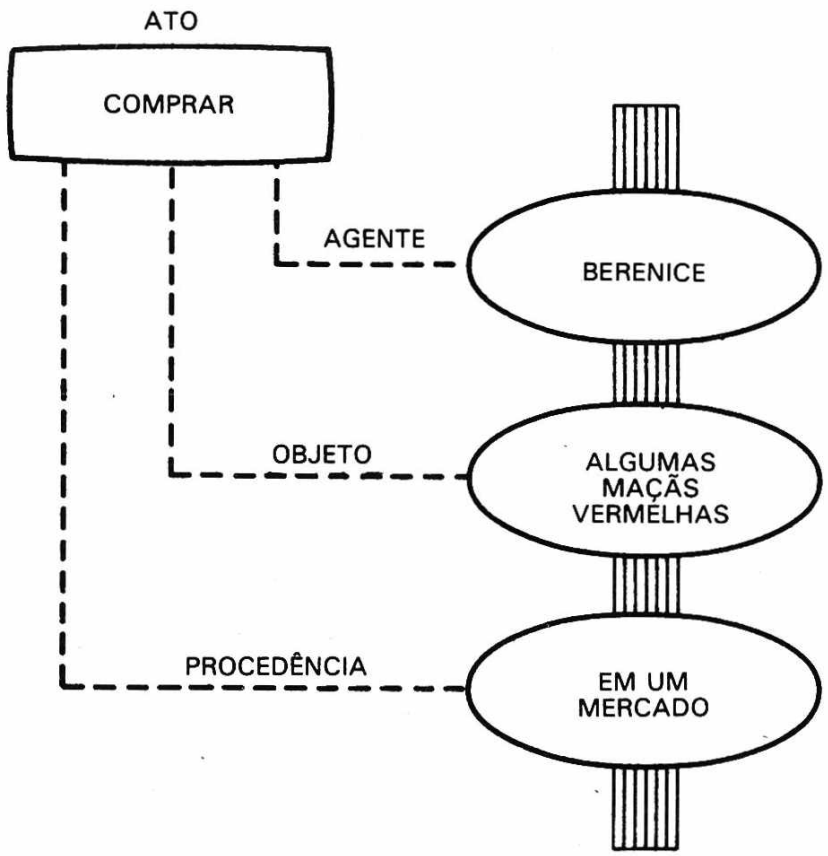


Figura 4. — Estrutura temática da oração "Berenice comprou algumas maçãs vermelhas em um mercado". A análise é semântica.

dos de construção dos mesmos: a gramática de livre contexto, as redes de transição, as redes de transição aumentadas (ATN) e o analisador WASP (Wait-and-see), sendo cada um deles mais complexo que o anterior.

Ao contrário, nas estruturas temáticas somente existe um método: averiguar o papel que os sintagmas nominais jogam dentro da oração. Um deles será identificado com o sujeito que realiza a ação, outro com o que se realiza (objeto direto) e outros com diversas circunstâncias. O número de papéis temáticos varia segundo o tema ou a pessoa que construa o sistema porém podem ser destacadas as seguintes;

— Sujeito ou Agente	Berenice comeu um pastel.
— Objeto temático	Nico quebrou o rádio.
— Instrumento	Nico quebrou o rádio com *um martelo.*
— Lugar	Carlos estuda *em casa*
— Beneficiário	Alice pegou uma banana *para Antonio.*
— Destino	Os alunos foram *a Santos.*

O DICIONÁRIO

A função fundamental de um dicionário, à parte de armazenar um conjunto de palavras, é representar as relações entre as mesmas. Uma palavra pode representar uma relação, um atributo ou um valor. Da mesma forma que o analisador deve conter categorias gramaticais e, por outra parte,um grande número de significados.

O dicionário pode estar estruturado em forma de redes, onde cada nó é uma palavra e o arco entre dois nós a relação entre os mesmos, ou então em forma de campos hierárquicos.

Uma característica que deve incorporar o dicionário é a possibilidade de aceitar novas palavras que o usuário vai introduzindo na conversação, por exemplo:

(usuário) — O professor dá aulas de LATIM.
(computador * Não entendo a palavra LATIM.
1) É um lugar.
2) É um idioma.

Na próxima vez o dicionário identificará a palavra latim corretamente.

O DIRECIONADOR

A representação interna das perguntas em linguagem natu-

ral é um dos aspectos chaves em um processador. De todas as possíveis representações devemos optar pela que mais facilmente se ajuste ao idioma utilizado. Não devem existir, ao contrário que a linguagem natural, inexatidões de nenhum tipo: todos os predicados na representação interna devem ser precisos. Uma ambigüidade pode produzir erros posteriores. Da mesma forma, a estrutura funcional (isto é, como estão ordenados funcionalmente os distintos objetos) deve ficar explícito no predicado.

Processadores de alto nível

Os processadores de alto nível conectam com sistemas de recuperação de informação não tradicionais; seu campo de ação são os sistemas expertos, com os quais formam um conjunto de prestações muito apreciáveis. Para dar uma idéia da capacidade destes processadores vejamos como é capaz de generalizar idéias e catalogar conceitos um sistema denominado "Anão Klaus":

Um submarino é um barco.
* Está dizendo que todo submarino é um barco?
Sim
* O.K. Agora assimilou o que é um submarino.
Um veleiro é um barco.
* Está dizendo que todo veleiro é um barco.
* É o submarino uma classe de veleiro?
Não.
* Está o submarino composto de diferentes elementos que o veleiro?
Sim.
* Compõem o submarino e o veleiro todos os barcos?
Não.
.
.
.

A partir desta conversação o sistema classificará aos submarinos e aos veleiros corretamente dentro do sistema.

Os sistemas de alto nível não estão atualmente comercializados e somente existem alguns poucos protótipos em experimentação.

CAPÍTULO VI

LINGUAGENS DE PROGRAMAÇÃO E FERRAMENTAS DE I.A.

Uma vez que foram estudados os métodos de representação do conhecimento e as técnicas de controle é estabelecida a eterna pergunta: Como se consegue criar um sistema experto? A resposta, em sua origem, é clássica: posto que um sistema experto é um programa de computador, para realizá-lo terá que utilizar uma linguagem de programação.

Se for respondida um pouco mais em profundidade a resposta tem que realizar algumas matizes. Obviamente é necessário uma linguagem de programação, mas haverá algumas linguagens que se adaptam melhor à nossas necessidades que outras; inclusive podem ser utilizadas ferramentas mais evoluídas que as linguagens clássicas, ainda que estejam baseadas nelas.

Baseando-se nos diversos níveis de software foram criadas e diferenciadas três ferramentas distintas que posteriormente serão analisadas: as linguagens de propósito geral, os sistemas esqueletos e os sistemas conchas. Cada um deles vai aumentando em simplicidade de utilização para o usuário, mas, por sua vez, vai perdendo generalidade em suas aplicações.

Níveis de software

A figura 1 mostra os níveis em que é dividido o software necessário para resolver um problema mediante o hardware de um computador. Na parte superior da figura é refletido o problema ao qual

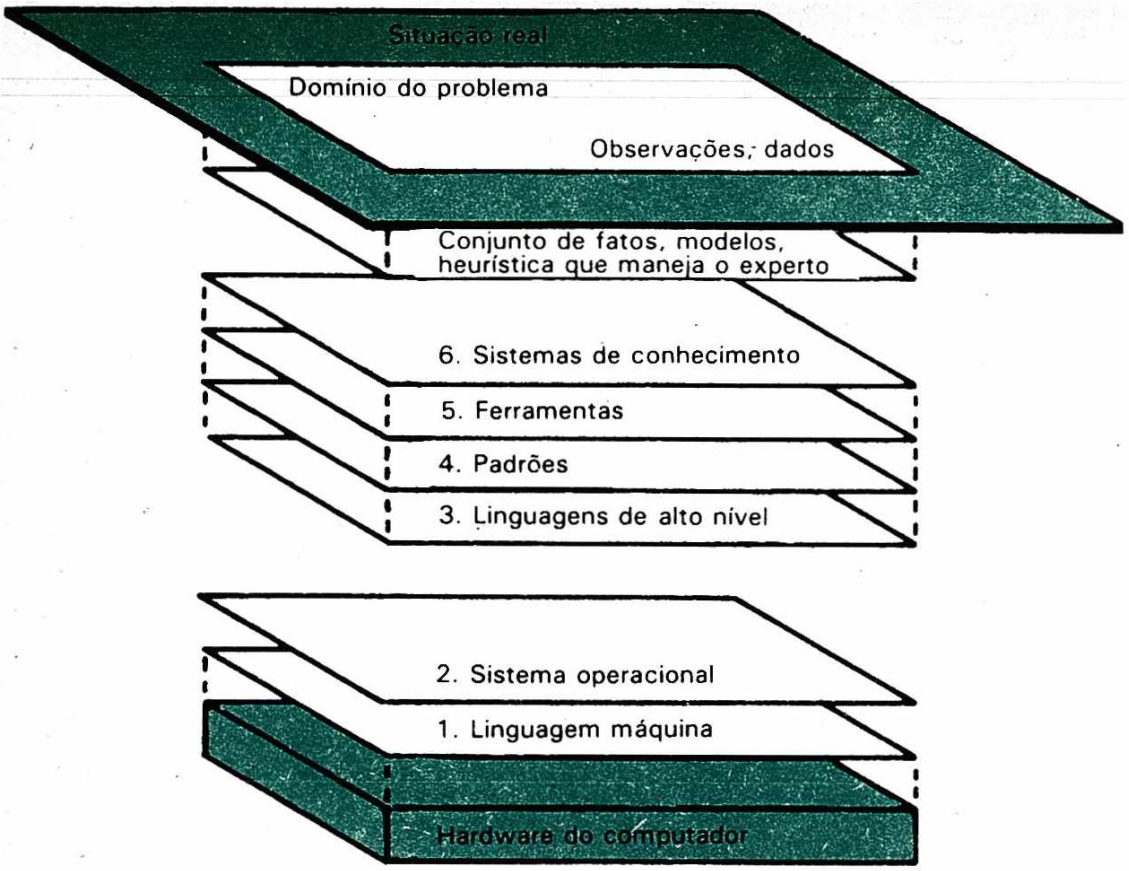


Figura 1. — Os seis níveis de software possíveis entre o problema real e o hardware do computador.

se enfrenta o experto na matéria, uma parte do mundo real: quando se deseja resolver dito problema tudo o que observamos é como o experto maneja os dados de entrada e a partir deles chega a algumas conclusões. O que não é apreciado são os modelos mentais, as regras de decisão e as estratégias que usa o experto quando decide o que fazer em uma determinada situação particular. Este tipo de conhecimento é o seguinte degrau na resolução de qualquer situação e é realmente o mais imprescindível.

Por outro lado, independentemente do software que é utilizado, todos os sistemas expertos dependem no fundo do hardware do computador. A parte física do mesmo é, ao fim e ao cabo, a encarregada de manejar em último termo toda a informação em forma de códigos binários, uma longa série de zeros e uns que o computador converterá em respostas físicas discretas. Este é o nível inferior do sotware denominado linguagem de máquina.

O nível imediatamente superior é o programa que dirige e direciona as operações fundamentais do computador: o sistema operacional; é encarregado de controlar as sentenças da linguagem

de máquina. Costuma estar escrito em linguagem de máquina ou, recentemente, impresso em hardware dentro de um chip.

Introduzindo-nos no que é propriamente a programação, quase toda ela é realizada nas denominadas linguagens de alto nível. As mais conhecidas são Basic, Cobol, Pascal, Fortran, Forth, C, etc. No entanto, os programadores em inteligência artificial utilizam fundamentalmente duas linguagens: LISP e PROLOG. LISP consiste em um conjunto de instruções que facilitam a criação de programas que manejam listas, enquanto que o PROLOG facilita o trabalho com expressões lógicas. Ambas as linguagens são muito úteis por seu caráter simbólico, enquanto que as outras trabalham melhor com cálculos numéricos.

Exatamente acima das linguagens de alto nível são encontrados os *padrões de programação*. Ditos padrões costumam estar associados com uma determinada linguagem e contém um conjunto de instruções escritas em dita linguagem, muito úteis para certas tarefas de programação. Para os conhecedores de linguagens estruturadas podem ser comparadas com as subrotinas.

As ferramentas são o quinto nível na figura 1. Estas ferramentas foram criadas para ajudar ao rápido desenvolvimento dos sistemas expertos. Os aspectos mais importantes que apresentam são as estratégias de controle, representação do conhecimento e inferência comuns à maioria dos sistemas expertos. A razão de uma ferramenta deste tipo é similar às de uma carpintaria: ao invés de criar uma nova para cada tipo de móvel, são utilizadas aquelas que nos foram úteis em anteriores ocasiões. Certamente, deve ser usada a ferramenta mais apropriada para cada situação; cada uma delas está especialmente desenhada para um trabalho em particular.

Quando é combinada uma ferramenta com o conhecimento sobre um tema específico, o resultado é um sistema experto baseado no conhecimento (nível 6). Dito sistema, se está bem desenvolvido, apresenta a mesma capacidade que o experto em dito tema.

Das linguagens aos sistemas concha

Em geral as linguagens são mais flexíveis, porém mais difíceis de usar na criação e utilização de um protótipo rapidamente. Somente quando se necessita uma aplicação muito particular ou quando o programador está muito bem preparado são construídos sistemas baseados em LISP ou PROLOG. Os sistemas concha

são menos flexíveis, já que levam incorporado um particular sistema de controle. Como conseqüência, se temos uma ferramenta apropriada a nosso problema, o desenvolvimento é muito rápido, e inclusive pessoas com muito pouca experiência podem criar pequenos porém muito úteis sistemas baseados no conhecimento.

Na figura 2 é apresentada uma vista geral dos sistemas e linguagens mais conhecidas. Os sistemas esqueleto ou padrões são situados na metade do caminho entre as linguagens ou os sistemas concha, e suas características são uma mistura de flexibilidade e facilidade de utilização.

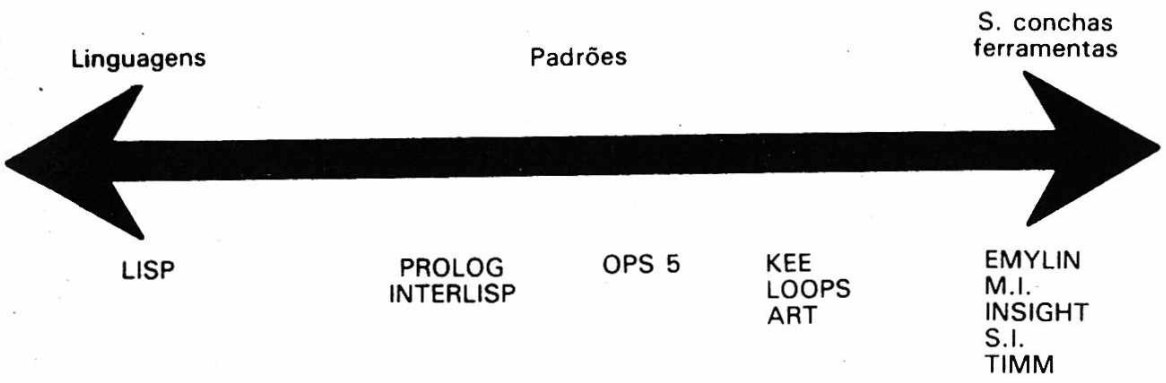


Figura 2. — Das linguagens de propósito geral (LISP, PROLOG) às ferramentas mais específicas. EMYCIN, S1,...

LISP é uma linguagem pura, enquanto que PROLOG ou INTERLISP estão mais próximas dos padrões. OPS 5, desenvolvido pela Carnegie-Mellon University, seria propriamente um padrão: contém um determinado sistema de controle das regras, mas é bastante flexível. KEE é uma ferramenta ou sistema concha, porém híbrida, isto é: pode optar por diversos controles e formas de representar o conhecimento; no entanto, é muito difícil de utilizar. Por último EMYCIN, SI, etc., são muito fáceis de usar, mas se ajustam a um tipo específico de sistemas expertos.

A figura 3 introduz uma nova complexidade na análise. As linguagens de programação em Inteligência Artificial costumam ser LISP ou PROLOG, porém é certo que um sistema experto pode ser escrito em Fortran ou Pascal, por exemplo. Quais são as vantagens e os inconvenientes de sua utilização? As linguagens de Inteligência Artificial têm algumas características que lhes possibilitam o manejo de símbolos enquanto que as linguagens convencionais trabalham melhor com números. É muito mais conveniente, por-

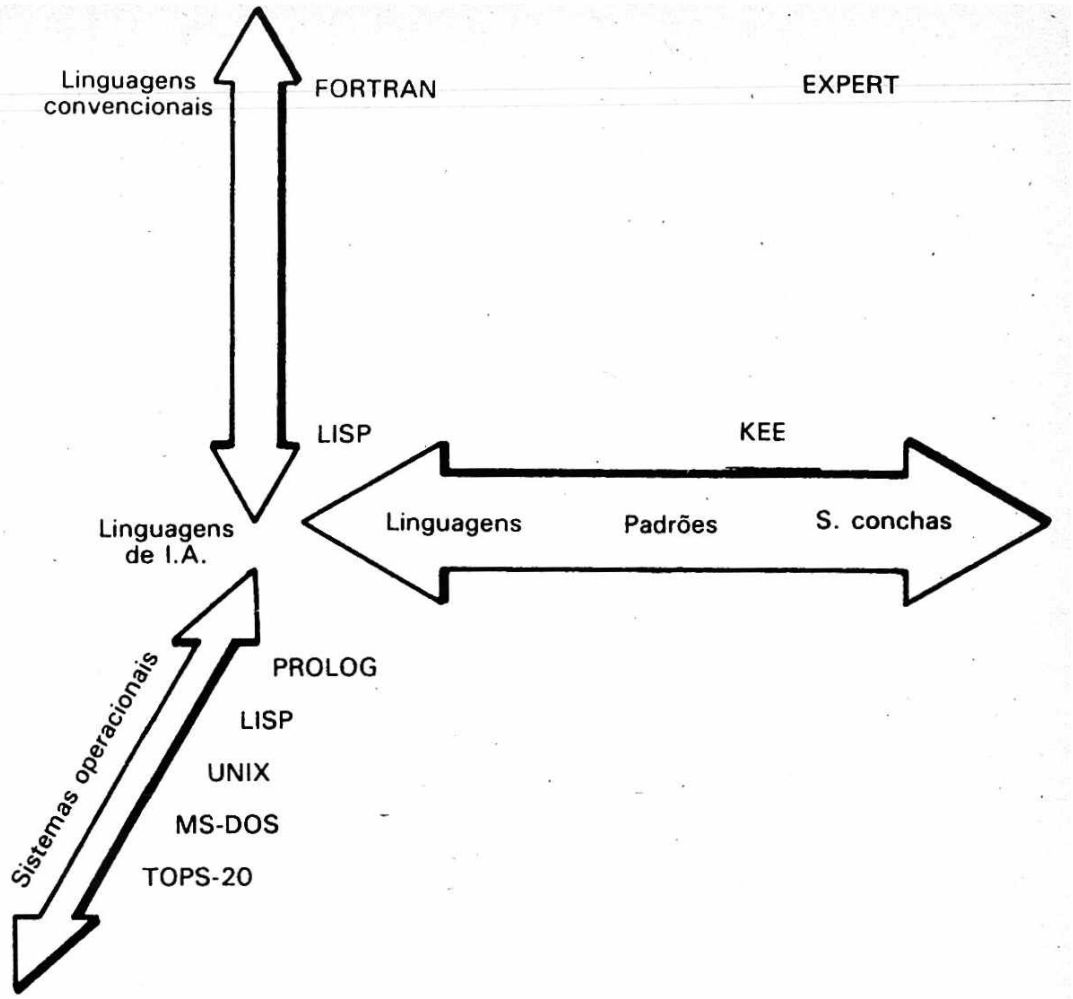


Figura 3. — Estrutura tridimensional das linguagens, as ferramentas e os sistemas operacionais em que trabalham.

tanto, ter ferramentas ou padrões programados internamente em linguagens LISP ou PROLOG que em linguagens clássicas.

A mais clara desvantagem para as "especializadas" é apresentada em que são menos conhecidas e estão menos estendidas que as clássicas linguagens de quarta geração; enquanto que quase todos os computadores têm implantados FORTRAN e PASCAL, poucos deles têm LISP ou PROLOG. Muitos dos sistemas concha estão, portanto, escritos em PASCAL, o que facilita sua implantação em toda classe de computadores, ainda que ao final sejam menos eficazes.

Outro problema é estabelecido no modo ineficaz em que os sistemas operacionais convencionais traduzem o LISP à linguagem de màquina, o que provoca a lentidão de trabalho de muitos sistemas expertos. O futuro mais próximo é a construção de máquinas cujo sistema operacional esteja escrito em LISP.

O estudo das linguagens de programação não é tarefa deste livro e por sua importância e interesse será objeto de outro volume da B.B.I., porém queremos oferecer uma pequena introdução às características mais sobressalentes dos mais importantes.

LISP

Pode ser dito que a linguagem LISP é a única utilizada pelos americanos dentro da inteligência artificial. LISP provém de List Processing Linguaje; foi criada por John McCarthy em 1958 e é, portanto a mais antiga das linguagens em uso depois do FORTRAN. Seu autor resumiu em 1978 suas características mais importantes:

- Trabalha com expressões simbólicas mais que com números; isto é, os bits da memória e os registros podem ser ocupados por símbolos arbitrários e não somente por operadores aritméticos.
- Processamento de listas; representação dos dados em listas estruturadas dentro da memória.
- Estrutura de controle baseada na composição de funções básicas para formar outras mais complexas.
- Utilização da Recursividade como uma forma de escrever processos e resolver problemas.
- Representação interna dos programas escritos em LISP como um conjunto estruturado de listas
- A função EVAL, escrita por sua vez em LISP, serve como um intérprete do mesmo e uma definição formal da linguagem.

A conclusão mais importante que se deduz é que não existem diferenças essenciais entre os dados e as sentenças do programa, ou melhor: um programa LISP pode usar alguns dados que, por sua vez, são outro programa LISP. Esta característica é traduzida na facilidade de modificar e ampliar um programa LISP. Conseqëntemente, o sistema experto associado é capaz, por uma parte, de ser modificado à vontade e, por outra, de diferenciar claramente o controle e as regras, ainda que no fundo ambas as coisas sejam programas LISP.

Existem muito poucas funções básicas em LISP; todas as demais são desenvolvidas a partir destas. Partindo da unidade fundamental, a lista (que por sua vez é composta de unidades indivisíveis denominadas átomos), é criada a rede do programa. Dita rede é construída mediante os nós CONS. Cada átomo tem associado uma lista de propriedades que contém informação sobre o áto-

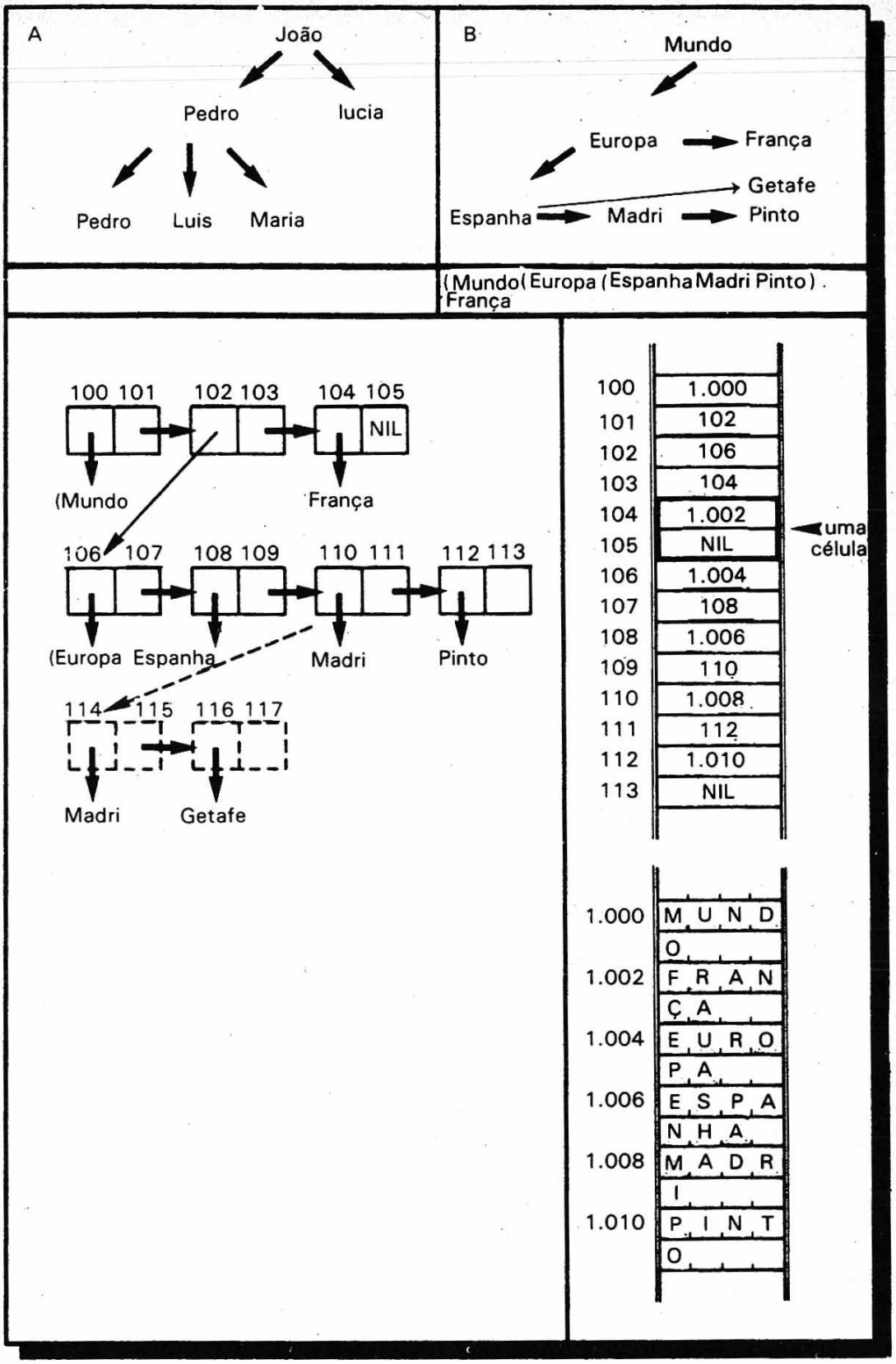


Figura 4. — A representação do conhecimento em LISP costuma ser realizada mediante árvores ou grafos de relações. Ditos grafos são representados mediante listas, facilmente codificáveis em memória.

mo: nome, valor e qualquer outra propriedade que se deseja. Os nós CONS são uma estrutura de dados dinâmica que é composta de dois campos (Fig. 4), cada um dos quais aponta para outro dado. Estes encadeamentos são facilmente modificáveis.

O manejo destas listas é realizado mediante as seguintes funções:

- CAR — Proporciona o valor do primeiro elemento de uma lista:
  CAR (Come João Batatas) = "Come"
- CDR — Dá o valor da lista sem o primeiro elemento:
  CDR (Come João Batatas) = (João Batatas)

- CONS — Junta duas expressões para formar uma lista:

  CONS COME (JOAO) = (COME JOÃO)
  átomo lista
  CONS LUIZ ( ) = (LUIS)

  átomo lista
  vazia

- ATOM — Analisa se uma estrutura é lista ou átomo; se é lista devolve o valor (False)

- EQUAL — Proporciona o valor (True) se as duas expressões são iguais.

A programação em LISP é executada construindo funções muito simples que realizam determinadas tarefas e em seguida agrupando as mesmas para completar o trabalho global.

O LISP não evoluiu muito ao longo dos últimos anos devido a seu caráter experimental que teve até agora, e a suas características próprias; no entanto, no quadro da figura 5 podem ser apreciadas as tendências que seguiu.

PROLOG

PROLOG, abreviatura de PROgraming language for LOGic, foi desenvolvida inicialmente por Colmeraver e P. Roussel em 1972 na Universidade de Marselha.

PROLOG é uma linguagem de programação que implementa uma versão simplificada do cálculo de predicados e utiliza uma lin-

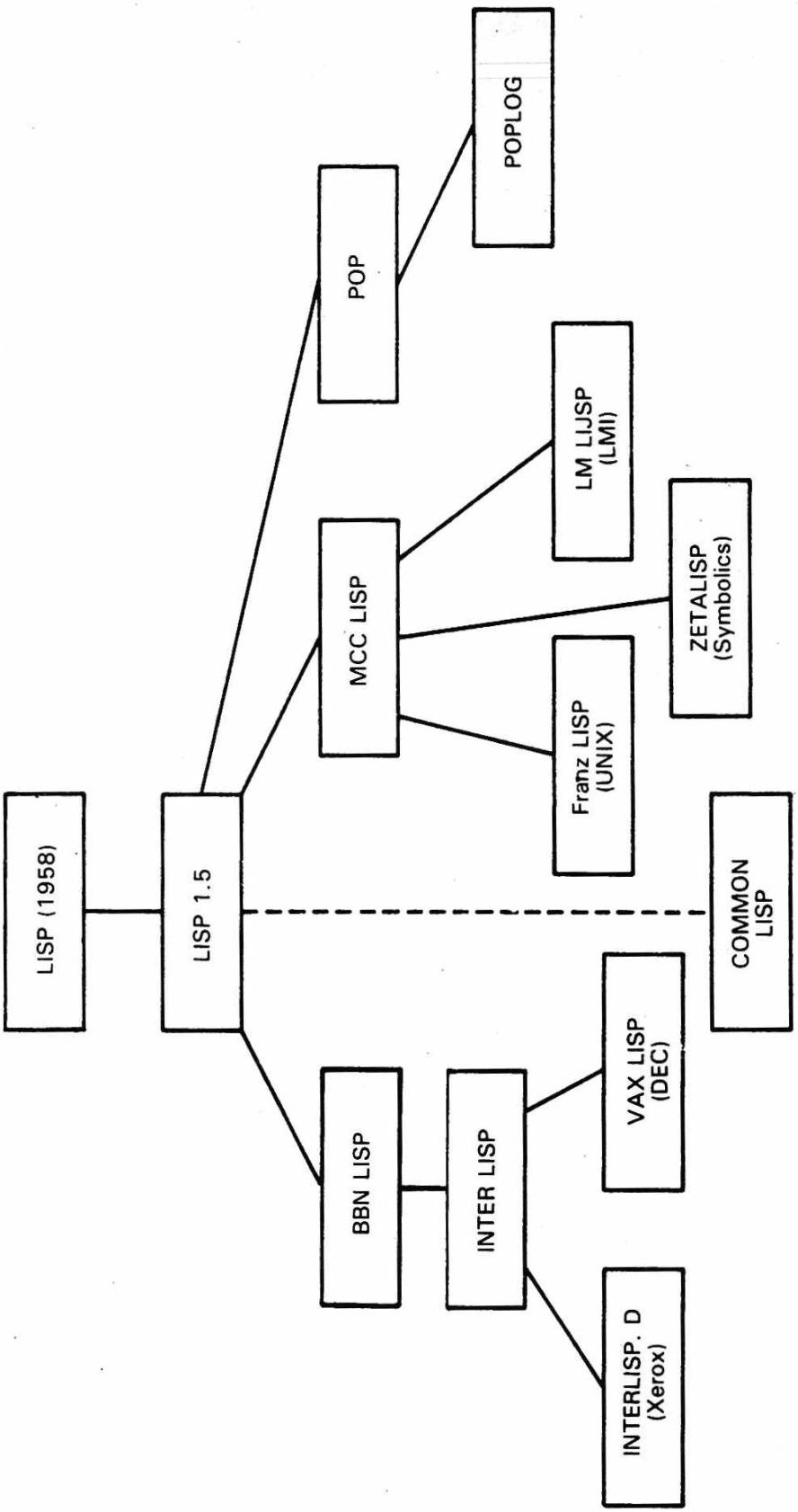


Figura 5. — O desenvolvimento da linguagem LISP foi produzido durante os últimos anos, porém as diferenças não são muito grandes.

guagem lógica. Foi ultimamente muito popularizada devido a sua inclusão dentro do programa sobre quinta geração de computadores que está sendo desenvolvido pelos japoneses; também está sendo utilizada nos programas de Inteligência Artificial que estão sendo desenvolvidos por ingleses e franceses.

Para programar em PROLOG são acompanhados os passos seguintes:

1.º - Especificar alguns fatos e relações.
2.º - Especificar algumas regras sobre ditos fatos e relações.
3.º - Fazer perguntas sobre os mesmos.

Por exemplo:

```
Ama (Carlos Ana)
Ama (Pedro Julia)
Ama (Gustavo Marcia)
```

Agora, se perguntamos:

```
    Ama (Pedro Julia)
```

PROLOG responderá

```
    SIM.
```

Neste exemplo tão trivial, "Ama" é um predicado que indica uma relação entre Carlos e Ana.

Ainda que PROLOG não incorpora toda a lógica que é necessária no cálculo de predicados, sua sintaxe é muito menos complexa que a maioria das linguagens normais de potência comparável. Uma linguagem não pode ser estritamente lógica: necessita alguns códigos básicos que controlem os aspectos de procedimento das operações; isto, e no grau mínimo, é o que incorpora Prolog. Desta forma, em PROLOG se diz ao computador em uma linguagem declarativa "o que" tem que fazer e ele se encarregará de "como" fazê-lo. Este é o aspecto mais sobressalente que incorpora PROLOG e o que o diferencia das demais linguagens de programação.

POPLOG

É uma combinação de LISP, PROLOG e POP-11 introduzidas em um mesmo pacote; quando se compila é mais rápido que LISP ou PROLOG incorporando além disso as vantagens de ambos os sistemas. É o sistema que está sendo desenvolvido pela IBM para suas aplicações internas

Sistemas esqueleto ou padrões de programação

Sobre um sistema experto já construído e bem provado é recolhido o maior número de elementos compatíveis com outros, formando assim um sistema esqueleto. Por exemplo, a forma de representar o conhecimento, o controle das regras, a inferência, etc., representam o esqueleto do antigo sistema experto.

Quando os criadores de um sistema experto pioneiro sobre diagnóstico médico (MYCIN) acabaram seu trabalho, deram conta que o podiam separar em duas partes diferenciadas: o banco de conhecimentos, onde era contida toda a informação sobre enfermidades infecciosas e que era específico desta aplicação, e o motor de inferência, um controle "backward chaining" de propósito geral que bem podia ser utilizado por outras pessoas para criar outro sistema experto de parecidas características, porém sobre um tema completamente distinto (Fig. 6).

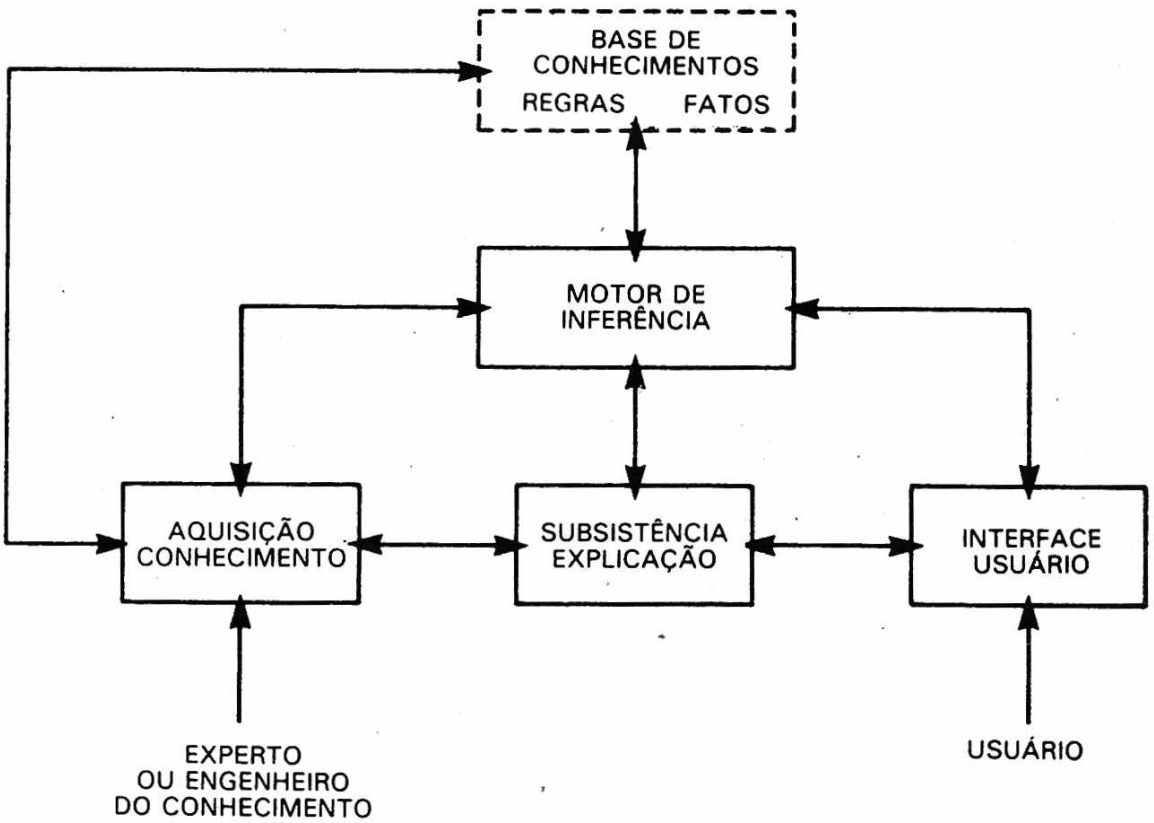


Figura 6. — Arquitetura de um sistema esqueleto: inclui o motor de inferência e os subsistemas de aquisição, explicação e interface com o usuário.

EMYCIN é uma parte de MYCIN que recolhe tudo exceto o banco de conhecimentos, porém que contém tudo o necessário para raciocinar sobre um banco de conhecimentos e responder às consultas do usuário. Não é, portanto, uma linguagem de propósito geral que se ajusta completamente aos requerimentos do sistema experto, mas que ao utilizar regras O-A-V, usar a lógica modus ponens e o motor de inferência "backward chaining" somen-

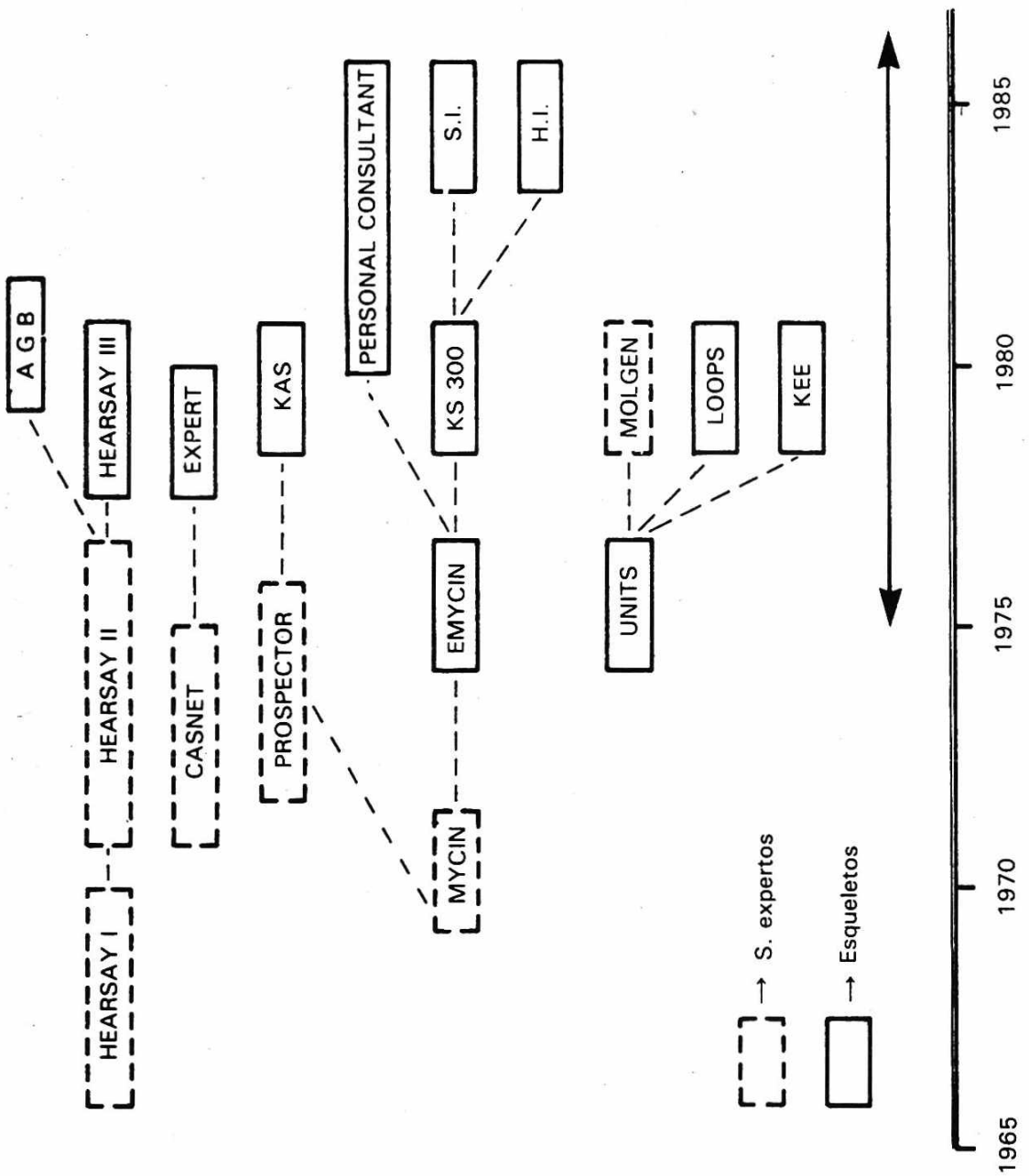


Figura 7. — Desenvolvimento ao longo do tempo dos sistemas expertos e os sistemas esqueletos a que deram lugar.

te pode ser adaptado a um tipo concreto de sistemas expertos, os de diagnóstico fundamentalmente.

Quase todos os sistemas esqueletos ou padrões de programação foram aparecendo associados a um sistema experto concreto, ao qual foi suprimida a parte do banco de conhecimentos. A figura 7 representa a evolução de diversos sistemas em padrão de programação e seu desenvolvimento ao longo dos últimos anos. Como foi observado, o utilizar parte de algo já criado é uma boa solução, mas certamente, não está isenta de certos problemas, sobretudo, se quisermos aplicar em tarefas distintas às concebidas para o sistema original. Este seria o caso de utilizar EMYCIN (diagnose fundamentalmente) para criar um sistema experto de ajuda à planificação (EGERIA).

Os principais problemas são:

— A armação do antigo sistema pode ser imprópria para a nova tarefa a realizar.
— A estrutura de controle pode não encaixar perfeitamente na forma de realizar inferências que se deseja.
— A linguagem antiga pode ser imprópria.
— Pode existir no sistema um conhecimento prévio sobre o controle que não se necessite.

Sistemas conchas ou ferramentas

São sistemas desenvolvidos a partir de linguagens de propósito geral que proporcionam maior potência para a criação dos sistemas expertos. Incorporam muitas facilidades sobre inferência e controle e agilizam enormemente a criação do sistema.

Os expertos em Inteligência Artificial, ou Engenheiros do Conhecimento, estudando o tipo de problemas aos que mais freqüentemente se enfrentam os profissionais de diversos campos identificaram três estratégias de resolução, bem definidas e claramente diferenciadas. São a diagnose ou prescrição, a planificação e o desenho. Na figura 8 podem ser observadas as características de cada estratégia e o tipo de problemas aos quais se enfrenta. Uma vez que os pioneiros definiram estes três tipos, os sistemas conchas que foram construídos ou estão sendo construídos são desenhados fundamentalmente para uma destas aplicações ou, ao máximo, para duas delas e somente permitem o desenvolvimento de sistemas expertos cuja tarefa é encravada dentro dos sistemas resolúveis por ditas estratégias.

Certas técnicas de representação do conhecimento (de infe-

Tipo de problema:	Diagnose/Prescrição	Planificação	Desenho
exemplo	Diagnóstico de enfermidades infecciosas e prescrição de medicamentos	Planificar um projeto. Configurar um computador	Desenho de um circuito impresso
Entrada	Condições, respostas a perguntas	Fins, limitações, meios	Fins, limitações
Conhecimento	Uso de regras e cálculos simples	Heurística, modelos, restrições	Heurística, modelos, restrições
Saída	Soluções heurísticas	Planificação heurística	Descobertas heurísticas

Figura 8. — Tipos normalizados de problemas: a diagnose, a planificação e o desenho.

rência ou as estratégias de controle) são em muitos casos específicas de uma aplicação; por exemplo, para a planificação, o mais normal é utilizar "forward chainning" (encadeamento para frente) ao invés de "backward chainning", mas, certamente, outras, como por exemplo os fatores de certeza (certainty factors) podem ser utilizadas em mais de uma aplicação, estando, portanto, incorporadas na maioria dos sistemas conchas. Em geral problemas de

PADRÃO/ FERRAMENTA	Diagnose/ Prescrição	Planificação	Desenho
EMYCIN	●	○	○
ES/P ADVISOR	●	○	○
Expert. Ease	●	○	○
INSIGHT	●	○	○
M.I.	●	○	○
Personal Consultant	●	○	○
EXPERT	●	○	○
KES	●	○	○
OPS 5	●	●	○
S.I.	●	○	○
TIMM	●	○	○
ART	●	○	○
KEE	●	●	○
LOOPS	●	○	○

● ○

Figura 9. — Ferramentas comerciais mais usuais e tipos de problemas aos quais se enfrentam.

um tipo particular levam associadas ferramentas construídas com uma determinada forma de representação, inferência e controle.

Os problemas ou paradigmas mais fáceis de resolver são os de Diagnose e prescrição e atualmente a maior parte dos sistemas concha comerciais (Fig. 9) estão preparados para estas aplicações; unicamente os mais complexos são capazes de atacar os problemas de planificação e, hoje em dia, ferramentas comerciais que ajudem a criar sistemas inteligentes em desenho não existem.

Comercialmente os sistemas concha são divididos em três categorias:

	CONHECIMENTOS									
	FATOS			REGRAS LÓGICAS					INCER-TEZAS	
● Função presente ◐ Presença restr. ○ Não presente, mas program.	Dupla A-V	Terno O-A-U	FRAMES	Regras IF-THEN	Regras variáveis	Exemplos	Múltiplo-objects	Inheritance	Coeficiente de certeza	Probabilidade
Pequenas ferramentas										
H.I.	●	○		●	●		○	◐	●	○
Personal Consultant		●		●			●	◐	●	
Grandes sistemas restringidos										
S.I.		●		●			●	◐	●	
KES	●			●			●	●		●
Grandes sistemas										
KEE	○	○	◐	◐			◐	◐	○	○
LOOPS	○	○	◐	◐			◐	◐	○	○

Figura 10.a). — Características dos sistemas concha mais comuns em função da representação do conhecimento.

- Ferramentas para pequenos sistemas. Podem trabalhar sobre computadores pessoais. São desenhadas para facilitar o desenvolvimento de sistemas que contenham menos de 400 regras.
- Ferramentas para grandes sistemas específicos. Podem trabalhar sobre máquinas LISP (computadores especiais) ou em grandes computadores e estão desenhadas para criar sistemas expertos que contenham entre 100 e vários milhares de regras, ainda que estejam restringidos para a resolução de um único tipo de problema.
- Ferramentas para grandes sistemas. Podem trabalhar sobre máquinas LISP ou em grandes computadores e estão

	INFERÊNCIA										
	Geração de novos fatos				Estratégias de controle						
● Função presente ◐ Presença restringida ○ Não presente, porém programável											
Pequenas ferramentas											
M.I.		●			●	◐	●				
Personal Consultant		●			●	◐	●				●
Grandes sistemas restringidos											
S.I.		●			●		●		●		●
KES					◐				●		
Grandes sistemas											
KES		○			○	○	○	○	○	○	○
LOOPS		○			○	○	○	○	○	○	○

Figura 10.b). — Características dos sistemas concha mais comuns em função dos motores de inferência.

desenhadas para criar sistemas que contenham entre 500 e vários milhares de regras; podem incluir as características necessárias para a resolução de distintos tipos de problemas.

Na figura 10 apresentamos uma vista geral das diferenças que apresentam vários sistemas conchas pertencentes às três categorias anteriores; pode ser apreciado como ao ir aumentando a potência aumentam as possibilidades de poder utilizar o conjunto de representação do conhecimento, inferência e controle que melhor se adapta ao problema.

CAPÍTULO VII

CRIAÇÃO DE UM SISTEMA EXPERTO

Através deste capítulo tentaremos dar uma idéia do processo real que é seguido na construção de um sistema experto pequeno, baseado na utilização tanto das técnicas de inteligência artificial como das ferramentas que foram colocadas ao serviço dos engenheiros do conhecimento.

A aplicação e o tipo de conhecimento devem ser ajustados aos modelos já estudados de forma que sua análise seja menos complexa; problemas cuja resolução implique embaraçosos cálculos ou processos iterativos terão provavelmente melhor solução na já muita provada programação clássica, e o custo, por outra parte, será sensivelmente inferior.

Com o problema já delimitado, os seguintes passos serão encaminhados para determinar o tipo apropriado de ferramenta necessária para atacá-los, desde a utilização de linguagens de propósito geral como LISP ou PROLOG aos sistemas conchas, passando pelos sistemas esqueletos ou padrões. A escolha de um ou outro não somente depende de aspectos técnicos, mas também de recursos humanos e em último termo, ainda que são ao final os que sobressaem, os recursos econômicos.

A última etapa do desenvolvimento passa forçosamente pela construção de um modelo reduzido do sistema que se quer implantar; com dito modelo a escala será experimentada e resolverão as primeiras dificuldades, servindo de campo de provas do desenvolvimento final. Uma vez suficientemente provado vão sendo aumentados seus préstimos, e portanto sua complexidade, e sofrerá diversos processos de realimentação e melhora até sua instalação final.

Até agora foram consideradas sinônimas as acepções "sistemas expertos" e "sistemas baseados no conhecimento" (Knowledge systems), porém nos sistemas pequenos isto não é de tudo certo. Os gráficos da figura 1, que relacionam o número de pessoas que conhecem um tema com o que sabe cada uma dessas pessoas

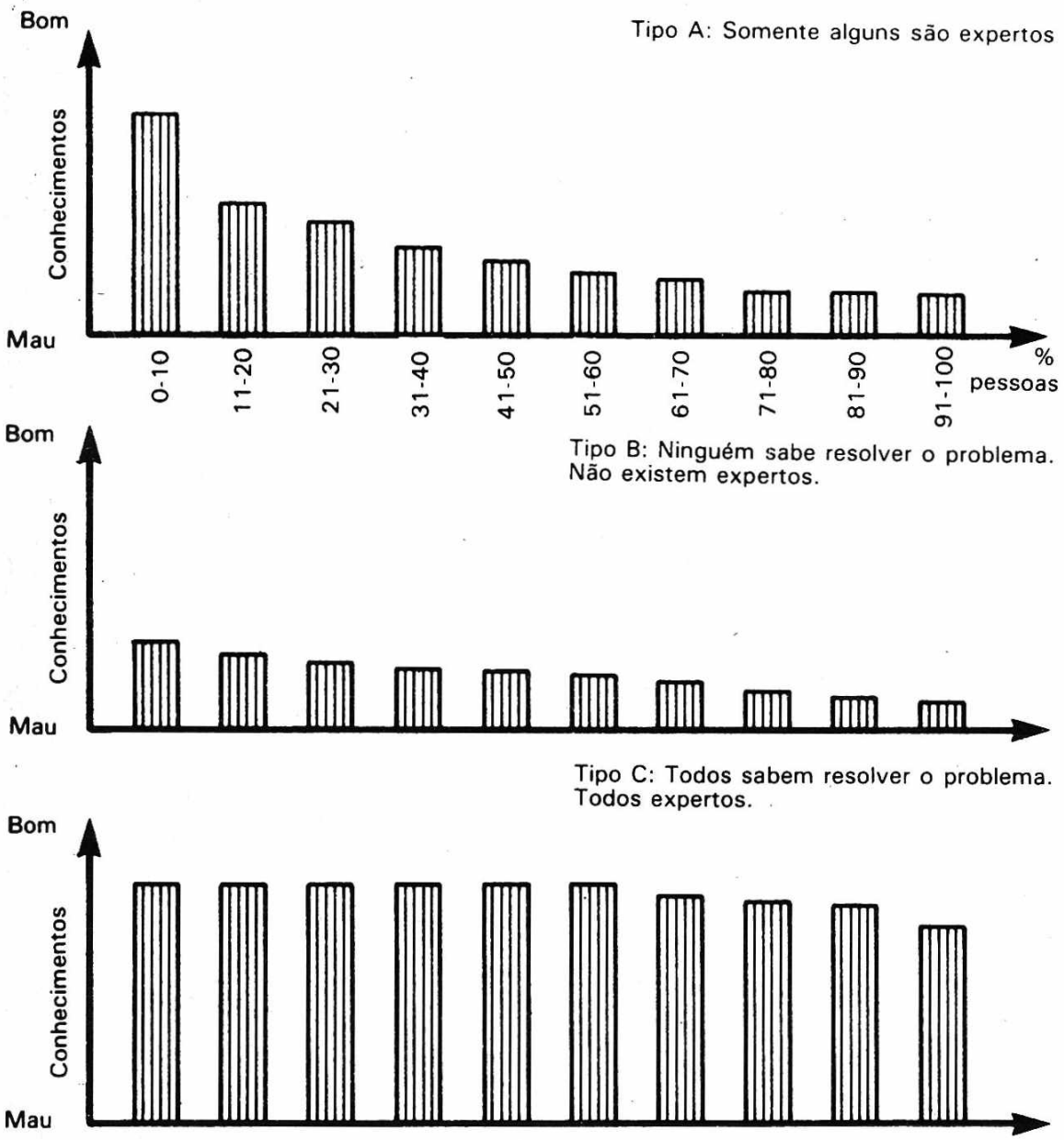


Figura 1. — Diferentes tipos de "profundidade de conhecimento" em um problema em função do número de expertos que conhecem dito problema.

sobre dito tema em três situações distintas, mostra como em algumas de ditas situações ou não existe o experto ou todos são expertos. Isto é, aqueles sistemas expertos que contenham o conhecimento dos problemas do terceiro tipo da figura 1 não modelam realmente ao experto humano, mas somente contém um tipo de conhecimento.

Se é construído um sistema experto que ajude a decidir o tipo dos seguros, a quantia dos mesmos e os bens que cobre, não pode ser dito que o conhecimento que possui, necessário para realizar estas funções, seja o de um experto, mas bem melhor se denominado sistema baseado no conhecimento. Por outra parte estão, por exemplo, os grandes programas de inteligência artificial como DENTRAL e MYCIN que assessoram e resolvem casos médicos muito complexos e onde realmente foi necessário recorrer a verdadeiros especialistas para poder introduzir, em alguns quantos milhares de regras e fatos, todo o saber que possui a humanidade em ditos campos científicos. Estes últimos programas sim podem ser denominados "sistemas expertos".

Os sistemas expertos "pequenos" têm uma concepção distinta à dos grandes sistemas; as pessoas que utilizam estes programas não têm por que ser engenheiros de conhecimento, mas normalmente, serão pessoas próximas ao problema. No exemplo anterior (aquele que ajudava na determinação do seguro mais apropriado) o supervisor do programa deverá ser o agente de seguros e, portanto, todo pequeno sistema experto será eficiente e terá êxito se é verdadeiramente simples em sua utilização.

Como conclusão: os sistemas pequenos são normalmente sistemas baseados no conhecimento e sua característica fundamental é a simplicidade de programação, manejo e modificação.

Passos a seguir

Na hora de decidir-nos pelo tema de um sistema experto pequeno poderíamos ter optado por alguns muito complexos, com os quais pudesse ser vislumbrada a potência destes programas, mas em benefício de um maior esclarecimento optamos por um tema de conhecimento geral como e a escolha de viagem em uma agência de turismo. É na realidade um sistema baseado no conhecimento, pois não necessita um experto em sua realização e resolve problemas que qualquer um pode solucionar.

Para sua construção os engenheiros do conhecimento devem acompanhar os seguintes passos:

1. Selecionar uma ferramenta e ter em mente o tipo de estratégia capaz de enfrentar o problema.
2. Identificar o problema e analisar o conhecimento que será incluído no sistema.
3. Desenhar o sistema. Inicialmente implica construir um com lápis e papel mediante a utilização de diagramas de blocos, matrizes e pequenos conjuntos de regras.
4. Desenvolver com o sistema concha selecionado um protótipo. Escrever o banco de conhecimentos e prová-lo em um número amplo de casos.
5. Ampliar e modificar o programa até que funcione tal e como queremos que o faça.
6. Manter e atualizar o sistema conforme se necessite.

Passo 1: Seleção da ferramenta

As ferramentas (ou sistemas conchas) pequenas são menos restritivas em suas aplicações que as maiores. Se um programa necessita somente 100 regras seu programador não requer uma série de características que exigiria se dito programa constasse de 1.000 regras, pelo simples motivo que 100 regras são relativamente fáceis de controlar com a cabeça, enquanto que se são 1.000 não existe pessoa que possa trabalhar com elas.

Por esta razão todas as ferramentas pequenas se adaptam bem e facilmente a qualquer problema, sempre que pertençam ao tipo de diagnose e prescrição. Neste caso podem ser utilizadas quaisquer das duas ferramentas exemplo do capítulo anterior (ver fig. 10 do capítulo 6); M.1 o Personal Consultant. Ambas representam o conhecimento mediante regras IF-THEN, e seu motor de inferência está baseado em "Backward chainning" (encadeamento para trás). A ferramenta que usaremos neste exemplo é M.1, vendida pela Teknowledge Inc. Ainda que é muito cara, está bem desenhada e é muito fácil de usar. Admite toda classe de problemas que impliquem consultas do tipo diagnose/solução ou prescrição e possui algumas funções que a tornam muito apta para ser manejada por toda classe de usuários.

Uma vez decididos por M.1 e o tipo de estratégia de resolução que isto implica, deve ser confirmado que nosso problema reune as seguintes características:

- O tempo que devemos demorar para resolvê-lo deve estar normalmente entre os 15 e 30 minutos. Se é menos de 10 o problema é muito simples, e se demora mais de 30 im-

plica muito conhecimento e será muito lento.
- Não deve ter que examinar diagramas ou qualquer contato físico; será tal que possa ser normalmente resolvido através de uma conversação telefônica:
- Certamente o processo de resolução poderá ser levado a cabo mediante uma série de regras e, como muito, alguns poucos cálculos. Se a solução requer muitos cálculos é melhor a programação tradicional como, por exemplo, uma planilha de cálculo eletrônico.
- O conjunto de possíveis soluções finais não deve sobrepassar algumas poucas dezenas. Se ditas soluções são mais de 50 haverá, provavelmente, outras ferramentas mais aptas que as utilizadas por I.A.

O conjunto destas normas reduz consideravelmente o número de possíveis temas a considerar. No caso da agência de turismo, o cliente costuma explicar seu caso e escolher a viagem em uns 20 minutos e não são precisos muitos cálculos matemáticos, talvez algumas somas e multiplicações para conhecer os preços exatos aplicando diversas tarifas. Através do telefone é possível contar tudo o referente à viagem e o conjunto de soluções básicas não sobrepassa a uma dezena e meia. Se bem que é certo que dentro de cada solução básica (por exemplo, apartamento na praia) existem muitas variantes, esta é a que em princípio se buscava, e delimitará enormemente o problema.

Passo 2: Identificação e representação do conhecimento

Consiste na explicitação de cada um dos fatores que concorrem na solução do problema, ou melhor, é o conjunto de dados iniciais que deve conhecer o agente de turismo para poder decidir o tipo de viagem que mais convém a cada situação.

Os fatores que concorrem no problema serão limitados inicialmente a seis: motivo, data de viagem, dinheiro, número de pessoas, lugar de residência e idade.

- Motivo. Qual é o motivo da viagem?, é um dos fatores mais destacados e pode tomar os seguintes valores discretos:

  Descanso.
  Diversão.
  Desporte.
  Turismo.

- Data da viagem. Temporada na qual pretende realizar a viagem: na hipótese de uma viagem à Espanha somente serão levadas em consideração as estações

  Verão (Junho-Setembro)
  Inverno (Dezembro-Fevereiro)
  Primavera, Outono (Março-Maio, Outubro-Novembro)

- Dinheiro. Quantidade de dinheiro que pensa gastar na viagem de uma forma global:

  Normal (inferior a 75.000 cruzados/pessoa)
  Superior (mais de 75.000 cruzados/pessoa)

- Número de pessoas. Indica de alguma forma a preferência ou necessidade de optar por um tipo de viagem ou outro; é composto de:

  Casais.
  Família.
  Grupos.

- Lugar da residência. Um fator pouco indicativo por um lado, mas muito exclusivo ou limitativo por outro:

  Cidade.
  Campo.
  Litoral.

- Idade. A idade costuma coadjuvar um tipo de viagem distinto. É composta de:

  Jovens (< 30)
  Adultos (30 < idade < 60)
  Anciãos (> 60)

Temos que fazer notar que nem todos os fatores precedentes têm a mesma importância e assim deveriam figurar no sistema experto, porém para facilitar a todos atribuiremos o mesmo valor. O leitor, por outro lado, deve abstrair se do exemplo particular e generalizar mediante o mesmo o conceito de sistemas baseados no conhecimento.

Uma vez determinados os fatores, a regra geral que resulta é da forma:

Sim (1) o motivo é..... *
(2) a data de viagem é ..... *
(3) o dinheiro disponível é ..... *
(4) o número de pessoas é ..... *
(5) o lugar da residência é ..... *
(6) a idade é de ..... *
Então é recomendada..... **

O seguinte passo será preencher estes vazios.

IF: OPÇÕES						THEN
Motivo	Data-viagem	Dinheiro	Pessoas	Residenc.	Idade	SOLUÇÕES
Descanso Diversão Desporte Turismo	Verão Inverno Prim.-outono	Normal Superior	Casal Família Grupo	Cidade Campo Praia	Jovens Adultos Anciãos	
Diversão	Verão	Normal	Família	Campo Cidade	Adultos	1. PRAIA-APART.-ESPANHA
Diversão	Verão	Superior	Casal	Campo Cidade	Adultos	2. PRAIA-HOTEL-ESPANHA
Descanso	Verão-inverno	Normal	Casal Grupo	Cidade	Jovens Adultos	3. MONTANHA-ESPANHA
Desporte	inverno	Normal	Casal Grupo	Cidade Praia	Jovens	4. ESQUI-ESPANHA
Desporte	Inverno	Superior	Casal	Cidade Praia	Jovens	5. ESQUI-EUROPA
Descanso	Inverno	Normal	Casal	Cidade Campo	Adultos Anciãos	6. PRAIA-CANÁRIAS
Turismo	Primavera-outono	Normal	Casal	X	Adultos	7. GRANDES CIDADES
Turismo	X	Superior	Casal	X	Adultos	8. PAÍSES EXÓTICOS
Turismo	Primavera-outono	Normal	Casal	X	Adultos	9. TOUR-AUTOBUS
Descanso	Primavera-outono	Superior	Casal	X	Adultos Anciãos	10. CRUZEIRO-MAR

Figura 2. — Matriz de relação Opções do usuário — Soluções propostas para o exemplo da agência de viagens.

Inicialmente o conjunto de soluções é limitado a 10; em seguida, na fase de ampliação e melhora, poder-se-á chegar até um número aproximado de 40. Com estas primeiras 10 soluções pode ser criada a matriz da figura 2, onde nas abcissas situamos as condições ou fatores de entrada e nas ordenadas as soluções recomendadas.

M.1, como trabalha com "backward chainning", requer um fim do qual partir; este é "viagem proposta" e a indicamos com:

```
Goal = viagem-proposta
```

Com isto lhe dizemos que tem que determinar a viagem-proposta e ele procurará sempre chegar a isto. Uma vez completada a matriz da figura 2 é finalizada a análise do conhecimento; a primeira regra pode ser:

```
if        Motivo = Diversão          and
     Data-viagem = Verão             and
        Dinheiro = Normal            and
        Pessoas = Família            and

     Lugar-resid. = Cidade           or
     Lugar-resid. = Campo            and
           Idade = Adultos

Then        viagem-proposta = Praia-Apartamento-Espanha
```

Para ilustrar como trabalha M.1 e as facilidades de comunicação com o usuário que apresenta, podemos construir um protótipo com esta única regra. Primeiramente, e trabalhando com um IBM-PC e um processador de textos como pode ser Wordstar ou qualquer outro, criamos um arquivo onde escrevemos o "Goal" e a regra número 1. A forma de escrita é idêntica a como foi realizado acima. Fechamos o arquivo e vamos ao programa principal M.1, desde onde chamamos ao arquivo criado anteriormente.

M.1 começa comprovando o fim (Goal). Observa se em sua memória ativa consta este dado; se não o encontra prossegue a execução do programa buscando alguma regra que conclua com viagem-proposta. Como encontra a que acabamos de introduzir começa a checar as cláusulas "if" ou antecedentes de dita regra. O primeiro antecedente é "Motivo". Busca em sua memória de tra-

balho a informação desse campo; como não a encontra olha se alguma regra conclui com um conseqüente "Motivo", o que não acontece, já que o arquivo contém uma única regra.

M.1 resolve o conflito da única maneira possível: perguntando pelo dado que lhe faz falta:

What is the value of: Motivo?
(Qual é o valor de "Motivo"?)

Podemos responder o que desejarmos, mas se a resposta não é "Diversão" receberemos uma mensagem de M.1 dizendo:

viagem-proposta was songht, but no value was concluded.
(procurei a solução mas não encontrei nenhuma)

Se respondermos com "Diversão" e a todos os outros campos corretamente a solução será:

viagem-proposta = Praia-Apartamento-Espanha

Este simples programa, qualquer pessoa que conheça alguma linguagem de programação poderia realizá-lo facilmente, porém seria muito mais difícil criar um que pudesse trabalhar com qualquer número de regras, fizesse as perguntas apropriadas e chegasse à solução corretamente.

O protótipo completo será composto das seguintes regras, algumas das quais são incorporadas para tornar o programa mais atraente face ao usuário:

regra 1	if	Motivo = Tranqüilidade	or
		Motivo = Paz	or
		Motivo = Sossego	or
		Motivo = Descansar	or
	then	Motivo = Descanso	
regra 2	if	Motivo = Esquiar	or
		Motivo = Jogar tênis	
	then	Motivo = Desporte	

Estas regras permitem ao usuário responder de forma mais natural às perguntas do computador, que, por outro lado, podem ser estruturadas de forma distinta à qual levava inicialmente; assim quando lhe faltar o dado do fator "Motivo" perguntará Questão (Motivo) – "Qual é o motivo de suas férias?"

Da mesma forma o usuário dispõe em todo momento das

possíveis respostas a cada pergunta:

legalvals(motivo) = tranqüilidade, paz, sossego, descansar, esquiar, jogar tênis, desporte, diversão, turismo.

regra 3	if	Data = Outono	
		Data = Primavera	
	then	Data-viagem = Primavera-Outono	
regra 4	if	Data = Junho	or
		Data = Julho	or
		Data = Agosto	or
		Data = Setembro	
	then	Data-viagem = Inverno	
regra 5	if	Data = Dezembro	or
		Data = Janeiro	or
		Data = Fevereiro	
	then	Data-viagem = Inverno	
regra 6	if	Gastos = 75.000 cruzados/pessoa	
	then	Dinheiro = Normal	
regra 7	if	Gastos > 75.000 cruzados/pessoa	
	then	Dinheiro = Superior	
regra 8	if	Idades < 30	
	then	Idade = Jovem	
regra 9	if	Idades > 30	and
		Idades < 60	
	then	Idade = Adulto	
regra 10	if	Idades > 60	
	then	Idade = Ancião	
regra 11	if	Motivo = Diversão	and
		Data-viagem = Verão	and
		Dinheiro = Normal	and
		Pessoas = Família	and
		Lugar-res = Campo	or
		Lugar-res = Cidade	and
		Idade = Adultos	
	then	Viagem-proposta = Praia-Apartamento-Espanha	
regra 12	if	Motivo = Diversão	and
		Data-viagem = Verão	and
		Dinheiro = Superior	and
		Pessoas = Casal	and
		Lugar-res = Campo	or
		Lugar-res = Cidade	and

		Idade = Adultos	
	then	Viagem-proposta = Praia-Hotel-Espanha	
regra 13	if	Motivo = Descanso	and
		Data-viagem = Verão	or
		Data-viagem = Inverno	and
		Dinheiro = Normal	and
		Pessoas = Casal	or
		Pessoas = Grupo	and
		Lugar-res = Cidade	and
		Idade = Jovens	or
		Idade = Adultos	
	then	Viagem-proposta = Montanha-Espanha	
regra 14	if	Motivo = Desporte	and
		Data-viagem = Inverno	and
		Dinheiro = Normal	and
		Pessoas = Casal	or
		Pessoas = Grupo	and
		Lugar-res = Cidade	or
		Lugar-res = Praia	and
		Idade = Jovens	
	then	Viagem-proposta = Esqui-Espanha	
regra 15	if	Motivo = Desporte	and
		Data-viagem = Inverno	and
		Dinheiro = Superior	and
		Pessoas = Grupo	and
		Lugar-res = Cidade	or
		Lugar-res = Costa	and
		Idade = Jovens	
	then	Viagem-proposta = Esqui-Europa	
regra 16	if	Motivo = Descanso	and
		Data-viagem = Inverno	and
		Dinheiro = Normal	and
		Pessoas = Casal	and
		Lugar-res = Cidade	or
		Lugar-res = Campo	and
		Idade = Anciãos	or
		Idade = Adultos	
	then	Viagem-proposta = Praia-Canárias	
regra 17	if	Motivo = Turismo	and
		Data-viagem = Prim/Out	and
		Dinheiro = Normal	and
		Pessoas = Casal	and
		Idade = Adultos	
	then	Viagem-proposta = Visita-grandes-cidades	

regra 18	if	Motivo = Turismo	and
		Dinheiro = Superior	and
		Pessoas = Casal	and
		Idade = Adultos	
	then	Viagem-proposta = Países-exóticos	
regra 19	if	Motivo = Turismo	and
		Data-viagem = Prim/out	and
		Dinheiro = Normal	and
		Pessoas = Casal	and
		Idade = Adultos	
	then	Viagem-proposta = Tour-autobus	
regra 20	if	Motivo = Descanso	and
		Data-viagem = Prim/Out	and
		Dinheiro = Superior	and
		Pessoas = Casal	and
		Idade = Adultos	or
		Idade = Anciãos	
	then	Viagem-proposta = Cruzeiro-mar	

Passo 4: Desenvolvimento com o sistema concha de um protótipo

Chegados a este ponto somente temos que escrever o banco de conhecimentos anterior e o sistema concha M.1 é encarregado do resto: maneja as regras e escolhe as que têm que aplicar em cada caso, pergunta quando desconhece algum fator e é capaz de mostrar-nos como chegou a essa solução ou por que, em um momento dado, faz uma pergunta determinada.

Estas duas últimas características são de grande ajuda e importância para o encarregado de colocar em funcionamento o programa, facilitando-lhe a depuração e eliminação de erros do sistema.

Passo 5: Ampliação e modificação

A facilidade de ampliar um banco de conhecimento deve ser uma das características mais importantes dos sistemas conchas. Em nosso caso podem ser escritas até 200 regras na base de M.1, com o qual poderiam ser completados todos os casos que não foram introduzidos no protótipo.

No exemplo da agência de viagens foi concedido um fator de importância, ou de verossimilitude, ao resultado de cada regra de

1.0, o que não é em todos os casos certo; deverão ser introduzidos em cada caso os fatores de certeza correspondentes.

Passo 6: Atualização e revisão

É somente uma extensão do passo 5 e consiste em adaptar ao banco de conhecimento às situações mutantes que contribuem no sistema experto.

Ainda que o exemplo seja realmente simples, com ele pudemos dar uma idéia do sistema utilizado nestes pequenos sistemas baseados no conhecimento.

CAPÍTULO VIII

FUTURO DOS SISTEMAS EXPERTOS

É muito difícil tentar descrever qual será o futuro de um ramo da ciência que é caracterizado por sua constante evolução e pela facilidade com que mudam seus fundamentos.

No entanto, ainda que seja comentado grandes tendências, sem que possa ser orientado para onde apontam os projetos de pesquisa que são realizados atualmente e que, seguramente, representam o futuro de todo o setor.

Atualmente nos Estados Unidos foram definidas duas tendências muito diferentes sobre como interpretar a filosofia de utilidazação e desenvolvimento dos sistemas expertos.

Por um lado são encontrados os defensores do formalismo e a lógica como pilares de seu desenvolvimento.

São cientistas que pretendem compreender o mundo mediante a lógica. Tentam formalizar tudo o que acontece na vida. Os limites dos sistemas expertos, neste sentido, encontram-se onde a lógica resulte insuficiente para representar a realidade.

Isto é: se fosse possível representar matematicamente o fato "o leitor está comendo um pudim enquanto lê o livro"; este poderia ser introduzido no computador e, portanto, manejado como informação conhecida. O paradigma destes cientistas poderia ser: "TODO INÍCIO DE ESTUDO DEVE COMEÇAR PELA SINTAXE".

Esta tendência encontra-se estabelecida nas universidades da costa oeste americana como Standorf.

Por outro lado são encontrados os cientistas que propõem começar "entendendo" o mundo antes de tentar formalizá-lo. Eles

opinam que ainda que resulte muito complicado entender os processos, uma vez conseguida esta compreensão é mais fácil modelar as relações em uma linguagem formal.

Para estes pensadores o importante é o significado, a SEMÂNTICA, antes que a sintaxe. Assim, um passo anterior ao tentar representar que "o leitor está comendo um pudim enquanto lê o livro" seria compreender realmente o que isto significa e tentar introduzir esta informação na representação utilizada.

Esta tendência é encontrada sobretudo nas universidades da costa leste americana, como, por exemplo, o M.I.T (Massachusetts Institute of Technology) ou YALE.

Como paradigma poderia ser concluído: "O QUE INTERESSA SÃO OS CONCEITOS".

Na hora de condensar as tendências, Tom Alexander as define como os "maltrapilhos" (leste) e os "elegantes" (oeste).

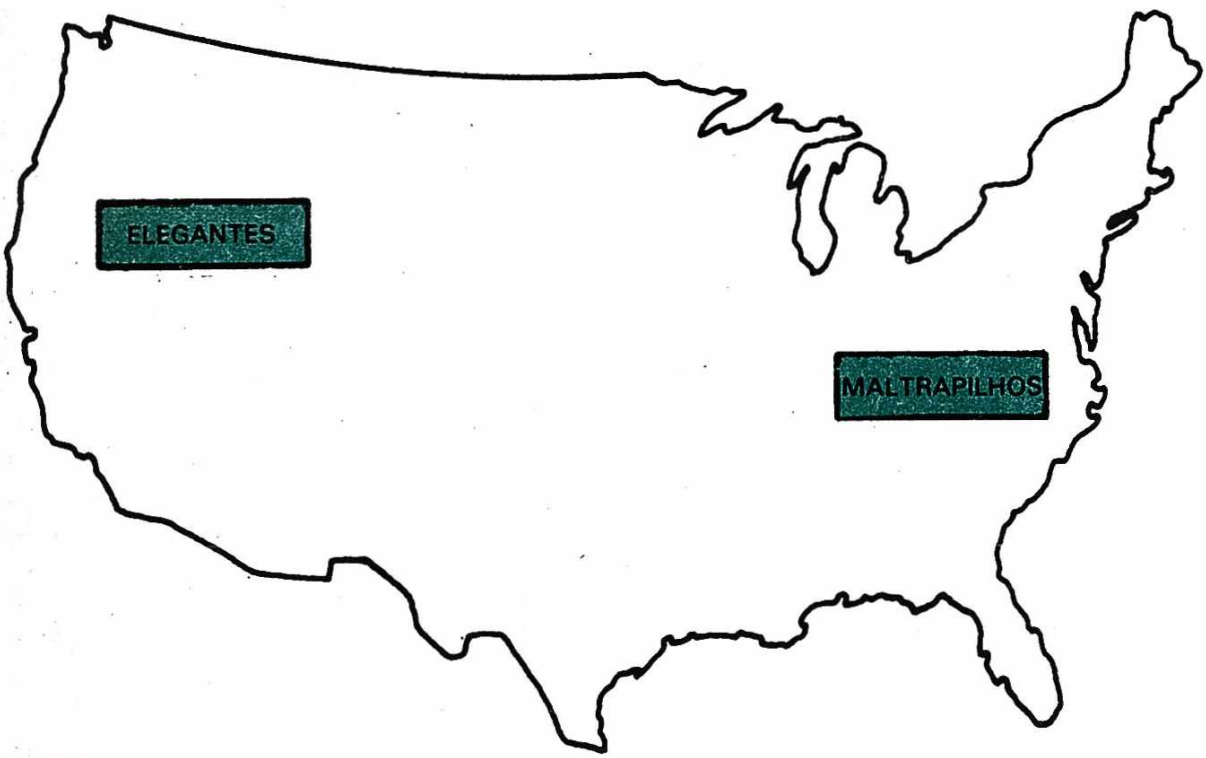


Figura 1. — Assentamento das correntes de pensamento americanas no que diz respeito aos sistemas expertos.

A quinta geração dos computadores

Referente ao futuro dos sistemas expertos teríamos que comentar, sem falta, o famoso projeto QUINTA GERAÇÃO DE COMPUTADORES.

Em outubro de 1981 o Japão anunciou que, mediante a criação de um projeto informático que teria uma duração de 10 anos, pretendia criar um novo modelo de computador que conseguisse revolucionar o mundo do tratamento da informação.

Pese a que não foram conseguidos todavia todos os resultados esperados, o projeto está obtendo alguns avanços significativos nessa área e, além disso, conseguiu que outros países se interessassem o suficiente pelo tema como para iniciar suas próprias pesquisas.

Analisemos brevemente o que se pretende conseguir no projeto de quinta geração, passo a passo.

- RAPIDEZ. Se pretende conseguir que os computadores funcionem mais rapidamente, isto é, que sejam capazes de processar mais instruções e informação que os computadores atuais no mesmo tempo.

Isto é tentado mediante alguns CHIPS mais rápidos com capacidade de processamento, isto é, de realizar operações básicas.

- TRABALHO COM LÓGICA SIMBÓLICA. Atualmente, os computadores funcionam com estruturas fundamentalmente numéricas. Ainda que também manejam símbolos distintos dos números, todavia resulta uma miscelânea a utilização de letras e caracteres não numéricos em programas de todo tipo.

Se pretende, com este impulso, criar um sofware (estruturas de programas, programas em geral) capaz de manejar no futuro qualquer tipo de informação desta maneira ótima.

Isto permitiria uma melhora substancial no processamento do conhecimento, tema que está diretamente relacionado com os sistemas expertos.

- UTILIZAÇÃO DA LINGUAGEM NATURAL. Certamente, esta possibilidade não será conseguida em 100% até dentro de muitos anos. No entanto, muitos tipos limitados de linguagem natural podem ser conseguidos sem muitos problemas.

O projeto de computadores da quinta geração propõe a consecução desta característica tanto em conceito de compreensão do usuário (que o computador seja capaz de compreender o que o usuário lhe mande em uma linguagem comum), como em geração de mensagens (que o computador seja capaz de gerar a informação que fornece ao usuário utilizando a linguagem do próprio usuário) inclusive de viva voz, isto é, que o computador seja capaz de falar.

- ARQUITETURA TIPO PARALELO. Até agora os computadores trabalham de uma forma seqüencial, isto é, o microprocessador

realiza as tarefas atribuídas uma por uma.

No plano de computadores da quinta geração de pretende conseguir que existam muitos microprocessadores que realizem muitas tarefas de uma só vez, porém sem que dependam sempre de um mesmo "chefe". Isto é muito complicado e custará também muito conseguí-lo. Em qualquer caso, o projeto está supondo e suporá um grande avanço na técnica informática.

CAPÍTULO IX

REFLEXÕES FINAIS

Freqüentemente o obstáculo mais importante que deve superar uma nova tecnologia para sua implantação é encontrada na própria sociedade humana. O homem necessita tempo para adaptar-se a mudanças que variam seus costumes, e mais se estes vem por conta do "futuro". Por si só, a informática estabeleceu uma transformação tão radical da sociedade humana que para muitas pessoas está sendo impossível seguir o rítmo.

Como parte integrante desta revolução informática é encontrada a Inteligência Artificial. De recente conhecimento a nível geral se encontra nestes momentos em plena "digestão social".

Enquanto que no mundo da técnica e da gestão foi acolhida com curiosidade, muitos outros já apresentam os primeiros sintomas de preocupação. Perguntas como "Aonde vamos chegar? estão na ordem do dia.

Mas realmente: Sabemos aonde vamos chegar? Temos segurança de que o problema não nos escapará das mãos? Se é estabelecida a questão com um pouco de visão futurista, nos assustaremos?

Podemos ter a segurança de que os processos industriais, cedo ou tarde (e provavelmente mais cedo que tarde) serão totalmente automatizados. Isto quer dizer que não será necessário pessoal humano para realizá-los nem para controlá-los. Conseguir este antigo sonho humano (que trabalhem outros) estabelece dois grande problemas:

- Pode ocorrer que um "erro" em grande escala possa chegar a paralisar a atividade mundial? Resulta claro o considerar este problema como fictício, precisamente pela enorme capacidade de operação e trabalho autônomo que podem conseguir os computadores.
- Será capaz a sociedade de aceitar uma mudança radical em suas estruturas e a chegada da "sociedade do ócio"? Talvez seja o processo mais difícil de alcançar por sua complicação.

O ser humano deve assumir o que em um princípio é fácil: temos que transformar o homem-trabalhador em homem-cultural.

Do mesmo modo, inerente a todo este processo, é encontrado outro grande risco. Tal e como está sendo levado o desenvolvimento informático atualmente, de maneira anárquica, muita gente delinea o problema de manter intata a liberdade de operação, atuação e pensamento que exigem os direitos naturais do homem. Realmente a impressão geral existente sobre a informatização é uma "perda de liberdade". Isto pode ser enganoso porquanto o que se critica pode ser o "não poder fazer o que não devo".

O delineamento anterior, a perda de liberdade, tem suas raízes na seguinte pergunta: Pode um computador chegar a decidir de forma global sobre algo? O problema dos temas de decisão é importante. Nosso mundo se vangloria de computadores com maior rapidez, capacidade de raciocínio e decisão, talvez sem pesar a necessidade de controlar sua evolução. Seriam necessárias algumas regras básicas da informática, ao estilo das de robótica propostas por Isaac Asimov (*).

Em qualquer caso não é motivo de grande preocupação, mas de estabelecer um simples controle. Enquanto isso, a evolução continua e serão os homens que estudam estes temas os que te-

---

(*) Conhecido escritor de obras de divulgação científica e histórica e de livros de ficção científica. Nestas últimas tem escrito muito freqüentemente o tema dos robôs inteligentes, desenvolvendo as que foram denominadas "Leis da robótica" (ver por exemplo, "Eu robô"), que foram adotadas e utilizadas, inclusive, por outros autores do gênero. Estas três leis são:

1. Um robô não pode danificar a um ser humano, nem por sua inércia permitir que sofra dano.
2. Um robô deve obedecer as ordens que lhe são dadas por um ser humano, exceto quando estas ordens estão em oposição com a primeira lei.
3. Um robô deve proteger sua própria existência até onde esta proteção não entre em conflito com a primeira ou segunda lei.

rão em suas mãos o poder de decidir sobre toda a sociedade. Ainda que não seja este o motivo mais importante para muitos, quanto mais pessoas trabalharem sobre o tema mais poderemos ouvir no futuro. Por isso temos que animar a todos os interessados a que pesquisem, estudem e proponham suas soluções. Tanto em Inteligência Artificial como em outros ramos, neles é encontrado nosso futuro.

BIBLIOGRAFIA

1. Inteligência artificial no Spectrum - Faça o seu microcomputador pensar
   Brain
2. Inteligência artificial no seu Spectrum e Spectrum +
   Hartnell
3. Inteligência artificial em Basic
   James
4. Padrões em programação - Métodos e procedimentos
   Longworth
5. Intelligent machines - An introductory perspective of artificial intelligence and robotics
   Gerarter
6. Artificial intelligence
   Ponomaryov
7. Artificial intelligence
   Rich

Da mesma forma que computadores e pacotes de aplicação ficam rapidamente obsoletos como conseqüência do aparecimento no mercado de outros melhores em um espaço de tempo incrivelmente curto, o mesmo já aconteceu à programação clássica: já no futuro não serão utilizados os programas, mas os sistemas expertos.

Os computadores da quinta geração serão capazes de "falar" sem problemas com o usuário e de programar-se automaticamente a partir tão somente das indicações do usuário sobre o problema a resolver.

Ambas as situações, que já começam a ser realidade nos países mais avançados, são tão somente alguns dos aspectos da chamada inteligência artificial, um novo desafio para o homem do nosso século.