

E N T E N D A O

COMPUTADOR

O DESAFIO DA APRENDIZAGEM



NOVA CULTURAL

E N T E N D A O

COMPUTADOR

O DESAFIO DA APRENDIZAGEM



NOVA CULTURAL

Authorized Portuguese Edition © 1988, Editora
Nova Cultural S.A. Original edition © 1986
Time-Life Books Inc. All rights reserved.

Nenhuma parte desta publicação pode ser reproduzida
por qualquer meio, seja eletrônico ou mecânico,
inclusive por equipamentos ou sistemas de classificação
de informações e pesquisa, sem a prévia autorização da
editora, exceto curtos trechos para resenhas, desde que
com o devido crédito.

Time-Life is a trademark of Time Inc., U.S.A.

Edição organizada pela Editora Nova Cultural Ltda.
Av. Brigadeiro Faria Lima, 2000 — 3º andar
CEP 01452 — São Paulo — SP — Brasil
(Artigo 15 da lei 5988, de 14/12/1973)

Esta obra foi composta na AM Produções Gráficas Ltda.
e impressa na Divisão Gráfica da Editora Abril S.A.

ISBN 85-13-00094-9

5

Conhecimento e analogia

1

23

Aprender: um desafio
A educação de um robô pelo exemplo

2

49

Vendo e prevendo
A fronteira espacial


3

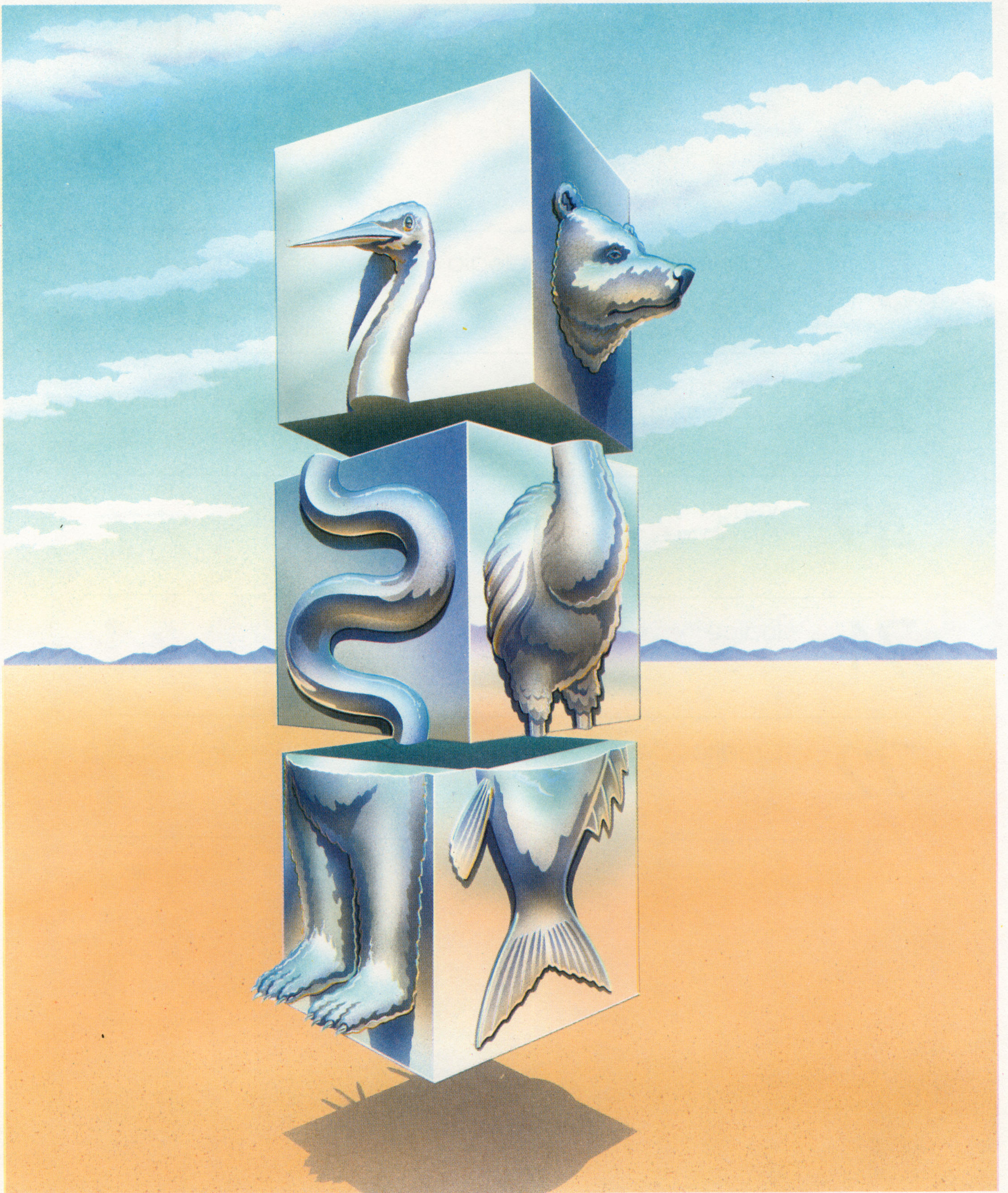
71

Índice

72

Créditos das fotos





Conhecimento e analogia

Em 1949, o eminente matemático norte-americano Warren Weaver escreveu um memorando, que circulou amplamente, onde descrevia o que chamou de “solução de problemas de tradução em escala mundial”. Propôs simular num computador as funções de um tradutor de linguagem. A idéia de Weaver continha algumas dificuldades; por exemplo, ele sugeriu o desenvolvimento de uma “linguagem universal”, que denominou Machineese (“Maquinês”), como um passo intermediário no processo da tradução por máquina. Apesar dessa barreira um tanto intimidante, o plano de Weaver era, em suas linhas gerais, simples e direto. Ao operar uma sentença numa língua, o computador consultaria cada palavra num dicionário bilíngüe automatizado para escolher seu equivalente na segunda língua e então ordenaria as palavras resultantes de acordo com as regras de gramática apropriadas.

A idéia encontrou ressonância entre lingüistas, outros cientistas e engenheiros. O próprio Weaver trazia credenciais impecáveis: era diretor de ciência natural para a fundação Rockefeller. E o computador — embora fosse usualmente considerado naquela época como um processador de números — já se mostrava muito promissor num empreendimento relativamente similar, a decifração de códigos. Durante a Segunda Guerra Mundial, foi inventada, na Inglaterra, a máquina Colossus — um decifrador de códigos, precursor eletrônico de grande escala dos computadores digitais. Parecia que a decifração de outra linguagem era o próximo passo lógico. Nas palavras de Weaver: “Quando olho para um artigo em russo, digo que está na verdade escrito em inglês, mas codificado em alguns símbolos estranhos. Procedo então à sua decodificação”.

Funcionários do Departamento de Defesa dos Estados Unidos também gostaram da idéia. Ansioso por um meio fácil para examinar publicações técnicas em russo para saber em que pé estavam os avanços científicos soviéticos, o Departamento destinou milhões de dólares a pesquisas sobre tradução por máquina. Anthony G. Oettinger, que liderou um projeto patrocinado pelo Pentágono, em Harvard, durante a década de 1950, lembrou mais tarde que os cientistas soviéticos estavam igualmente desejosos de traduções rápidas de publicações norte-americanas.

Os computadores, no entanto, produziam traduções a tal ponto estapafúrdias que certas histórias sobre sua incompetência tornaram-se célebres. Uma delas, provavelmente apócrifa, contava que um pesquisador instruiu seu computador para traduzir o familiar provérbio “o espírito é forte mas a carne é fraca”. Depois que a máquina processou a frase em russo e a reconvertiu ao inglês, obteve-se como resultado: “A vodca é boa mas a carne está estragada”. E houve outros exemplos não menos absurdos. Num dos casos, os pesquisadores que experimentavam a tradução de documentos técnicos, do inglês para o russo e vice-versa, ficaram perplexos pelas referências repetidas a uma certa “cabra d’água” (*water-goat*). Rastreado as referências até a fonte em inglês, descobriram que a máquina apenas tentava lidar com um termo de engenharia, “carneiro hidráulico” (*hydraulic ram*).

Em meados da década de 1960, a maior parte dos financiamentos cessou e os projetos sobre tradução por máquina foram sendo abandonados. Ficou bem claro

Os seres humanos adquirem conhecimentos e trocam informações por intermédio da flexível e extraordinariamente complexa atividade da linguagem natural. Um objetivo fundamental das pesquisas sobre a AI é o de possibilitar aos computadores participarem de diálogos com os seres humanos, dando às máquinas a capacidade de entender, por exemplo, que a palavra “cabeça” pode aplicar-se a um crânio emplumado e provido de bico e também ao crânio robusto de um urso.

que a tradução de uma língua exigia mais que o mero conhecimento das regras da gramática e dos significados das palavras isoladas. Para que uma pessoa — ou um computador — possa traduzir de uma língua para outra, é preciso primeiro que compreenda essa língua.

A tarefa de ensinar computadores a compreender o inglês e outras línguas naturais tornou-se, assim, um dos primeiros e mais frustrantes desafios com que se depararam os pesquisadores da inteligência artificial. A própria definição precisa do que significa “compreender” não era o menor desses problemas. Havia controvérsias em torno de várias questões: que fator — a gramática ou o sentido — é mais importante na compreensão da linguagem? Pode tal compreensão ser expressa por meio de algum sistema de regras ou de lógica formal?

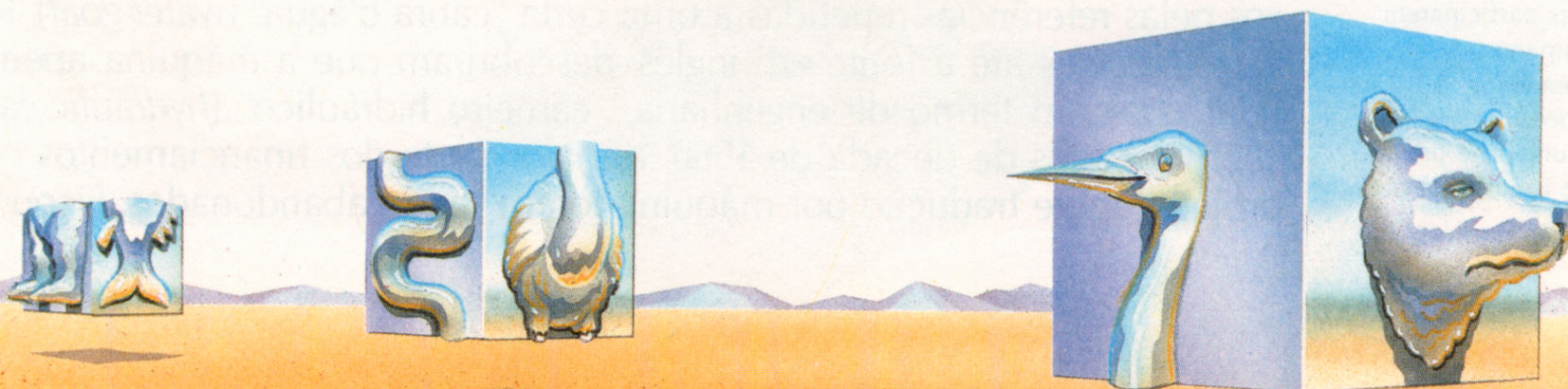
Mais de vinte anos depois, as respostas a essas questões permanecem esquivas. A maioria dos pesquisadores da inteligência artificial (AI) que trabalham no campo da compreensão da linguagem não chega a um acordo nem mesmo quanto aos objetivos de seu trabalho. Alguns pensam que sua meta é eminentemente prática: projetar para o mercado produtos destinados ao entendimento da linguagem. Eles querem planejar programas que facilitem a comunicação com computadores, permitindo aos usuários digitar suas instruções numa língua natural, em vez de fazê-lo usando linguagens de programação, como BASIC ou Pascal. Outros abordam a questão de uma maneira mais teórica. Uma vez que a linguagem e o pensamento parecem entrelaçados, inextricáveis e intimamente ligados com a inteligência, esses pesquisadores acreditam que os esforços para a construção de máquinas capazes de compreender as línguas naturais resultará num novo entendimento sobre como a mente humana funciona — e, por fim, em programas de computadores realmente inteligentes e capazes de competir com ela.

Programas para o entendimento da linguagem são, pois, o foco de grande parte do debate esotérico e, com freqüência, rude, que há anos atormenta a comunidade dos pesquisadores da AI.

ASSENTANDO OS ALICERCES

No entanto, muito antes que se levantasse qualquer uma dessas controvérsias, os pesquisadores da AI tiveram de superar várias dificuldades referentes à utilização de suas máquinas. Dois dos elementos mais importantes para melhorar a interação entre computadores e seres humanos foram o acesso mais fácil à máquina e uma linguagem de programação mais flexível. Ambos foram proporcionados, no final da década de 1950, por John McCarthy, então no MIT, que também cunhou a expressão “inteligência artificial”.

Um dos quatro homens geralmente reconhecidos como criadores da AI (ao lado de Allen Newell, Herbert Simon e Marvin Minsky), McCarthy continua sendo a figura mais extraordinária dessa área, pois desempenhou papéis decisivos na inauguração de dois dos mais proeminentes centros acadêmicos de AI — no MIT, em 1957, e em Stanford, em 1963. Dentre suas muitas contribuições nesse campo, as



mais duradouras podem ser encontradas em seu trabalho para o MIT durante o final da década de 1950. Nesse instituto, em 1959, desenvolveu o conceito de partilha de tempo, que permite a várias pessoas trabalharem simultaneamente em terminais que empregam a potência e a memória de um único computador de grande capacidade. Entre outras aplicações, isso significava que os pesquisadores da AI podiam escrever e testar seus programas experimentais no próprio local em que se encontravam, e na hora, sem ter de esperar às vezes dias para que seus projetos fossem processados em lotes de cartões perfurados por um único operador especializado.

O ADVENTO DE UMA LINGUAGEM FLEXÍVEL

Pouco tempo depois, uma equipe liderada por McCarthy criou uma linguagem de programação conhecida como LISP, sigla de *List Processing* ("Processamento de Listas"). A diferença do FORTRAN, a outra das duas principais linguagens de programação da época, o LISP destinava-se a trabalhar com símbolos não numéricos, especialmente palavras e frases em inglês.

Tudo num programa em LISP é expresso como uma lista — um conjunto de um ou mais elementos, usualmente contidos entre parênteses, que podem por sua vez consistir num único item ou em outras listas. Uma característica-chave dessa linguagem é que tanto dados quanto os programas são, simplesmente, listas de símbolos envolvidos por parênteses, estrutura que permite a um programa manipular outro programa — ou a si mesmo — como um conjunto de dados. Essa característica favorece o tipo de programação, dinâmica e flexível, essencial para quaisquer tentativas de reproduzir o pensamento humano.

O instrumento básico para a manipulação de listas denomina-se procedimento, ou função, e especifica a operação a ser executada. Por exemplo, em FORTRAN, a adição das variáveis X e Y seria representada por $X + Y$. Em LISP, a mesma operação seria escrita assim $(+XY)$, onde o operador (o procedimento de adição é representado pelo sinal $+$) precede os operandos X e Y . Embora essa notação seja deselegante para operações aritméticas simples, ela permite que muitos conceitos matemáticos sejam expressos de maneira sucinta e manipulados com facilidade. Isso também é verdadeiro para conjuntos não numéricos. Por exemplo, com a função (SETQ), um programador pode atribuir um valor a algo sem especificar antecipadamente do que se trata: assim, a sentença (SETQ CHUVA (GET 'ONTEM' TEMPESTADE)) diz ao programa para apanhar (GET) o valor TEMPESTADE de uma lista designada por ONTEM e atribuí-lo como o novo valor do item CHUVA. No LISP, o valor TEMPESTADE pode ser 1/100 de polegada, mas também pode ser a palavra "aguaceiros" — dois diferentes tipos de informação. Para flexibilidade adicional, o LISP permite aos pesquisadores escrever programas experimentais sem ter de pôr de lado, antes do tempo, grandes fatias de memória valiosa só para poder reter informações que muitas vezes acabam ocupando apenas uma pequena porção de memória — ou então extravasam sua capacidade.

Equipadas com a nova linguagem e auxiliadas pela interação imediata com os computadores, proporcionada pela partilha de tempo, as pessoas que trabalhavam na área começaram a produzir programas capazes de articular um diálogo elementar com o operador da máquina. O mais conhecido desses parceiros conversacionais foi criado em meados da década de 1960 por Joseph Weizenbaum, um antigo engenheiro da General Electric que ensinava ciência do computador no MIT.

Weizenbaum batizou-o com o nome de ELIZA porque, como escreveu mais tarde, "à semelhança do que ocorria com Eliza Doolittle, personagem do *Pigmalião* de G. B. Shaw, pode-se ensinar o programa a 'falar' cada vez melhor". No entanto, como o próprio Weizenbaum observou na seqüência "nunca ficou muito claro" se Eliza chegou a ficar mais inteligente ou não.

Originalmente projetado como um sistema para todos os fins, capaz de conversar em inglês sobre qualquer assunto para o qual tivesse sido programado, ELIZA ganhou fama na versão que se tornou conhecida como DOCTOR. Em tal versão, ELIZA simulava as respostas de um psicoterapeuta ao entrevistar um paciente pela primeira vez. Weizenbaum decidiu basear o programa na escola unidirecional da psicoterapia fundada por Carl Rogers, basicamente porque os terapeutas dessa linha tendem a falar muito pouco. É típico entre os rogerianos limitar-se a perguntas breves e observações neutras, visando com isso manter os pacientes falando acerca de seus sentimentos. Como resultado, ELIZA exigia um mínimo conhecimento para ser programada; as próprias observações do paciente forneciam grande parte das informações a partir das quais o programa derivava algo para o médico simulado dizer.

O sabor da sessão terapêutica que daí resultava é maravilhosamente ilustrado por este extrato de um diálogo típico entre uma pesquisadora, que faz o papel do paciente, e ELIZA, que faz o papel do médico (e cujas respostas aparecem em *itálico*):

"Os homens são todos parecidos."

"Em que sentido?"

"Estão sempre nos aborrecendo a respeito de uma coisa ou outra."

"Você pode pensar num exemplo específico?"

"Bem, meu namorado me fez vir até aqui."

"Seu namorado a fez vir até aqui?"

"Ele diz que eu fico deprimida a maior parte do tempo."

"Sinto ter de ouvir que você está deprimida."

"É verdade. Estou infeliz."

"Você pensa que vir aqui ajudará você a não ficar infeliz?"

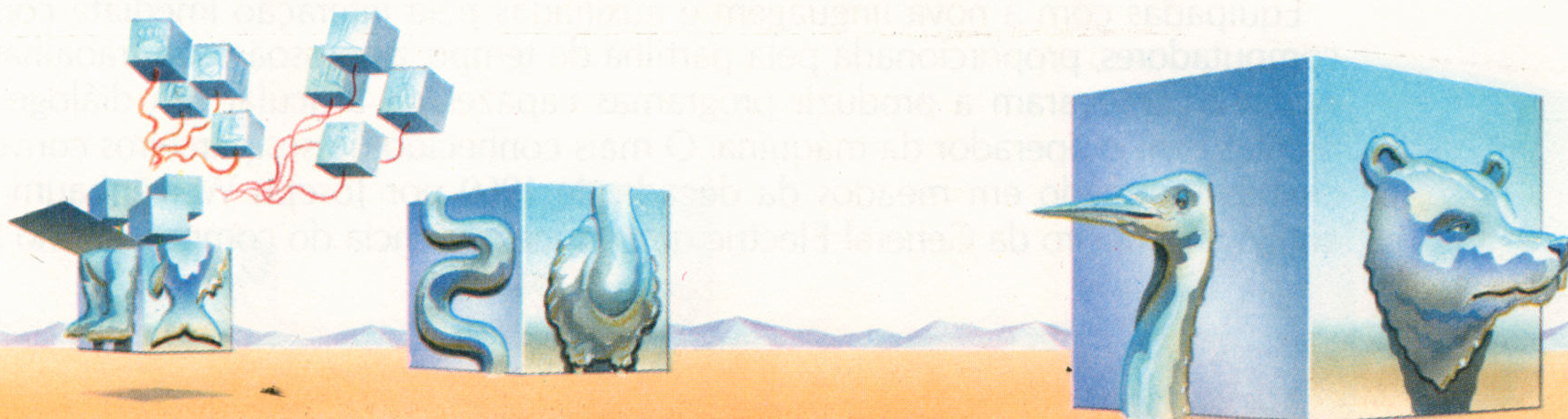
"Preciso de alguma ajuda, parece que não há dúvida quanto a isso."

"O que significaria para você se conseguisse alguma ajuda?"

"Talvez eu pudesse aprender a me relacionar melhor com minha mãe."

"Me fale mais a respeito de sua família."

Embora ELIZA pareça estar entendendo a paciente, isso é em grande parte uma ilusão. A máquina foi programada para reconhecer certos padrões e palavras-chave fornecidos a ela via teclado. Quando o computador descobria um padrão que "batia" com informações residentes no programa, ele respondia com sentenças simples que continham as frases "enlatadas" adequadas. Por exemplo, a referência da paciente à "mãe" levou à pergunta de ELIZA sobre "sua família". Outro dos truques do programa era transformar o comentário do paciente numa pergunta, à



semelhança do que um terapeuta humano também faria em um caso análogo.

Se ELIZA não era capaz de reconhecer um padrão, nem por isso lhe faltavam palavras para prosseguir. O programa recorria ao seu reservatório de perguntas e comentários neutros, tais como "Percebo" ou "Isso é muito interessante" ou "Por que você pensa desse modo?" Instruções no programa impediam o médico de usar duas vezes, numa mesma parte do diálogo, uma determinada frase estocada.

Em resumo, como o próprio Weizenbaum admitiu, ELIZA só era capaz de entender "no mais frágil sentido possível". Um escritor comparou essa compreensão à de um cão doméstico, quando ouve palavras de estímulo familiares, tais como "comida" ou "passear". Mais tarde, Weizenbaum declarou que não teve a intenção de fazer com que ELIZA realmente compreendesse. Escreveu que apenas desejava parodiar um terapeuta rogeriano, com o objetivo de examinar a ilusão de compreender, que freqüentemente emerge nas conversações humanas.

Quaisquer que tenham sido as intenções de Weizenbaum, as respostas preparadas de ELIZA eram tão sedutoramente plausíveis que fascinavam todo tipo de pessoas. De várias maneiras, o programa enganou o vice-presidente de uma firma de pesquisas sobre computadores e fez com que um cientista em computadores soviético, internacionalmente respeitado, sentasse frente a um terminal em Stanford e lhe confiasse seus problemas. Pretensos pacientes importunavam Weizenbaum com chamadas telefônicas pedindo-lhe que os colocasse em linha com seu empático computador.

Tudo isso surpreendeu o criador de ELIZA. "Eu estava espantado", escreveu mais tarde Weizenbaum, "de ver quão rápida e profundamente as pessoas ficavam emocionalmente envolvidas com o computador, e quão inequivocamente elas o antropomorfizavam. Certa vez a minha secretária, que me observara trabalhar durante muitos meses no programa e sabia, portanto, que ele não passava de um mero programa de computador, começou a conversar com ele. Depois de apenas algumas trocas de palavras com ele, me pediu para deixar a sala."

O que perturbava ainda mais Weizenbaum era que alguns psiquiatras viam em ELIZA um instrumento promissor para aliviar a insuficiência de terapeutas humanos. Um deles, Kenneth Colby, da Stanford, amigo de Weizenbaum e que, evidentemente, influenciou a decisão de programar ELIZA como terapeuta, era particularmente entusiasta. Colby escreveu que, com ELIZA funcionando como DOCTOR numa rede de terminais em partilha de tempo, "várias centenas de pacientes por hora poderiam ser tratados por um sistema de computador".

A idéia de Colby sobre uma terapia computadorizada assustava Weizenbaum. Isso, juntamente com o fato de que Colby dedicou-se a planejar sua própria versão de ELIZA, desencadeou disputas que acabaram com a amizade entre os dois. De fato, todo esse episódio parece ter amargurado Weizenbaum em relação às pesquisas sobre a AI em geral. Ele continuou no MIT, mas como um dos mais ferozes céticos a respeito da AI.

Enquanto ELIZA e outros programas conversacionais semelhantes da década de 1960 exploravam as possibilidades de manipular palavras de acordo com seus significados, outros programas trabalhavam em grande medida com a forma, ou sintaxe, da linguagem. Essa abordagem foi influenciada pelas teorias de Noam Chomsky, no MIT, notável lingüista defensor da tese de que os seres humanos nasceram com estruturas mentais inatas, nas quais traduzem as sentenças que ouvem ou lêem. As idéias de Chomsky atraíram vários pesquisadores que tinham preferência pelos

sistemas de AI baseados na lógica e na matemática pura. Sua noção de que um modelo formal, semimatemático, podia ser feito de linguagem sugeriu a esses cientistas que poderiam ser capazes de adaptar algumas das hipotéticas estruturas chomskianas a programas de computador. Assim, segundo essa linha de pensamento, a máquina para compreensão da linguagem não precisaria entender o significado de uma palavra, assim como um computador que calcula folhas de pagamento tampouco precisa entender o que é dinheiro ou trabalho.

Num típico sistema baseado em sintaxe, os computadores são equipados com vocabulários extensos e algum conhecimento acerca de partes da fala. Um deles, um analisador de sintaxe residente, conhecido como *parser*, usa centenas de regras de gramática, semelhantes àquelas aprendidas pelas crianças no nível escolar, para fragmentar sentenças em componentes tais como as locuções nominais e as locuções verbais. Uma regra, por exemplo, indica que as palavras "o" e "um" são, provavelmente, seguidas por locuções nominais.

Alguns pesquisadores, no entanto, começaram a ficar insatisfeitos com a abordagem sintática. A seu ver, ela não produzia uma compreensão da linguagem mais válida que o impassível sistema do ELIZA para equipar palavras-chave; assim como os abortados esforços iniciais de tradução por máquina, a abordagem sintática tendia a encalhar também na ambigüidade. Um exemplo favorito dos críticos era a simples sentença: "O tempo voa como uma flecha" (*Time flies like an arrow*). Um sistema baseado em sintaxe analisará gramaticalmente essa sentença e surgirá com não menos de quatro significados diferentes, incluindo a seguinte amostra de erudição automatizada: "Moscas de uma certa espécie, denominadas *time flies* gostam de flechas".

O QUE HÁ DENTRO DE UMA PALAVRA

Convencidos de que dominar as regras formais da gramática não era suficiente, esses críticos começaram a enfatizar de novo a importância da semântica. Os significados das palavras individuais, assinalaram, constituem apenas uma pequena fração disso. Os seres humanos resolvem facilmente as ambigüidades da linguagem porque a abordagem compreende o contexto de uma dada sentença. A compreensão do contexto depende, por sua vez, de um conhecimento detalhado do assunto em questão.

Um importante passo nessa direção foi dado em 1968, quando Terry Winograd, na época um estudante graduado no MIT, começou a trabalhar num tipo diferente de programa para processamento de linguagem. Winograd batizou-o com o nome de SHRDLU, cadeia de letras aparentemente mutilada, mas que era na verdade parte de um símbolo que fazia sentido para o impressor: na época que antecedeu a composição gráfica computadorizada, os operadores das máquinas de linotipo usavam a frase sem sentido "etaoin shrdlu" — que podia ser composta rapidamente, correndo o dedo pelas duas primeiras linhas de teclas — para marcar uma posição de parada temporária em seu trabalho. Com seu programa SHRDLU, Winograd fornecia ao computador um corpo de conhecimentos acerca de um minúsculo domínio, um micromundo que consistia numa caixa vazia, num tampo de mesa, menos de uma dúzia de cubos e pirâmides coloridas e um braço simples de robô para manipular esses blocos semelhantes a brinquedos, empilhando-os ou colocando-os na caixa. Tudo isso, inclusive o braço de robô, era simulado na memória do computador e exibido em três dimensões na tela.



SHRDLU foi um marco — e provavelmente o programa de computador mais complexo escrito até essa época. Acreditando que os vários aspectos do entendimento da linguagem eram inseparáveis, Winograd integrou com êxito no SHRDLU as até então discretas funções da sintaxe, semântica e capacidade de raciocínio; todas as três operadas simultaneamente no processamento do inglês, em vez de seqüencialmente, como nos programas anteriores.

Como resultado, o SHRDLU demonstrou uma notável capacidade de abarcar elementos que pertenciam tanto ao seu minúsculo domínio quanto ao inglês. O programa respondia a comandos digitados tais como "Apanhe o bloco vermelho que está perto da pirâmide verde" ordenando ao braço de robô a execução da tarefa. Tal feito impressionou bastante naquela época; mas, além disso, o SHRDLU também era capaz de indicar que não estava entendendo algo. Por exemplo, se o comando fosse vago, a máquina exibiria na tela do computador uma sentença como esta: "Não entendo a que pirâmide você se refere".

Por mais impressionante que fosse, o SHRDLU mostrou ser impotente fora de seu próprio domínio e não prático para se expandir com o conhecimento do mundo real. Mesmo dentro de seu minúsculo domínio, ele não é capaz de entender comandos que não sejam expressos no inglês mais preciso. A máquina empacava em questões imprecisas como esta: "Quantos blocos é necessário colocar um no topo do outro para fazer um campanário?" Ele podia não entender o sentido da frase "colocar um no topo do outro", que contém um paradoxo óbvio, se a frase for entendida em seu sentido literal. Os seres humanos tendem a não tomar literalmente tal frase. Eles compreendem de imediato seu significado graças à miscelânea de fatos e intuições conhecida como senso comum. Por isso, depois do SHRDLU, muitos pesquisadores procuraram dotar suas máquinas de quantidades sempre maiores de conhecimento — não apenas dados altamente especializados, mas também algo que se aproxima do senso comum que as pessoas trazem às conversas cotidianas. O truque tem consistido em descobrir meios de organizar tudo isso e representá-lo num programa de computador. A representação do conhecimento, nome atribuído a esse problema, preocupou pesquisadores não apenas interessados no entendimento da linguagem mas também na AI em geral. Os debates sobre a melhor maneira de resolvê-lo tornaram-se cada vez mais acesos, de modo semelhante ao que ocorria com a controvérsia sobre sintática e semântica.

A maior parte dos proponentes de uma abordagem formal, lógica, do entendimento da linguagem — e das pesquisas sobre a AI em geral — preferem o método rigorosamente matemático das regras se-então para a representação do conhecimento, que foram empregadas com resultados muito bons nos sistemas expert. Outros pesquisadores, que, tipicamente, preferem uma abordagem semântica, argumentam que as regras se-então são demasiadamente simples para dominar todas as coisas que as pessoas conhecem e usam de modo a ajudá-las a compreender a linguagem. Esses cientistas do computador trabalham com métodos para co-



Redes semânticas: ligando nódulos de conhecimento

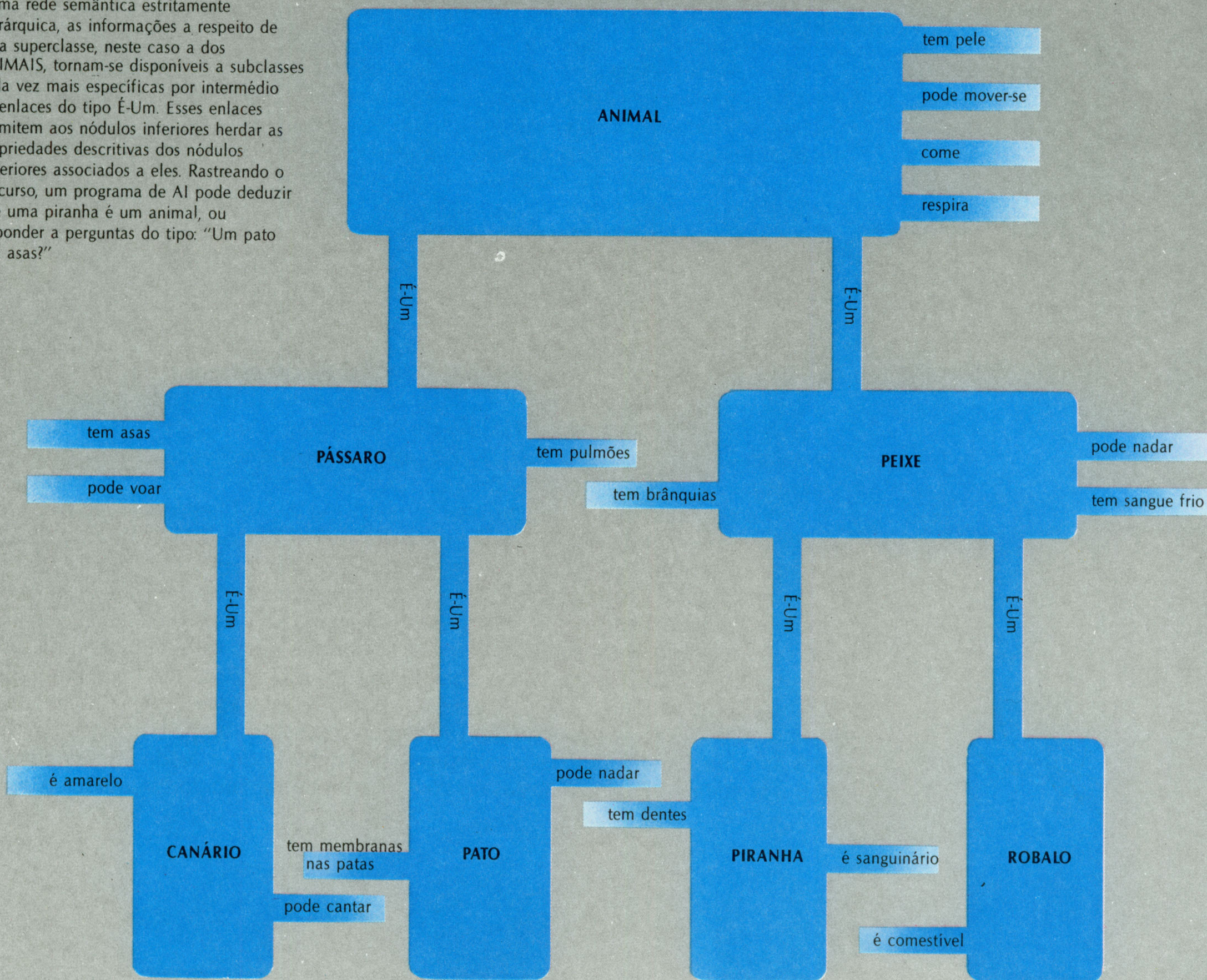
A capacidade de raciocínio, que as pessoas usam rotineiramente, e com facilidade, requer vastas quantidades de informação sobre todo tipo de coisas e suas inter-relações, bem como conhecimento de senso comum acerca do mundo como um todo. Para dotar um computador com essa capacidade, todos esses dados devem ser organizados e armazenados dentro da memória do computador de tal maneira que possam ser rapidamente manipulados para produzir inferências válidas ou para responder a perguntas específicas.

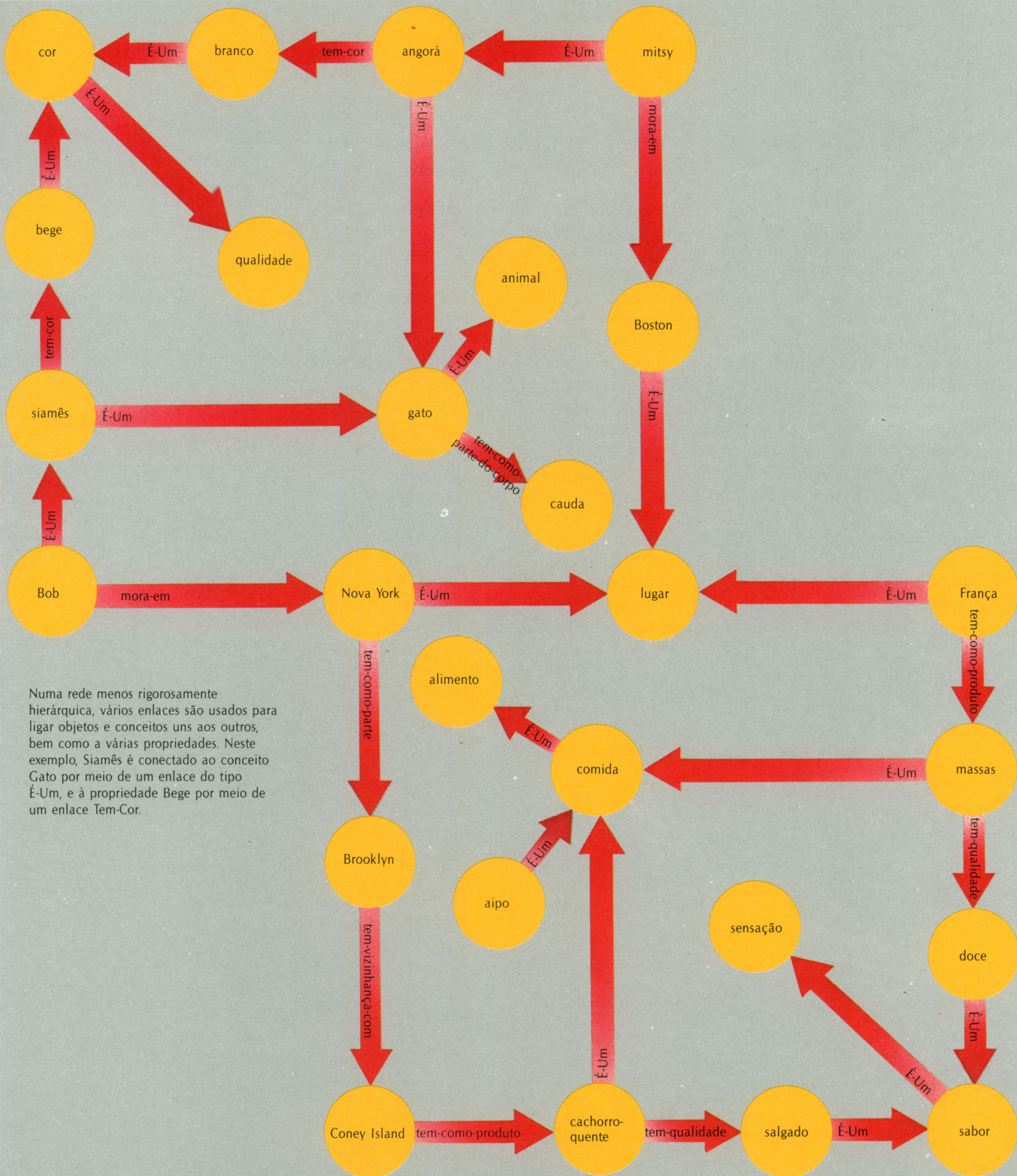
Uma das mais eficientes dessas técnicas de armazenamen-

to é a rede semântica — uma estrutura de dados formada por nódulos, que significam objetos ou conceitos, e enlaces que indicam as relações entre os nódulos. Nódulos e enlaces são facilmente representados por esquemas. Dentro do computador, os nódulos correspondem, efetivamente, a registros ou posições na memória, ao passo que os enlaces correspondem a indicadores — códigos de endereços que enviam o programa a essas posições na memória.

O tipo mais importante de enlace denomina-se enlace É-Um: tais conexões organizam o conhecimento de uma rede semântica numa hierarquia que permite nódulos inferiores herdar as propriedades de nódulos superiores. Assim, o nódulo CANÁRIO, na rede semântica mostrada nesta página, herda todos os atributos descritivos dos nódulos PÁSSARO e ANIMAL, situados acima dele. Esse mecanismo de herança responde por grande parte da eficiência das redes semânticas. A herança também conserva espaço para o armazenamento na memória do computador e possibilita uma forma de raciocínio dedutivo.

Numa rede semântica estritamente hierárquica, as informações a respeito de uma superclasse, neste caso a dos ANIMAIS, tornam-se disponíveis a subclasses cada vez mais específicas por intermédio de enlaces do tipo É-Um. Esses enlaces permitem aos nódulos inferiores herdar as propriedades descritivas dos nódulos superiores associados a eles. Rastreando o percurso, um programa de AI pode deduzir que uma piranha é um animal, ou responder a perguntas do tipo: "Um pato tem asas?"





Numa rede menos rigorosamente hierárquica, vários enlaces são usados para ligar objetos e conceitos uns aos outros, bem como a várias propriedades. Neste exemplo, Siamês é conectado ao conceito Gato por meio de um enlace do tipo É-Um, e à propriedade Bege por meio de um enlace Tem-Cor.

dificar arranjos mais complexos de conhecimentos, num esforço para competir com as elaboradas associações mentais que as palavras parecem evocar na mente humana.

Nas palavras de Scott Fahlman, pesquisador voltado para a abordagem semântica, em Carnegie-Mellon: "se digo a você que Clyde é um elefante, você num instante passa a saber uma porção de coisas sobre Clyde. Você sabe que ele provavelmente tem dois olhos, quatro patas e é de cor cinza. Você sabe o que acontecerá a ele se você levá-lo até o topo de um edifício e deixá-lo cair, e o que acontecerá se você o mantiver debaixo d'água durante duas horas. Você sabe o que acontecerá a você se tentar mantê-lo debaixo d'água durante duas horas. Mas nenhuma dessas coisas está armazenada em sua cabeça sob a palavra 'Clyde'. Estão armazenadas sob coisas como 'elefante' ou 'gravidade' ou 'objeto físico'".

Essa concepção de como as associações são armazenadas na memória — e de como um programa de computador pode incorporá-las — foi assunto de um influente esquema de representação de conhecimento proposto por M. Ross Quillian no fim da década de 1960. Quillian, na época um estudante graduado, concebeu a memória humana como uma imensa teia intricadamente interconectada, a que deu o nome de rede semântica. Tal rede, sugeriu, consiste em nodos que correspondem a objetos, conceitos e eventos, e ligações, ou vínculos, entre os nodos, que especificam a natureza das conexões. No exemplo do elefante, dado por Fahlman, a rede semântica abarcaria vários nodos e vínculos. O nome "Clyde" seria ligado ao objeto "elefante" que, por sua vez, seria conectado a outros nodos contendo conceitos tais como "mamífero" e propriedades tais como "grande" e "cinzento".

Para criar uma rede semântica num computador, Quillian deu início a um metódico processo de pesquisa de dicionário. Em primeiro lugar, procurou as defini-





ções de palavras comuns em inglês; a seguir, procurou a definição de cada palavra em cada definição; a partir desses dados, compilou listas de referências cruzadas semelhantes a dicionários, nas quais cada palavra era definida em termos das outras. Essas listas requeriam mais memória do que os computadores da época podiam oferecer; por isso, o programa completo de Quillian consistia em menos de sessenta palavras e suas elaboradas referências cruzadas.

UMA ESTRUTURA PARA O CONHECIMENTO

Embora o programa não vivesse o bastante para ver realizadas as altas esperanças de Quillian — ele mesmo acabou trocando a AI pela ciência social —, o conceito de rede semântica inspirou maneiras similares de abordar a representação do conhecimento. Em 1974, Marvin Minsky, do MIT, sugeriu que a mente humana interpreta toda a situação nova, inclusive a linguagem, à luz de estruturas de dados mentais a que ele deu o nome de quadros (*frames*, páginas 16 e 17). Um quadro é complexo pacote de conhecimento — armazenado na mente ou num programa de computador — que descreve um objeto ou conceito. Cada pacote contém brechas, que enumeram atributos e valores associados. Um pacote para o conhecimento da palavra cão pode ter, por exemplo, brechas para dados sobre raça, sexo e dono, juntamente com brechas vazias para serem preenchidas por novos fragmentos de conhecimento.

Isso seria um pacote simples. Na realidade, Minsky antevê estruturas de conhecimento muito mais complexas encadeadas umas às outras, e nas quais algumas das brechas podem consistir em pacotes inteiros, que por sua vez podem aninhar-se dentro de pacotes maiores. Dado um conjunto suficiente de pacotes, um computador pode varrer uma sentença procurando símbolos que correspondam aos que estão nas brechas existentes, ativando desse modo o pacote básico e quaisquer pacotes relacionados. Por esse caminho, um sistema pode inferir conhecimento sobre um assunto que não estava explícito no início: por exemplo, que o cão mora na Rua das Flores, nº 40, e gosta de ossos.

Minsky descreveu sua proposta sobre pacotes de conhecimento num memorando caracteristicamente vigoroso e cheio de especulações — deixando os detalhes para que outro os desenvolvessem. Era um cientista do computador de fôlego incomum, que estivera, como diz, “pensando sobre o pensamento” desde o colégio.

Se Minsky tornou-se o mais antigo defensor da antilógica, na ala semântica da compreensão da linguagem o *enfant terrible* é Roger Schank. Ele é o eloqüente diretor do departamento de ciência do computador em Yale, onde, juntamente com sua equipe de estudantes, produz idéias e esquemas de representação do conhecimento tão originais que os colegas às vezes referem-se a eles como a “escola schankiana” de pesquisas sobre AI.

De acordo com Schank, as pessoas não equiparam, meramente, palavras e significados; em vez disso, traduzem tudo em estruturas conceituais subjacentes — uma espécie de “mentais” — que capta o sentido daquilo que ouvem ou lêem. Assim, por exemplo, que as pessoas quase nunca se lembram exatamente das palavras faladas ou escritas; em vez disso, lembram-se dos pontos essenciais do que ouvem ou lêem, que exprimem usando suas próprias palavras. Essa capacidade conceitual, sugere, permite-nos decodificar com facilidade pesadelos sintáticos tais como este: “Golpeio o rapaz com a garota com cabelos longos com um martelo com raiva”.

Quadros: pacotes de conceitos e dados

As redes semânticas (páginas 12 e 13) não são a única maneira de abordar o problema da codificação do conhecimento, de modo que ele possa ser aplicado por um computador. Outro método consiste em organizar as informações sobre objetos ou conceitos em pacotes complexos, ou quadros.

O desenvolvimento desses pacotes foi influenciado por pesquisas que mostram que as pessoas arquivam e "consultam" mentalmente suas experiências em amplos termos conceituais,



PACOTE MENINO

É-UM | INDIVÍDUO

sexo | masculino

idade | menos de 12 anos

casa | uma residência

PACOTE ALEX

É-UM | MENINO

sexo | masculino

idade | 8 anos

casa | R. União, 10

comida favorita

sobe

tipo de corpo

banana

árvores

magro

Para conservar espaço na memória do computador, os pacotes podem ser organizados hierarquicamente com subpacotes herdando as propriedades de superpacotes relacionados. No pacote MENINO, acima, a primeira brecha atua como um enlace É-Um para identificar a relação do pacote com seu pacote de origem, neste caso, INDIVÍDUO. De maneira semelhante, a primeira brecha do pacote ALEX indica que esse pacote é uma subclasse do pacote MENINO. Como tal, ALEX herda todas as brechas — sexo, idade e casa — de suas duas superclasses, MENINO e INDIVÍDUO.

Manipulando a analogia "Alex é semelhante a um macaco", o computador usa um conjunto de heurísticas para examinar o pacote-fonte, MACACO, à procura de brechas cujos valores possam ser transferidos ao pacote-alvo, ALEX. Outro conjunto, que inclui heurísticas tais como "selecione brechas não preenchidas no pacote-alvo", permite à máquina escolher a melhor das analogias propostas. Por meio desse processo, o computador pode ampliar o que ele conhece a respeito de Alex.

e a seguir preenchem os detalhes na medida das necessidades. O sistema imita essa atividade fornecendo uma estrutura global de conhecimento que consiste em brechas a serem preenchidas com exemplos e dados específicos. As brechas identificam propriedades de um objeto ou conceito e contêm os valores particulares associados a cada propriedade. Em alguns casos, uma brecha pode reter um valor *default*, que entrará em vigor na ausência de entrada de um outro valor. Tipicamente,

os valores armazenados nas brechas são ponteiros para outros pacotes.

À semelhança das redes semânticas, os pacotes complexos podem ser ligados para formar uma hierarquia que permite que propriedades descritivas sejam herdadas. Neste exemplo, a máquina transfere valores da brecha de um pacote-fonte para um pacote-alvo, usando a heurística para guiar a seleção efetiva de propriedades análogas.



PACOTE MACACO

É-UM

PRIMATA

sexo

masculino
ou feminino

idade

um número
inteiro

casa

default = selva

comida
favorita

default =
banana

sobe

árvores

tipo de
corpo

default =
magro

PACOTE BANANA

É-UM

FRUTO

cor

amarelo

sabor

doce

casca

espessa

fonte

bananeira

Outra característica dos sistemas de conhecimento baseados em pacotes de conhecimento é a capacidade das brechas para apontar pacotes inteiros. Aqui, o valor BANANA no pacote MACACO dirige o programa até uma posição na memória onde está armazenado o conhecimento do computador sobre bananas — um pacote completo em si mesmo, com brechas para cor, sabor, casca e fonte. O pacote BANANA, por sua vez, pode ser ligado a pacotes relacionados, formando uma complexa estrutura de intercomunicações de conhecimento.



tivas, como aquelas com que os seres humanos abordam uma dada situação. Isso permite à máquina completar-se com informações que estão faltando quando o exame é efetuado a partir de sentenças individuais, mas que o contexto da situação implica. “Ordenar um bife é diferente de ordenar uma tarefa a um empregado ou ordenar seus assuntos”, explica Schank. “Mas, se eu disser a você que se trata de um restaurante, você nem mesmo pensará nos outros sentidos da palavra ordem.”

Para explorar scripts, Schank e seus alunos planejaram, em 1974, um programa denominado SAM (sigla para *Script-Applier Mechanism*, isto é, “Mecanismo Aplicador de Scripts”). O primeiro de vários sistemas para entendimento de histórias desenvolvido em Yale, o SAM utiliza scripts para resumir ou responder perguntas sobre narrativas simples fornecidas à máquina. Por exemplo, SAM foi instruído para processar a seguinte historinha com a ajuda de seu script do restaurante: “John foi a um restaurante. Ordenou um hambúrguer. Pagou com cheque e saiu”. Quando perguntaram a SAM: “O que John comeu?” ele respondeu: “John comeu um hambúrguer”. Embora essa informação não tivesse sido especificamente declarada na história, o SAM pôde, com base no contexto, inferir a resposta.

Outra inovação criada em Yale foi a adição de estruturas semelhantes a pacotes para ajudar a máquina a entender motivações. Essas várias estruturas — denominadas objetivos, planos e temas — suplementam os scripts, explicando ações humanas em termos de finalidades desejadas. Desse modo, quando se diz a um programa equipado com essas estruturas que “John não tinha dinheiro e entrou na adega empunhando uma arma”, ele pode inferir que John provavelmente tinha outra coisa em mente além de comprar bebidas, e automaticamente reporta-se ao seu conhecimento sobre motivos de roubo.

A CRIAÇÃO DE TEIAS DE MEMÓRIA

Embora os scripts se comprovassem úteis, tinham vários defeitos, entre os quais sua inflexibilidade e o restrito leque de situações que eram capazes de retratar. Schank e seus alunos dedicaram-se a esses problemas criando estruturas de dados mais extensas, denominadas MOPs (sigla para *Memory Organization Packets*, isto é, “Pacotes de Organização da Memória”). Diferentemente dos scripts, que são fatias de conhecimento isoladas, os MOPs são ligados em teias mais complexas, que reconhecem as similaridades entre eventos ocorridos em diferentes contextos. Essas teias de memória permitiram aos pesquisadores da Universidade de Yale construir um programa capaz de responder perguntas a respeito de uma história complicada, que requeria o conhecimento de vários domínios diferentes. Numa das cenas da narrativa, sediada no local favorito dos programadores da universidade — um restaurante —, a máquina foi informada de que: “George estava lanchando quando a garçonete, acidentalmente, derrubou nele um copo de Coca-Cola. George ficou muito aborrecido e foi embora, recusando-se a pagar a conta”. Essa passagem ativa partes de vários MOPs diferentes, permitindo à máquina inferir que George saiu sem pagar devido ao péssimo serviço de mesa.

Outros programas em Yale utilizavam várias combinações de estruturas de dados para monitorar e resumir noticiários transmitidos pela TV a cabo. Um desses programas de computadores, equipado com o script adequado, mantinha-se informado a respeito das atividades terroristas no mundo todo, e então fazia generalizações a respeito das histórias que lia. Concluiu, por exemplo, que cada ataque terrorista na Irlanda era efetuado pela organização ilegal IRA, o Exército Republi-

cano Irlandês, e que cada seqüestro aéreo no Líbano tinha por objetivo protestar contra o desaparecimento de um líder muçulmano xiita.

Um promissor programa desse tipo devotou-se exclusivamente à leitura e resenha de reportagens da TV a cabo sobre uma pessoa: Cyrus R. Vance, que trabalhou como Secretário do Estado dos EUA durante o final da década de 1970. Apropriadamente, esse programa foi denominado CYRUS, e alguém em Yale chegou até mesmo a reconhecer nesse nome uma autêntica acrossemia: *Computerized Yale Reasoning and Understanding System*, isto é, "Sistema de Entendimento e Raciocínio Computadorizado de Yale". CYRUS era capaz de lembrar e de analisar tantas informações a respeito de Vance que podia até mesmo fornecer fatos que nunca haviam aparecido efetivamente nas notícias transmitidas. Quando lhe foi perguntado se a esposa de Vance tinha alguma vez se encontrado com a esposa do Primeiro Ministro de Israel, Menachem Begin, CYRUS arriscou uma adivinhação bem informada. Confiando no seu conhecimento de que Vance certa vez esteve presente num banquete governamental, e que as esposas com freqüência estão presentes nessas ocasiões, ele supôs — corretamente — que as esposas de fato haviam se encontrado, e chegou até a especificar a data.

Os que criticam a maneira como Schank aborda a representação do conhecimento argumentam que ela é confusa e de difícil manejo. Assinalam também que o desempenho dos programas daquela universidade só é impressionante em estreitos domínios do conhecimento, tais como restaurantes e atividades oficiais de um único funcionário público.

O DEBATE CONTINUA

Muitos desses críticos são engenheiros comprometidos no planejamento de sistemas de entendimento de linguagem comuns e práticos, baseados na lógica matemática simples das regras se-então, que se mostraram bem-sucedidas nos sistemas expert. Um porta-voz que se pronuncia claramente sobre o assunto é Nils Nilsson, líder em inteligência artificial no SRI Internacional, organização de pesquisas privadas em Menlo Park, Califórnia.

Nilsson crê que as imensas coleções de conhecimentos fortuitos requeridas pela escola schankiana acabam comprovando-se impossíveis para o engenheiro. "A história toda da engenharia sempre foi a de tentar simplificar e procurar princípios", disse ele. "Penso que ao construir máquinas inteligentes, temos o dever de pelo menos tentar manter esse mesmo espírito de engenharia. A questão é se você acredita que existem certas coisas tão inexprimíveis que realmente não podem ser descritas formalmente."

Pode ser que cada um, por seu lado, tenha razão em tais debates, e que as máquinas, de algum modo, tenham de incorporar todos os aspectos das abordagens conflitantes — semântica e sintática, baseadas na lógica formal e nos scripts schankianos — antes que possam adquirir um entendimento substancial da linguagem. O fato é que, a despeito de todo o impressionante progresso nesse campo, em mais de duas décadas de experimentação nenhum programa de AI sequer chegou perto das capacidades de linguagem de uma criança média de quatro anos. Além disso, muitos críticos duvidam que as máquinas cheguem algum dia a adquirir entendimento, no sentido mais amplo da palavra. John R. Searle, professor de filosofia em Berkeley, insiste no fato de que os computadores podem chegar apenas a uma simulação do entendimento, e não à coisa real. "Ninguém irá supor que

as simulações por computador de um incêndio ponham fogo na vizinhança ou que uma simulação de tempestade por computador nos deixe ensopados”, escreveu Searle. “Por que alguém na terra iria supor que uma simulação do entendimento por computador realmente entenda qualquer coisa?” Joseph Weizenbaum, o desiludido criador do programa ELIZA, coloca a questão de outra maneira: “Talvez seja possível, seguindo os procedimentos de Schank, construir uma estrutura conceitual que corresponda ao significado da sentença ‘Você virá jantar comigo esta noite?’ mas é difícil conceber”, continua Weizenbaum, “de que maneira esquemas semelhantes aos de Schank poderiam entender que essa mesma sentença é capaz de significar a desesperada ânsia por amor pronunciada por um jovem tímido. Mesmo que o computador consiga simular sentimentos de desespero e de amor, seria ele, por isso, capaz de se desesperar e de amar? Poderia então o computador entender o desespero e o amor?” Weizenbaum responde negativamente à sua própria pergunta, e conclui que “até mesmo o mais poderoso sistema schankiano”, no final das contas, não consegue, há mais de duas décadas, superar em capacidade de entendimento sua própria e rudimentar ELIZA.



Aprender: um desafio

2

Nos primeiros dias da computação eletrônica, um problema demasiado familiar para cientistas e engenheiros era o de conseguir dinheiro para construir suas máquinas. Por isso, não constitui nenhuma façanha o ato de, em meados de 1947, um grupo de professores, na Universidade de Illinois, ter persuadido os diretores da escola a procurar financiamento para o planejamento e a construção de um computador. Mas os 110 000 dólares que a universidade conseguiu não renderam o quanto originalmente se esperava. Um dos membros do grupo, um professor de engenharia elétrica chamado Arthur L. Samuel, surgiu com uma proposta instigante: montar um pequeno computador e ensiná-lo a jogar damas. Argumentou que esse novo experimento de aprendizagem da máquina geraria extensa publicidade e talvez atraísse dinheiro do governo e de fontes privadas para manter ativo o projeto principal.

Samuel escolheu damas em vez de xadrez — o jogo preferido pelos pesquisadores que vieram mais tarde, e que explorariam o campo que viria a ser conhecido como inteligência artificial — porque suas regras são muito mais simples e requerem menos espaço na memória limitada do computador. Além disso, recordou ele muitos anos depois, “aconteceu que na primavera seguinte haveria uma competição de campeões mundiais de damas na pequena cidade vizinha de Kankakee. Acreditávamos que o jogo de damas era um jogo trivial. No final da competição, desafiáramos o campeão mundial e ganharíamos dele, veja você, e isso chamaria sobre nós um bocado de atenção. Éramos bastante ingênuos”.

Nessa época, Samuel — um homem genial, ativo, então beirando os cinquenta anos — ocupava um cargo no magistério e também dirigia o Electron Devices Laboratory, que fora encarregado de planejar os componentes para a nova máquina. Envolvia-se agora no desafio adicional de escrever um programa para jogar damas. Era um terreno novo para ele. Trabalhara como pesquisador técnico nos Laboratórios Bell, antes e durante a Segunda Guerra Mundial, tornando-se uma autoridade em válvulas eletrônicas — os dispositivos que acionaram a primeira geração de computadores digitais. Mas ele não sabia nada sobre programação. Assim como não sabia muita coisa a respeito do jogo de damas.

Como seria de esperar, a competição de damas de Kankakee teve de ser realizada sem a participação do computador. Mas o malogrado projeto ensinou-lhe o suficiente sobre computadores e programação para atirá-lo numa nova carreira e numa obsessão com aquilo que desde essa época tornou-se um foco-chave das pesquisas sobre AI: ensinar as máquinas a aprender a partir da experiência.

Em 1949, procurando uma oportunidade para trabalhar em tempo integral com computadores, Samuel ligou-se à equipe do laboratório de pesquisas da IBM em Poughkeepsie, Nova York, quando ajudou a planejar a primeira linha de computadores comerciais da IBM, enquanto prosseguia no desenvolvimento do programa para jogar damas, em seu tempo livre. De fato, a crescente extensão e complexidade de seu programa tornava-o ideal para testar o novo 701, o primeiro computador da IBM produzido em série, com programa armazenado e para todos os fins, lançado em 1952. Samuel jogava damas nesse computador à noite, de-

A capacidade de aprender através da experiência é um atributo-chave da inteligência humana — e a meta mais difícil de perseguir nas pesquisas sobre AI. Os cientistas em computador desenvolveram programas que podem aprender a identificar formas simples, mas um desafio como o da ilustração — o reconhecimento de que as esferas, cones e cilindros podem, todos os três, projetar sombras circulares — está muito além dos poderes de qualquer computador atual.

purando seu programa favorito ao mesmo tempo em que testava a nova máquina.

A sobreposição dos papéis de Samuel, como planejador de computadores e programador em tempo parcial, acabou despertando comentários — que só eram piada em parte — de que a lógica das primeiras máquinas da IBM fora efetivamente planejada para facilitar a representação de um tabuleiro de damas. Naturalmente, tais insinuações não eram vistas com bons olhos pela empresa. Sob a direção de Thomas Watson Sr., o perspicaz presidente, cioso da imagem de sua firma, a IBM tolerava o trabalho extracurricular de Samuel mas pouco fazia para encorajá-lo ou divulgá-lo. “Era algo que se parecia demais com uma máquina pensando”, lembrou Samuel, “e eles queriam dissipar quaisquer preocupações que as pessoas tivessem a respeito de máquinas invadindo o mundo e esse tipo de coisa.” O lema da IBM. “Pense”, aplicava-se apenas aos funcionários.

OS RUDIMENTOS DA APRENDIZAGEM

Procurando não fazer estardalhaço demais, Samuel trabalhou contínua e tranquilamente no aperfeiçoamento de seu programa. No início, a capacidade da máquina dependia apenas daquilo que Samuel denominou aprendizagem de rotina. Ele programou as regras do jogo e ensinou o computador a olhar vários movimentos à frente, e avaliar as conseqüências potenciais de uma dada posição, atribuindo, para isso, valores numéricos a várias opções.

Embora diferente em outros aspectos, o programa de Samuel é precursor dos programas expert atuais, que tiram proveito do conhecimento de especialistas num dado campo. Infelizmente, Samuel não era um perito, e a literatura sobre damas e os jogadores talentosos que consultou pouco ajudaram para a obtenção de heurísticas melhores. Por isso, inicialmente, o programa não jogava damas melhor que seu criador.

Samuel remediou essa deficiência equipando o programa com um tipo melhor de perícia, a que deu o nome de aprendizagem por generalização. Ele incorporou ao programa providências para que este se lembrasse e agisse de acordo com a experiência que fosse adquirindo. Calculando o valor das várias opções, o programa levava em consideração o sucesso ou fracasso de movimentos semelhantes no passado. Assim, a máquina era guiada não apenas pela heurística residente mas também pela sabedoria nascida da experiência. Assim como o jogador humano, ele era capaz de aprender a partir de seus próprios erros.

Devido a isso, o programa passou a melhorar alguns aspectos de seu desempenho em quase todas as competições. Logo ele vencia Samuel tão facilmente que este prometeu não voltar a jogar de novo até que seu próprio jogo estivesse muito melhor. No verão de 1962 — quinze anos depois que Samuel deu à luz a idéia — o programa já estava jogando a nível de mestre, e preparado para enfrentar seu primeiro campeão humano. Samuel desafiou Robert W. Nealey, antigo campeão do Estado de Connecticut, que chegara a ficar invicto durante oito anos, e a máquina ganhou facilmente. “Quanto ao jogo final”, Nealey disse mais tarde, “não tinha enfrentado um desempenho como este com nenhum ser humano desde 1954, quando perdi minha última partida.”

A META DO AUTO-APERFEIÇOAMENTO

O programa de Samuel permaneceu um marco nas pesquisas sobre AI, menos por sua habilidade em jogos de damas (em 1977, o programa encontrou um adversá-

rio à altura num programa rival desenvolvido na Universidade de Duke) do que por sua capacidade notável — embora rudimentar — de modificar seu comportamento de acordo com as experiências pelas quais ia passando. O desenvolvimento dessa capacidade — a própria essência da aprendizagem — tornou-se a partir daí o objetivo de muitos pesquisadores, que estão convencidos de que qualquer inteligência artificial digna desse nome deve ser capaz de se auto-aperfeiçoar.

Embora o objetivo em si pareça suficientemente claro, as técnicas com que se pretende alcançá-lo são todas sujeitas a controvérsia. Um problema é, simplesmente, o da vasta quantidade de tempo necessário para fornecer às máquinas um banco de conhecimentos suficiente para começarem a funcionar. Outro problema, estreitamente relacionado ao primeiro, é o de saber se há, de fato, aprendizagem da máquina, considerando que o programa requer a presença de um ser humano para abastecer seu banco de dados e para equipá-lo, mais tarde, com regras que operam sobre esses dados. Os sistemas expert, por exemplo, são bastante elogiados em muitos círculos, devido a suas aplicações práticas. Mas até mesmo os mais sofisticados sistemas são capazes de realizar apenas o equivalente em máquina da aprendizagem por rotina: quaisquer informações e heurísticas que um sistema expert utilize devem ser explicitamente fornecidas por programadores trabalhando com engenheiros em conhecimento, num procedimento tão dispendioso e demorado que constitui um gargalo crítico. Conforme observou Roger Schank, líder das pesquisas sobre AI na Universidade de Yale: "Cada fatia de conhecimento que você coloca dentro de um computador requer uma tremenda codificação de novas informações. Nós simplesmente não podemos codificar manualmente todas essas coisas. Os seres humanos não aprendem dessa maneira".

Tais frustrações deram origem a outra abordagem, chamada aprendizagem por exemplos (ou, às vezes, aprendizagem por indução). Uma criança pequena aprende o conceito de árvore observando exemplos que as pessoas apontam como sendo árvores, extraindo as características essenciais que distinguem essa classe de plantas e generalizando a partir desses dados específicos para chegar a um conceito. Os pesquisadores de AI que investigam a aprendizagem por exemplo esperam evitar o entediante processo de programar em máquinas a perícia humana. Descobrir esse atalho não somente acelerará a criação de sistemas expert como também instilará neles um pouco da maravilhosa capacidade da mente humana de adaptar o comportamento às circunstâncias em mudança.

Patrick H. Winston, que se tornou diretor do laboratório de AI no MIT em 1973, tentou resolver o problema de programar uma máquina para aprender por exemplos enquanto ainda era estudante do curso de graduação no MIT, no final da década de 1960. Winston planejou um programa, chamado Arches, e, por meio de alguns exemplos cuidadosamente selecionados, ensinou-o a reconhecer conceitos arquitetônicos simples.

Em primeiro lugar, Winston forneceu ao computador várias configurações de blocos de brinquedo; desenhos de montagens com esses blocos foram convertidos em descrições estruturais de configurações particulares. Então veio a parte da aprendizagem: para ajudar a máquina a identificar relações espaciais entre os blocos, Winston forneceu ao programa uma heurística que lhe permitia identificar relações tais como "à esquerda de", "encosta" e "sustenta". Para iniciar a fase da aprendizagem, Winston forneceu à máquina descrições de três blocos que formavam um arco e disse a ela, de fato, que "Isto é um arco". A seguir, forneceu à máquina alguns

assim chamados quase-acertos, que diferiam num atributo — por exemplo, um desenho no qual o terceiro bloco não era sustentado pelos outros dois — dizendo à máquina: “Isto não é um arco”.

O programa imitava um processo que, segundo se acreditava, ocorria na mente humana. Consistia em isolar as características relevantes presentes nos exemplos positivos e observar similaridades e diferenças, com relação aos exemplos negativos. Depois de algumas tentativas, o programa ensinava a si mesmo um conceito simples mas essencial: um arco consiste num bloco central no topo de dois blocos de sustentação que não se tocam um ao outro.

O sucesso de Winston com o Arches encorajou outros pesquisadores no campo, que levaram seu trabalho um passo adiante. Enquanto o Arches limitava-se ao mundo dos blocos de brinquedo, um programa desenvolvido em 1976 por Ryszard Michalski e seus colegas, na Universidade de Illinois, lidava com a formação de conceitos de importância prática. Seu programa, o AQ11, foi de início equipado com descrições de 290 pés de soja doentes, incluindo sintomas tais como a cor e a condição das folhas e do caule, juntamente com um diagnóstico de perito para cada planta. Processando esses dados, o AQ11 ensinou a si mesmo regras “se-então” para identificar doenças nas plantas a partir de determinados sintomas, tornando-se, assim, uma espécie de sistema expert autoprogramado.

Num experimento notável, Michalski contrapôs a heurística autodesenvolvida do AQ11 a heurísticas planejadas por peritos humanos em patologia de plantas. O objetivo dessa batalha de saberes era diagnosticar corretamente as quinze doenças mais comuns da soja. Os pesquisadores consultaram alguns dos principais patologistas de plantas e traduziram seu conhecimento humano em diagnósticos na forma de regras “se-então” — procedimento muito parecido ao que usariam para programar um sistema expert comum. Então digitaram os sintomas observados em 340 pés de soja doentes, que haviam sido coletados para os experimentos.

A máquina era então instruída para diagnosticar as doenças em duas rodadas separadas. Numa das rodadas, ela usava heurística proveniente dos cérebros dos peritos humanos; na outra, usava as regras que o AQ11 formulara por si mesmo depois de ser exposto a exemplos de doenças de soja. O AQ11 ganhou a batalha, acertando o diagnóstico correto 97,6 por cento das vezes com suas próprias regras, em comparação com uma taxa de sucesso de 71,8 por cento quando confiava na heurística derivada de peritos humanos. Versões posteriores do AQ11 foram aplicadas em áreas tais como a medicina, a física e o xadrez.

TESTE DE HIPÓTESES COM EXPERIMENTOS

As técnicas de autoprogramação do AQ11 e suas diversas configurações representaram um grande avanço na aprendizagem da máquina. Em outros lugares, começava-se a trabalhar com programas de aprendizagem que eram capazes não só de perceber padrões em dados observados e de inferir hipóteses a partir desses padrões, mas também de submeter tais hipóteses ao teste da experiência. Esse processo de descoberta autoguiada está no cerne do método científico. Como os pesquisadores de AI gostam de assinalar, um de seus termos favoritos, “heurística”, relaciona-se lingüisticamente a “eureka” — que em grego significa, literalmente, “achei” —, palavra que Arquimedes, ao que se supõe, gritou ao saltar fora de seu banho e correr nu pelas ruas da antiga Siracusa, após descobrir o princípio da gravidade específica.

Desde o final da década de 70, o processo de aprendizagem por descoberta foi assunto de uma série de programas elaborados por Herbert Simon, Patrick Langley e outros colegas da Universidade de Carnegie-Mellon. Os programas são conhecidos coletivamente como BACON, em homenagem ao filósofo da ciência inglês Sir Francis Bacon, um dos primeiros defensores da pesquisa empírica e do raciocínio indutivo. Esse nome histórico é especialmente adequado, uma vez que tais programas incluem entre suas realizações a redescoberta de muitas das mais profundas leis da ciência, inclusive a lei do deslocamento de Arquimedes, que foi incluída em sua formulação da gravidade específica.

À PROCURA DE RELAÇÕES

A capacidade do BACON depende basicamente de um pequeno número de heurísticas destinadas a descobrir relações matemáticas significativas entre os conjuntos de dados introduzidos na máquina. Uma vez que o programa tenha detectado tendências e outras regularidades na mistura aparentemente caótica de números, ele formula uma série de hipóteses. Cada hipótese apóia-se na precedente, e o processo resulta finalmente numa lei científica.

Por exemplo, num dos experimentos pediu-se a BACON para considerar medições relativas ao sistema solar. A partir dos dados, ele propôs a hipótese de uma relação direta entre as distâncias de cada planeta ao Sol e a duração dos anos dos planetas. Depois de descobrir correlações adicionais entre os números, BACON chegou à seguinte conclusão: "A razão entre o quadrado da duração de um ano planetário e o cubo da distância desse planeta ao Sol é a mesma para todos os planetas". Acontece que essa é a terceira lei do movimento planetário, de Johannes Kepler. Formulada pelo astrônomo alemão aproximadamente há quatro séculos — depois de vários anos de observações, em vez dos minutos de agitação digital requeridos pelo BACON —, tal lei constituiu o alicerce onde se assentaria a descoberta posterior de Newton das leis da mecânica.

Em outro experimento, forneceu-se a BACON o mesmo tipo de dados numéricos sobre química que eram disponíveis ao pesquisador inglês John Dalton no início do século XIX. Devido à sua capacidade de determinar se cada membro de uma seqüência de medidas é múltiplo de algum outro número, o programa redescobriu a lei das proporções múltiplas, de Dalton, que estabelece que os elementos químicos tendem a combinar-se uns com os outros segundo razões simples, de números inteiros. Essa descoberta levou Dalton a postular um conceito fundamental da química e da física modernas — a existência do átomo.

Embora o talento do BACON para descobertas — ou, mais precisamente, para redescobertas — seja impressionante, representa pouco progresso rumo ao objetivo de tornar a aprendizagem da máquina menos dependente das pessoas. À semelhança das explorações de Patrick Winston com a aprendizagem por exemplos, o BACON deixa a maior parte do trabalho criativo para os programadores humanos. São eles, e não o próprio programa, que selecionam os dados e decidem quais as características ou medições relevantes que serão introduzidas na máquina.

De modo semelhante, argumentam os críticos, o BACON só é tão hábil em fazer descobertas graças aos vários séculos de laborioso esforço humano que o precederam. Douglas B. Lenat, amigo do principal criador do programa, Herb Simon, assinalou que o BACON depende de medições precisas e de técnicas matemáticas cuidadosamente lapidadas, desenvolvidas por várias gerações de cientistas, desde

EXEMPLOS
DADOS DE
ADVERTÊNCIA

CONHECIMENTO INICIAL

REALIMENTAÇÃO

Projeto para uma máquina automodificadora

REGRAS
DE DESEMPENHO
MODIFICADAS

Programas de tarefa específica, para aplicações como processamento de texto, folhas de pagamento e jogos de computador, são planejados para executar uma função específica de uma maneira específica, seguindo regras de operação que o próprio programa não pode alterar.

Os programas de aprendizagem, ao contrário, destinam-se a modificar seu desempenho com base na experiência. Alguns já são capazes de simplificar as pesquisas científicas discernindo padrões em massas de dados desorganizados; aprendizagens de máquinas ainda mais sofisticadas serão necessárias para aventuras espaciais, onde os computadores a bordo de espaçonaves precisarão contar com capacidade própria de investigação e de tomada de decisões.

A ilustração à esquerda sintetiza os elementos comuns à maioria dos modelos de aprendizagem de máquina, sejam eles programas destinados a aprender por analogia, por descoberta ou por instrução. O núcleo que tem o nome de conhecimento inicial, ilustrado aqui como uma massa amarela, possui importância central nesse modelo. Como no caso dos sistemas expert, esse conhecimento inicial é fornecido pelo programador e inclui fatos e heurísticas pertinentes ao assunto (que geralmente entram no programa sob a forma de afirmações "se-então"). O conhecimento inicial também pode incluir instruções codificadas para organizar informações, assim como metac conhecimento, que é às vezes descrito como o entendimento que o programa tem de suas próprias forças e fraquezas.

A aprendizagem de máquinas resulta da interação do conhecimento inicial com vários tipos de entrada e saída. A entrada (*seta verde*) pode consistir em advertências do usuário, que o programa deve interpretar e aplicar, bem como em casos e exemplos específicos, a partir dos quais o programa deve induzir regras ou conceitos gerais. À medida que processa essas informações, o programa pode modificar os fatos e a heurística em seu banco de conhecimentos; a modificação se reflete na saída do programa, ou no desempenho de tarefa melhorado (*seta azul*). (Em alguns modelos, define-se o desempenho melhorado como capaz de fornecer soluções mais precisas ou de encontrar soluções a um custo mais baixo; em outros, é definido como uma expansão da gama de problemas que a máquina pode manipular ou como a capacidade da máquina para simplificar suas próprias regras, a fim de preservar a memória.) Finalmente, o laço de realimentação (*seta vermelha*) avalia o desempenho do programa, comparando resultados com objetivos, isolando erros e recomendando correções. Melhoramentos subseqüentes são denotados aqui pelos alargamentos sombreados do banco de conhecimentos.

as descobertas originais. "Você dá ao BACON pares de números livres de erros e ele aparece com uma expressão matemática que relaciona aquela encontrada com esta saída", observou Lenat. "Você pode dar a isso o nome 'descobrir as leis da física', se quiser, mas terá na verdade de manter os olhos firmemente semifechados para poder chamá-lo assim. Certamente, se Kepler tivesse sido informado de que poucas variáveis eram relevantes e tivesse à sua disposição os dados perfeitos de que BACON dispunha, então ele precisaria de apenas uma hora para obter sua lei."

Lenat, nascido em 1950, pertence a uma geração que não teme questionar seus mentores, ou tentar novas abordagens. Na visão de alguns pesquisadores, o próprio trabalho de Lenat sobre aprendizagem por descoberta tem sido extremamente promissor, e talvez esteja abrindo caminho para sistemas expert capazes de resolver problemas com um mínimo de intervenção humana.

UM CAMINHO SINGULAR PARA PESQUISAS SOBRE AI

Em 1975, Lenat criou um programa de aprendizagem por descoberta para sua tese de doutoramento, em Stanford. Assim como o BACON, o programa de Lenat era mais eficiente para a redescoberta de velhas leis, só que no campo da matemática e não da física. Lenat chamou esse sistema pioneiro de AM, as iniciais de *Automated Mathematician*, isto é, "Matemático Automatizado".

Para permitir que o AM explorasse os mundos da teoria elementar dos conjuntos e da teoria dos números, Lenat forneceu-lhe 115 definições de conceitos — tais como o de igualdade — e de operações — tais como a intersecção de conjuntos. Ele também o equipou com 250 regras do tipo "se-então" para guiá-lo na procura de conceitos mais complexos. Essas regras forneciam critérios para focalizar vias de exploração particulares. Por exemplo, a máquina podia ser instruída desse modo: "Se alguma ação comprovar-se útil numa dada situação, então tente a mesma ação em situações semelhantes".

Provavelmente, as heurísticas mais produtivas que Lenat escreveu para o AM foram as 51 regras que ajudavam o programa a avaliar se algo era ou não suficientemente interessante para ser investigado. Os critérios incluíam a presença de regularidades e similaridades nas relações matemáticas. Por exemplo, no curso de uma investigação do conceito previamente descoberto dos "divisores de", o AM achou interessante o fato de que alguns números só admitem divisão exata por dois divisores — eles mesmos e um — descobrindo, assim, o conceito de número primo.

O AM provou ser um aprendiz apto. Em apenas poucas horas de operação, progrediu a partir de um estado de quase ignorância em matemática até a redescoberta de uns duzentos dos mais importantes conceitos de teoria dos números. Ensinou a si mesmo o conceito de quadrados (números que podem ser expressos como outro número multiplicado por si mesmo) e chegou até mesmo a uma hipótese conhecida como conjectura de Goldbach — cada número par maior que dois é a soma de dois números primos.

Assim como o programa clássico de Arthur Samuel para jogar damas, o AM também demonstrou uma capacidade limitada para aprender a partir da experiência. Ele retinha as hipóteses que suas heurísticas residentes julgavam interessantes e verdadeiras, e descartava as restantes.

Veteranos pesquisadores da AI consideravam tudo isso uma realização perfeitamente digna de crédito, proveniente, como era, de um recém-chegado a esse cam-

po — um mero estudante graduado —, mas o próprio Lenat estava insatisfeito. Por um motivo: o AM limitava-se ao seu domínio, a matemática; era altamente improvável que seu programa tropeçasse com uma nova grande descoberta num campo já bem explorado ao longo dos séculos por algumas das maiores inteligências do mundo. Mais significativo, do ponto de vista de Lenat, foi o fato de que, embora pudesse descobrir conceitos, o AM não podia gerar novas regras simples e práticas baseado no que descobrira. De fato, quanto mais o AM se afastava de suas regras simples e práticas pré-programadas — quando procurava investigar situações que sua heurística não abarcava —, mais insustentáveis suas operações se tornavam. Ele podia, por exemplo, supor que um número era ímpar e par ao mesmo tempo — algo que um ser humano ou uma heurística viável nunca fariam.

Em 1976, Lenat ocupou um posto de professor no Carnegie-Mellon e começou a desenvolver uma versão mais poderosa e versátil do AM. Seu novo programa — um sistema expert com capacidade de aprendizagem — exploraria outros domínios além da matemática; Lenat deu-lhe o nome de Eurisko, que em grego significa “descubro coisas”.

Mas Lenat enfrentou uma árdua batalha para tornar o funcionamento do Eurisko digno de seu nome. Em Carnegie-Mellon e, depois, em Stanford, onde voltou a lecionar em 1978, lutou para desenvolver uma compreensão melhor acerca da natureza das heurísticas e de como programar um computador para produzi-las.

INDO AO FUNDO DO MISTÉRIO

Aconteceu, então, uma ruptura. Analisando a história do AM, Lenat compreendeu que o sucesso de seu velho programa em matemática — assim como seu fracasso em heurística — decorria, em parte, do fato de o AM ter sido escrito em LISP. Como as funções matemáticas — por exemplo, o teste para determinar se um número é primo ou não — podiam ser escritas sucintamente em apenas duas ou três linhas do código LISP, as operações fundamentais do AM eram, de fato, soberbamente adequadas a uma manipulação eficiente de conceitos matemáticos. Mas não havia uma relação semelhante entre o LISP e a produção de heurísticas; uma única heurística programada no AM requeria, com frequência, duas páginas repletas de códigos. Qualquer tentativa de manipular tal programação complexa levava inevitavelmente o AM a investigar tantas possibilidades e propor tantos conceitos que a miríade de resultados era quase invariavelmente inútil.

Lenat respondeu construindo um novo tipo de linguagem de programação — a que deu o nome de RLL, sigla para *Representation Language Language*, isto é, “Linguagem da Linguagem de Representação” —, que lhe permitia expressar regras heurísticas de maneira mais compacta dentro da estrutura do LISP. Em vez de escrever as regras num grande aglomerado de códigos LISP, ele quebrava as complicadas sentenças “se-então” em conceitos básicos, ou pacotes complexos; os modificadores que representavam os atributos fundamentais de um conceito ocupavam brechas dentro de cada pacote.

Usando esse método de representar as informações, Lenat inseriu na sintaxe e no vocabulário da nova linguagem do Eurisko seus *insights* sobre a natureza da heurística, de modo muito parecido àquele pelo qual McCarthy construiu o LISP, para manipular relações matemáticas e lógicas. O resultado disso foi que o Eurisko podia separar uma extensa hipótese “se-então” e examinar todos os seus componentes para determinar o que o fazia funcionar ou deixar de funcionar.

Inicialmente, Lenat equipou o Eurisko com 150 heurísticas para todos os fins, aplicáveis a qualquer domínio. Algumas dessas constituíam o que Lenat chamou de metaregras — heurísticas destinadas a criar novas heurísticas. Eis um exemplo: “Se uma regra heurística demora para rodar ou ser executada, então especialize-a numa nova regra que provavelmente rodará com menos freqüência mas levará aos mesmos resultados úteis que a primeira”.

Para testar o Eurisko, Lenat selecionou campos do conhecimento relativamente inexplorados, onde parecia existir uma chance de fazer novas descobertas, em vez de apenas refazer as antigas. Ele também escolheu domínios onde era provável que se apresentassem possibilidades a tal ponto numerosas que derrotariam a mente humana não aparelhada.

Um campo que prontamente demonstrou satisfazer a ambos os objetivos foi o do desenho de circuitos integrados mais eficientes para computadores. Embora os chips de computadores do tipo conhecido como circuitos VLSI (*Very-large-scale integrated*, isto é, “integrados” em escala muito grande) já contivessem centenas de milhares de transistores e outros componentes numa única fatia minúscula de silício, os engenheiros queriam que esses chips acondicionassem ainda mais componentes. Uma das idéias para obter maior densidade consistia no assim chamado desenho de perfil largo (*high-rise design*). Além de distribuir horizontalmente os componentes numa superfície bidimensional, como nos arranjos convencionais, esse desenho empilha-os verticalmente em muitas camadas.

A visualização desses complexos arranjos tridimensionais é difícil para os seres humanos, mas o Eurisko aprendeu rapidamente a achar o caminho em meio a esses pequenos mundos labirínticos.

O EURISKO TRABALHANDO E BRINCANDO

O Eurisko também se revelou um aprendiz competente e um solucionador de problemas em domínios tão diversos como a simulação da evolução biológica e a limpeza de produtos químicos derramados. Mas atraiu inicialmente ampla atenção ao realizar a tarefa aparentemente trivial que Arthur Samuel designou para seu programa de jogo de damas, mais de três décadas antes: aprender a jogar um jogo e ganhar, então, um campeonato.

A competição, realizada em 1981, na Califórnia, era uma versão futurista de batalha naval, conhecida como Traveller TCS. Cada competidor tinha de planejar uma frota imaginária de espaçonaves, que seria então posta em confronto com uma frota oponente numa batalha simulada. O objetivo do jogo era destruir toda a frota adversária. A chave para ganhar o jogo estava mais no planejamento das naves do que na estratégia, uma vez que as próprias batalhas eram determinadas por lances de dados e por elaboradas regras e tabelas que avaliavam a possibilidade de um dado tipo de nave sobreviver a um embate. Dentro das limitações do dinheiro disponível — cada jogador dispunha de um trilhão de “créditos” em moeda intergaláctica —, os jogadores podiam variar o tamanho e o número de suas naves, juntamente com características tais como couraças, armamentos e agilidade.

Meses antes do torneio, Lenat — que nunca havia jogado Traveller — começou a preparar o Eurisko para sua estréia numa competição. Introduziu no programa as cem páginas de regras do jogo, que avaliavam as “despesas militares” e as qualidades de diversas características do desenho. Então, todas as noites, antes de deixar seu escritório, Lenat carregava o Eurisko no seu computador Xerox 1100. Du-

2

rante a noite toda, a máquina desenhava espaçonaves e as testava em batalha contra oponentes imaginários. Se uma nave perdia a luta, o Eurisko analisava a causa, aprendendo desse modo lições que eram abstraídas numa nova heurística. A seguir, retornava à prancheta de desenho — substituindo as armas de uma nave, por exemplo, por canhões de laser — e testava na batalha o novo desenho.

UM TIME DE CAMPEONATO

Todas as manhãs, Lenat selecionava as idéias que a máquina havia desenvolvido durante a noite, conservando as que gostaria de ver o Eurisko continuar a explorar. Lenat descreveu posteriormente esse processo como um “sinergismo homem-máquina” — cerca de 60 por cento Lenat e 40 por cento Eurisko.

À medida que trabalhava, o Eurisko demonstrava uma capacidade rudimentar para adquirir conhecimentos por meio de processos que também são vitais para grande parte da aprendizagem humana — o uso de analogias e a aplicação a novas situações de coisas que se mostraram funcionais em contextos prévios. Por exemplo, em sua primeira tarefa de desenhar chips de computador tridimensionais, ele descobriu que a simetria é uma propriedade desejável para tais chips, embora não entendesse por quê. O Eurisko lembrou-se dessa regra, e durante uma das primeiras sessões de construção de naves decidiu fazer a tentativa de tornar simétrico o espaço da frota, balanceando fatores tais como o tamanho e o tipo da nave. Justificou essa ação referindo-se à sua experiência com circuitos integrados.

Alguns dos guias mais importantes do Eurisko compunham-se do estoque de heurísticas para todos os fins que o orientavam independentemente do assunto explorado. Uma dessas regras, “Dê uma olhada em casos extremos”, mostrou-se de valor inestimável. Aplicando essa regra ao jogo, o programa testou-a e, a seguir, modificou-a, criando uma nova heurística que dava um alto valor aos casos “quase extremos”. Em grande parte como consequência disso, Lenat escreveu mais tarde, a frota do Eurisko evoluiu para uma formação extremamente não convencional, “consistindo quase que inteiramente em pequenas e velozes naves de ataque, que mais se pareciam com lanchas torpedeiras e que incluíam um navio tão rápido e minúsculo que era virtualmente indestrutível”.

A navezinha criada pela estratégia “quase extrema” tornou-se o trunfo do Eurisko. Seu papel não era ofensivo nem defensivo, mas apenas o de sobreviver a todas as batalhas e garantir assim, pelo menos, um empate forçado.

Na época em que Lenat e Eurisko se apresentaram para a competição de Traveler, tinha-se gasto cerca de 2 000 horas de computador na preparação. Mas os competidores zombaram de sua excêntrica frota de 96 naves. As outras armadas eram muito mais convencionais — contendo, tipicamente, cerca de um quinto do número de naves da frota de Eurisko, “sensatamente” balanceadas em termos de tipo e tamanho. Mas assim que humanos foram defrontados com o Eurisko, os combatentes reconheceram a lógica de sua estratégia “quase extrema”. Compreendendo que a minúscula nave era capaz de sobreviver praticamente a qualquer batalha, muitos deles resignaram-se, sem dar um único tiro. Eurisko ganhou facilmente a competição e recebeu seu prêmio: galões de almirante espacial e título de propriedade de um planeta de sua escolha, fora do sistema solar.

No ano seguinte, os organizadores da competição mudaram as regras do jogo, com a intenção óbvia de eliminar a possibilidade de Eurisko vencer novamente com sua estratégia da nave minúscula. Sem se perturbar com isso, Lenat previu a vitória.

“Quando eles escrevem novas regras, o que realmente estão fazendo é acrescentar complexidade”, disse. “Isso apenas leva o Eurisko a descobrir novas armas”.

O Eurisko justificou a impetuosa confiança de Lenat. Descobriu, nas novas regras, um número de possibilidades de defesa suficiente para vencer o campeonato de 1982 — e para forçar os organizadores a adotar uma estratégia que acabou eliminando Eurisko dos campeonatos seguintes. Se Eurisko ganhasse novamente, declararam, o campeonato seria abolido. Vítima de sua própria destreza, Eurisko retirou-se da competição.

Lenat e seu programa atraíram considerável atenção fora do pequeno círculo de jogadores de Traveller — até mesmo a atenção do Pentágono foi atraída, contribuindo para persuadir seus oficiais a investigar sistemas de controle de batalha baseados em programas semelhantes ao Eurisko.

Mais importante ainda, o trabalho de Lenat também intrigou executivos da MCC (Microelectronics and Computer Technology Corporation), um consórcio de pesquisas fundado em 1982, em Austin, Texas. Recrutado para que prosseguisse aí suas pesquisas, Lenat deu início a um projeto novo e ambicioso: equipar um computador com o *background* necessário para ler e compreender uma enciclopédia de um volume.

UM BANCO DE CONHECIMENTOS ENCICLOPÉDICO

Lenat e a maioria dos outros pesquisadores de AI “partem do axioma de que você não aprende algo a menos que, de alguma maneira, você já praticamente saiba o que é”, disse Patrick Winston, do MIT, que desenvolveu o programa Arches.

O novo projeto de Lenat é extraordinário. Denominado Cyc — um homófono, em inglês, de “*psych*” —, espera-se que esse programa gigantesco mantenha ocupada uma equipe de 35 pessoas durante pelo menos uma década. Do modo como foi concebido, conterà no final cerca de 250 000 segmentos de conhecimento, tais como regras simples e práticas, fatos e recordações de experiências — a quantidade de informações que Lenat estima ser necessária para que um leitor humano entenda todos os verbetes de uma pequena enciclopédia.

Lenat qualificou essas informações como senso comum — o rico repertório de fatos e intuições que os seres humanos adquirem automaticamente através da experiência diária e que até agora não conseguiu ser atingido nem pelas mais avançadas máquinas de aprendizagem. “Agora, Eurisko está congelado”, disse ele. “Será descongelado quando tivermos esse corpo de conhecimentos de senso comum.”

A educação de um robô pelo exemplo



Dentre todas as habilidades que denotam inteligência humana, a capacidade para aprender é talvez a mais difícil de explicar e de codificar em procedimentos de programação passo a passo. Sem essa habilidade, as máquinas nunca seriam capazes de enfrentar circunstâncias que não são previstas com precisão por um programador humano. De posse dela, os computadores teriam a capacidade de modificar seu comportamento e de aperfeiçoar seu desempenho através da aquisição independente de novos conhecimentos. Não é, pois, surpreendente que várias abordagens ao problema da aprendizagem da máquina tenham sido empreendidas. As técnicas de aprendizagem variam desde a memorização de rotinas até o método mais sofisticado do raciocínio por analogia. Mas talvez a abordagem mais extensamente estudada seja a da aprendizagem por exemplos, também conhecida como aprendizagem por indução.

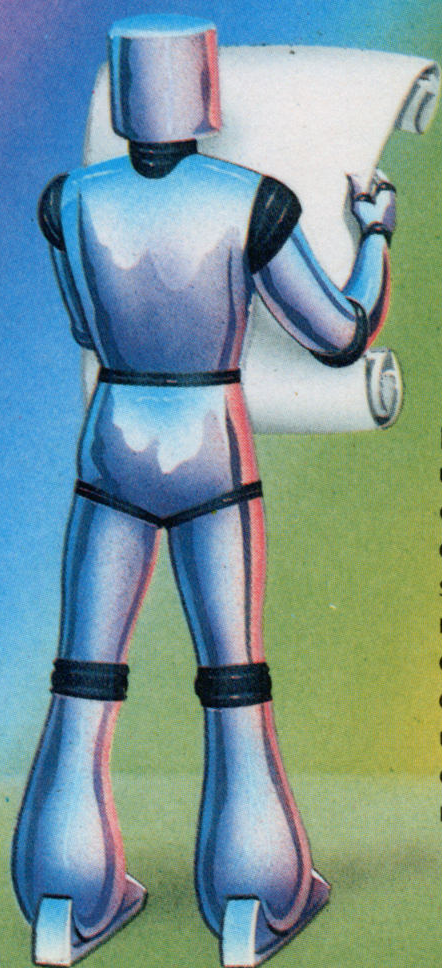
Diz-se que um sistema aprende por exemplos se ele pode tirar conclusões gerais a respeito de um dado fenômeno, estudando casos específicos designados como exemplos positivos ou negativos do fenômeno. Para esse fim, o sistema requer heurísticas, ou regras simples e práticas, para guiá-lo no processo de indução, instruindo o sistema para observar certas similaridades e diferenças entre amostras positivas e negativas.

Na aprendizagem por exemplos, também se fornece ao sistema dois tipos de serviço: uma tarefa de aprendizagem (por exemplo, formar generalizações) e uma tarefa de desempenho, que é a atividade a ser aperfeiçoada por meio da aprendizagem. Para realizar qualquer uma dessas tarefas, o sistema deve possuir fatos acerca do assunto no qual operará, e outras regras simples e práticas que facilitarão seu progresso através desse domínio.

A aprendizagem por exemplos é ilustrada nas páginas seguintes com uma história bastante fantasiosa de um robô que deve aprender a distinguir entre dois tipos de dragões — aqueles que querem dar seu ouro de presente e aqueles que só querem ouro para si mesmos. Aprendendo a reconhecer quais dragões são gananciosos e quais são generosos, o robô torna-se mais hábil em sua tarefa de desempenho: reunir ouro para encher um pote no fim do arco-íris. Os métodos usados pelo robô podem ser aplicados por computadores reais para executar tarefas tais como jogar xadrez ou distinguir doenças que apresentem sintomas similares.

Armazenando o conhecimento inicial

No cerne de um sistema de aprendizagem há um pacote de informações residentes, que permite ao sistema dar início a suas tarefas de desempenho e aprendizagem. Esse conhecimento inicial cobre três áreas: a tarefa de desempenho, a tarefa de aprendizagem e o domínio no qual se processam essas duas tarefas. Nos sistemas projetados para aprender por exemplos, os casos específicos encontrados podem provar que um dado fato, ou regra, está errado ou incompleto. A tarefa de aprendizagem do sistema consiste, por isso, em refinar seu conhecimento inicial



Preparando-se para iniciar sua tarefa, o robô lê um rolo de pergaminho representando o conhecimento inicial contido em seu programa de aprendizagem (*box à direita*). Suas instruções são: seguir o arco-íris, coletar ouro dos dragões no caminho e encher o pote na outra extremidade. Para realizar essa tarefa de desempenho, o robô também deve levar consigo uma tarefa de aprendizagem — formar um conceito, ou hipótese, que lhe permita reconhecer dragões generosos.

ou em criar novo conhecimento devido a essa experiência, melhorando, no processo, seu desempenho.

O robô cuja tentativa de aprendizagem é ilustrada aqui recebe como conhecimento inicial várias instruções e fatos (*box*); por exemplo, a instrução: "Siga o arco-íris do começo ao fim" pertence à tarefa de desempenho do robô. Sua tarefa de aprendizagem pode ser colocada assim: "Aprenda a reconhecer dragões generosos". Seu conhecimento de domínio consiste em afirmações como "Dragões possuem ouro" e nas assim chamadas hierarquias

de generalização de atributos dos dragões. As hierarquias organizam as características relevantes de uma maneira que permite ao robô fazer generalizações a partir de encontros com dragões específicos. O conhecimento de domínio também inclui a heurística, que guia o sistema de aprendizagem quase como um professor o faria. Se o robô fosse programado para aprender a jogar xadrez, e não para viajar pelo arco-íris e encontrar dragões, seu conhecimento de domínio incluiria, entre outras coisas, as regras do jogo e uma descrição de cada peça do xadrez.

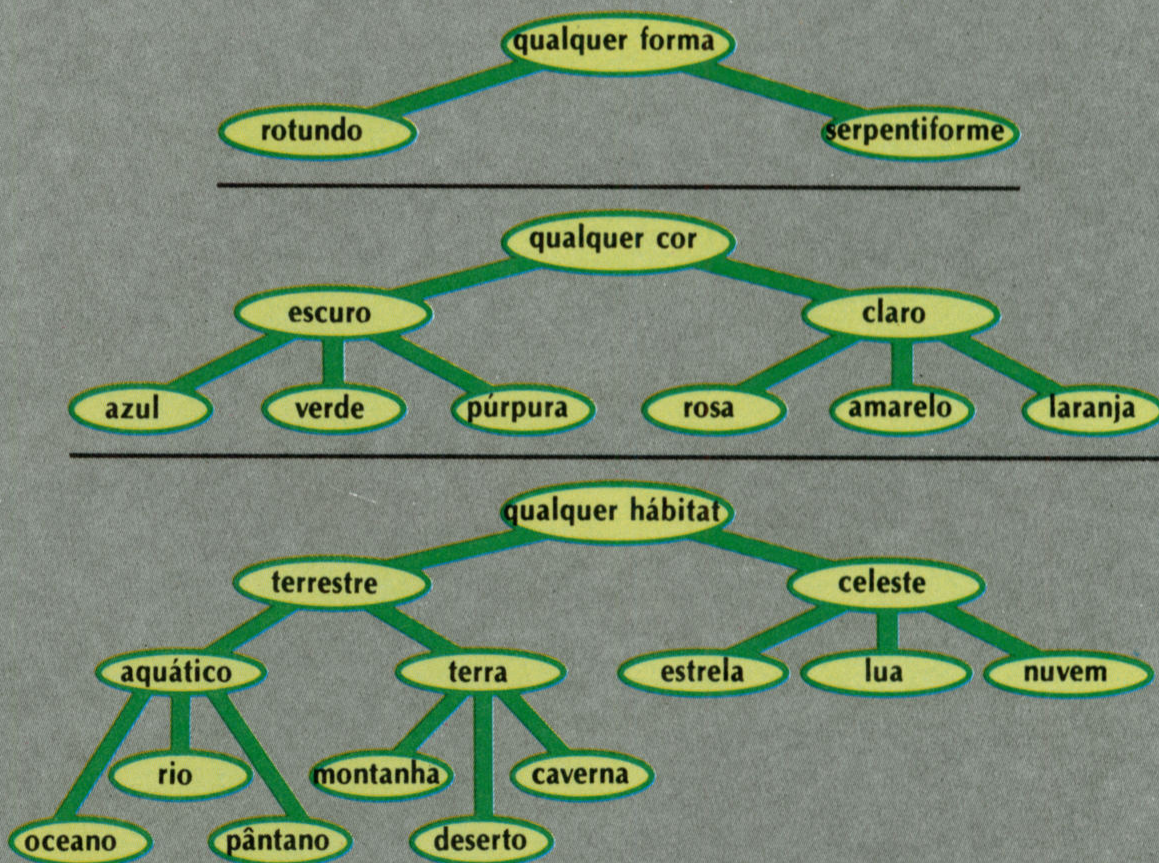
Um kit de partida com regras e fatos

A lista abaixo revela os diferentes tipos de conhecimento inicial residentes no programa do robô. O item 1 revela a tarefa de aprendizagem do robô, e os itens 2 a 4 listam as condições da tarefa de desempenho. Os itens 5 a 8, juntamente com as três hierarquias de generalização (*abaixo, à direita*), dão ao robô fatos a respeito

dos dragões; o item 9 sublinha as hipóteses que o robô deve formar por aprendizagem para associar certas características ao comportamento generoso. Finalmente, é dada ao robô uma heurística, o item 10; sem ela, o robô ficaria paralisado — ele não teria base para aproximar-se do primeiro dragão.

1. Aprenda uma generalização que inclua todos os exemplos de dragões generosos e exclua todos os exemplos de dragões gananciosos.
2. Siga o arco-íris do começo ao fim.
3. Peça ouro aos dragões generosos; evite os dragões gananciosos.
4. Encha de ouro o pote no fim do arco-íris.
5. Dragões possuem ouro.
6. Dragões gananciosos tentam roubar ouro.
7. Dragões generosos dão ouro.
8. Um dragão pode ter qualquer forma e qualquer cor e pode viver em qualquer hábitat.
9. Um dragão generoso tem forma (----), cor (----) e vive no hábitat (----).
10. Considere que um dragão é generoso, a menos que você tenha uma evidência em contrário.

hierarquias de generalização

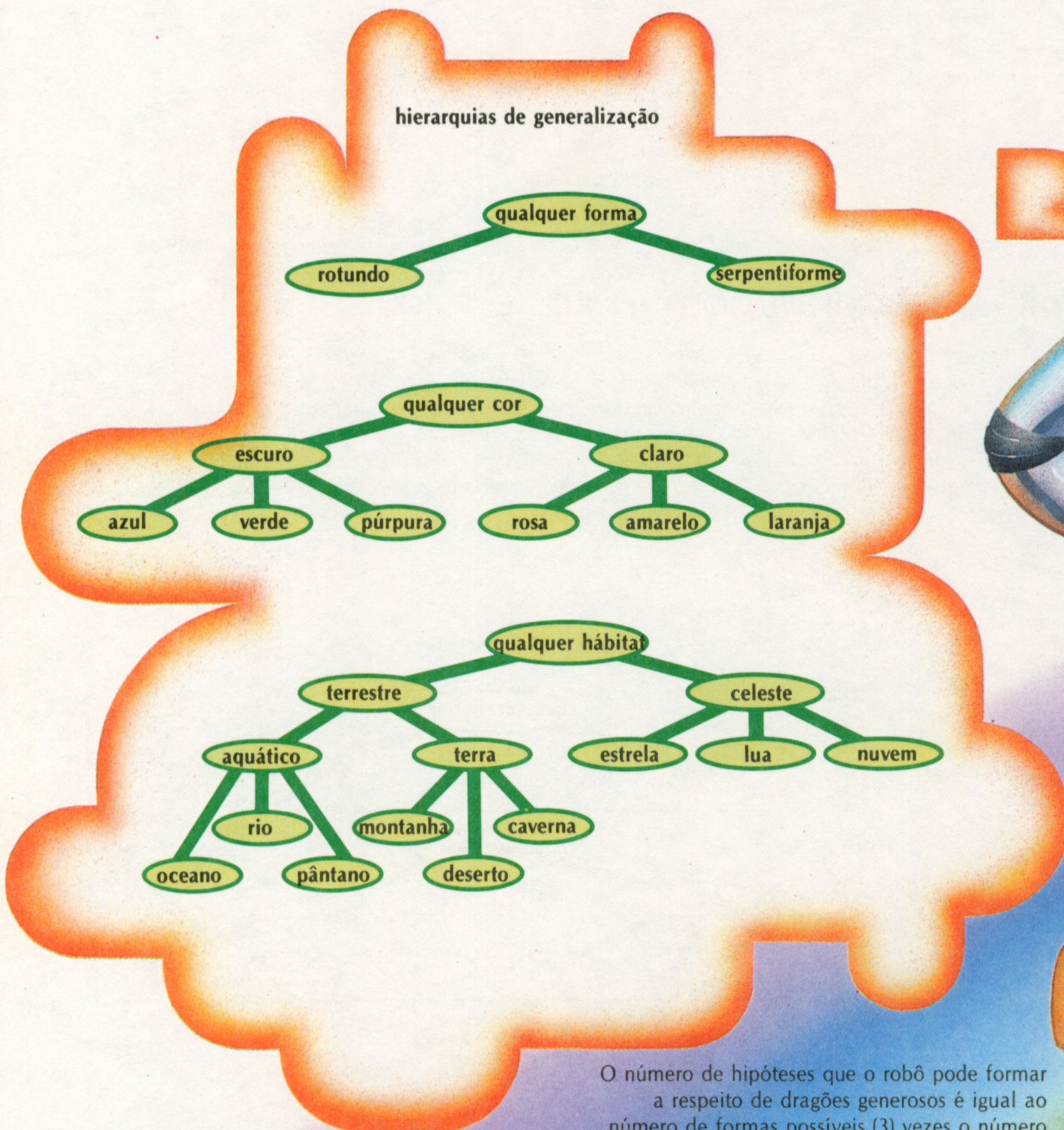


Uma proliferação de possibilidades

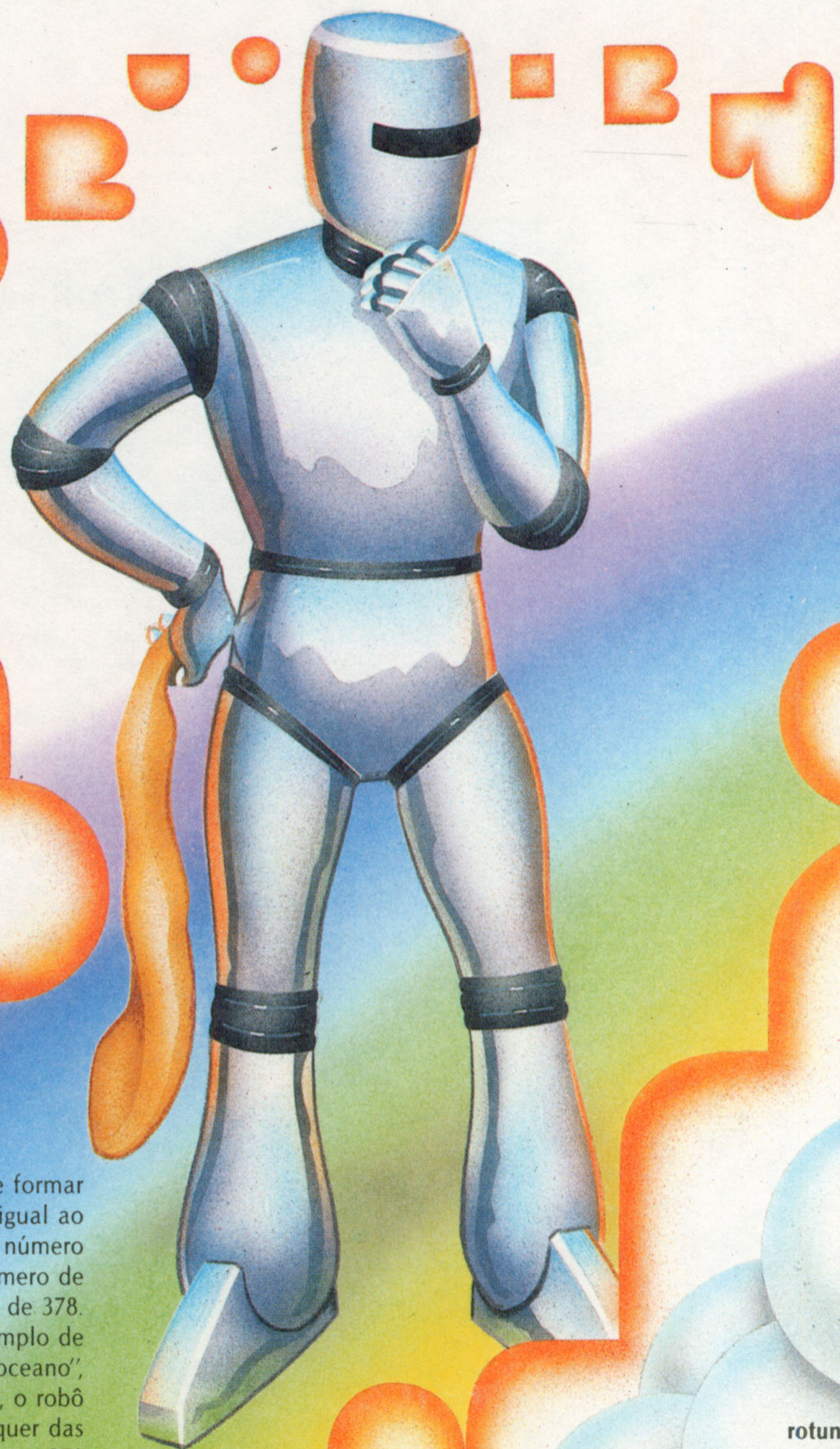
Os atributos nas hierarquias de generalização, que fazem parte do conhecimento inicial do robô (*abaixo*), servem de matéria-prima para a tarefa de aprendizagem, pois definem as características que se supõe relevantes. Como terá de encontrar alguns dragões de verdade (e como ele não tem noções preconcebidas a respeito dos atributos necessários para descrever dragões generosos), o robô considera todas essas informações como igualmente válidas para descrever dragões generosos.

As generalizações que o robô faz aqui — conceitos que ele sustenta serem hipóteses — incluem desse modo todas as combinações possíveis de atributos de forma, cor e hábitat apresentados ao robô. Essas combinações vão das mais amplas às mais estreitas. Por exemplo, a generalização “qualquer forma, qualquer cor, qualquer hábitat” cobre todos os dragões possíveis, ao passo que “rotundo, escuro, celeste” cobre um pequeno subconjunto da suposta população de dragões, e “serpentiforme, amarelo, rio” descreve um número ainda menor de dragões.

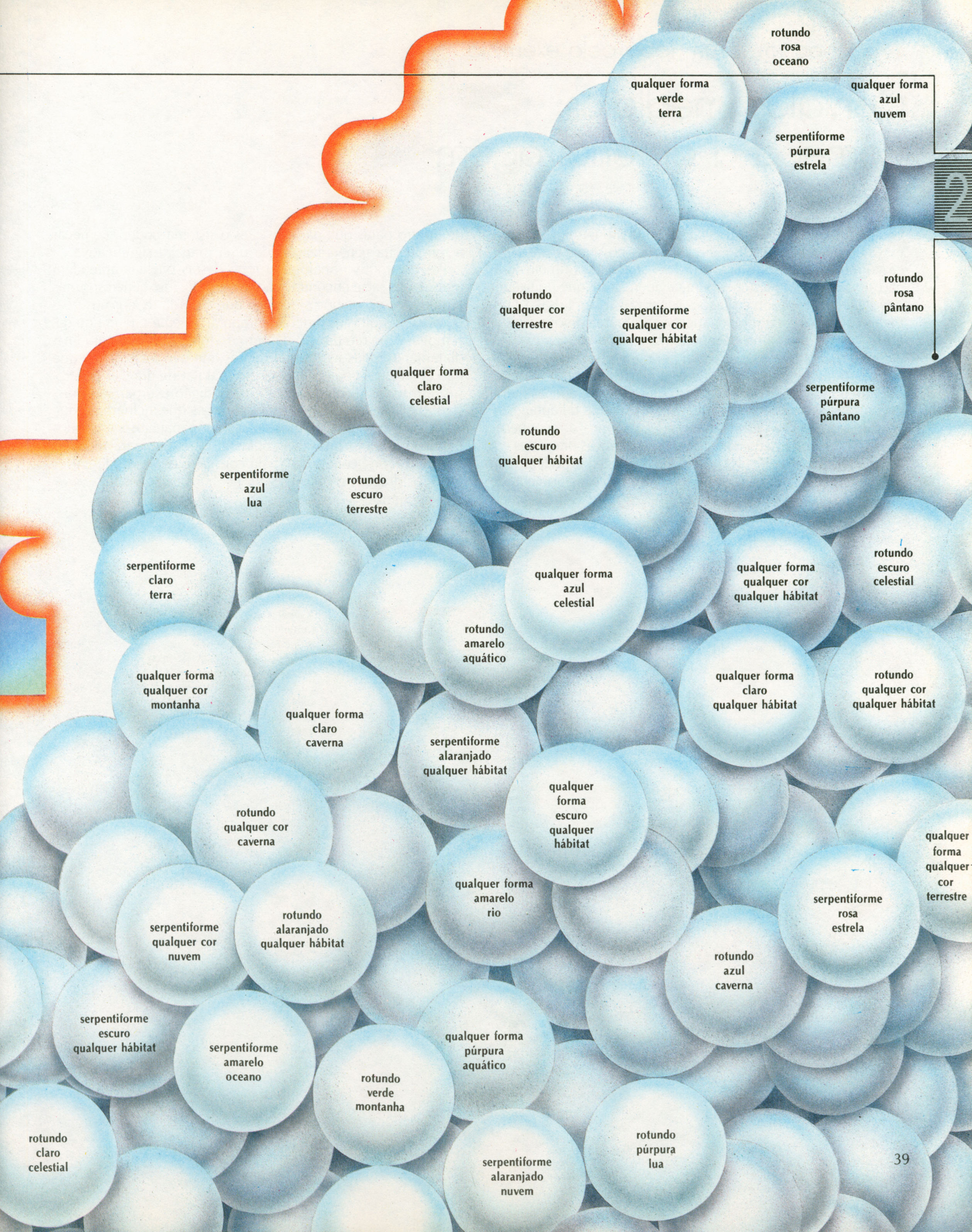
Entretanto, nem todos os dragões são generosos; se o robô deve realizar sua tarefa de desempenho, precisa aprender depressa a distinguir dragões generosos de gananciosos. E faz isso por meio de um processo de raciocínio indutivo.



O número de hipóteses que o robô pode formar a respeito de dragões generosos é igual ao número de formas possíveis (3) vezes o número de cores possíveis (9) vezes o número de habitats possíveis (14), o que dá um total de 378. Até que encontre seu primeiro exemplo de dragão — digamos, “rotundo, escuro, oceano”, ou “serpentiforme, laranja, estrela” —, o robô não pode começar a eliminar qualquer das possibilidades; deve reter todas as 378 combinações como generalizações plausíveis de dragões generosos.



rotundo
claro
terra



rotundo
rosa
oceano

qualquer forma
verde
terra

qualquer forma
azul
nuvem

serpentiforme
púrpura
estrela

rotundo
rosa
pântano

rotundo
qualquer cor
terrestre

serpentiforme
qualquer cor
qualquer hábitat

qualquer forma
claro
celestial

serpentiforme
púrpura
pântano

rotundo
escuro
qualquer hábitat

serpentiforme
azul
lua

rotundo
escuro
terrestre

rotundo
escuro
celestial

serpentiforme
claro
terra

qualquer forma
azul
celestial

qualquer forma
qualquer cor
qualquer hábitat

rotundo
amarelo
aquático

qualquer forma
claro
qualquer hábitat

rotundo
qualquer cor
qualquer hábitat

qualquer forma
qualquer cor
montanha

qualquer forma
claro
caverna

serpentiforme
alaranjado
qualquer hábitat

qualquer
forma
escuro
qualquer
hábitat

qualquer
forma
qualquer
cor
terrestre

rotundo
qualquer cor
caverna

qualquer forma
amarelo
rio

serpentiforme
rosa
estrela

serpentiforme
qualquer cor
nuvem

rotundo
alaranjado
qualquer hábitat

rotundo
azul
caverna

serpentiforme
escuro
qualquer hábitat

serpentiforme
amarelo
oceano

qualquer forma
púrpura
aquático

rotundo
verde
montanha

rotundo
claro
celestial

serpentiforme
alaranjado
nuvem

rotundo
púrpura
lua

Contatos imediatos: começa a aprendizagem

Quando o robô começa sua viagem, também tem início o processo de reduzir suas 378 generalizações possíveis a uma única correta. Ele se aproxima, sem hesitar, de seu primeiro dragão, pois ainda não tem nenhuma evidência de que quaisquer das generalizações sobre dragões generosos sejam falsas. O primeiro contato e cada um dos sucessivos atuará como realimentação para o programa de aprendizagem do robô.

Um programa de AI planejado para aprender por exemplos recebe usualmente casos positivos e negativos como entrada, pois um programa de aprendizagem que encontra apenas exemplos positivos irá supergeneralizar. Se um sistema de aprendizagem de máquina se destina a aprender, digamos, a reconhecer cães, mas é confrontado com figuras de dez raças diferentes e a cada uma diz-se a ele: "Isto é um cão", o sistema poderá concluir que todos os animais de quatro patas são cães. Somente se lhe forem mostradas figuras de elefantes, gatos e outros animais

de quatro patas que não são cães o sistema poderá começar a refinar sua generalização de uma maneira significativa.

Por um motivo análogo, o robô destas páginas aprenderá mais depressa se encontrar não apenas dragões generosos mas também os gananciosos, mesmo que a perda de ouro usurpado por estes últimos dificulte temporariamente a tarefa de desempenho. Se um dragão dá ouro ao robô, este eliminará imediatamente todas as generalizações que não incluam ou que não coincidam com a forma, cor e hábitat de dragões generosos, em todos os cálculos. Se um dragão usurpar seu ouro, o robô eliminará todas as generalizações que incluam ou que coincidam com os atributos do dragão ganancioso, em todos os cálculos. No final das contas, o robô eliminará todas as generalizações, com exceção da correta, que deverá descrever todos os dragões generosos — e nenhum dos gananciosos — que encontrou.

rotundo rosa montanha	qualquer forma rosa montanha	rotundo claro montanha	qualquer forma claro montanha	rotundo qualquer cor montanha	qualquer forma qualquer cor montanha
rotundo rosa terra	qualquer forma rosa terra	rotundo claro terra	qualquer forma claro terra	rotundo qualquer cor terra	qualquer forma qualquer cor terra
rotundo rosa terrestre	qualquer forma rosa terrestre	rotundo claro terrestre	qualquer forma claro terrestre	rotundo qualquer cor terrestre	qualquer forma qualquer cor terrestre
rotundo rosa qualquer hábitat	qualquer forma rosa qualquer hábitat	rotundo claro qualquer hábitat	qualquer forma claro qualquer hábitat	rotundo qualquer cor qualquer hábitat	qualquer forma qualquer cor qualquer hábitat

O robô se aproxima do primeiro dragão — que é rotundo e rosa e vive nas montanhas — pede ouro e recebe. Com base nesse comportamento generoso, o robô elimina 354 generalizações, tais como "serpenteforme, amarelo, celeste", que não igualam ou abrangem os atributos do dragão em todos os cálculos e que, portanto, não podem ser a generalização correta para dragões generosos. São ainda plausíveis apenas 24 generalizações — variando do estreito "rotundo, rosa, montanha" ao amplo "qualquer forma, qualquer cor, qualquer hábitat".



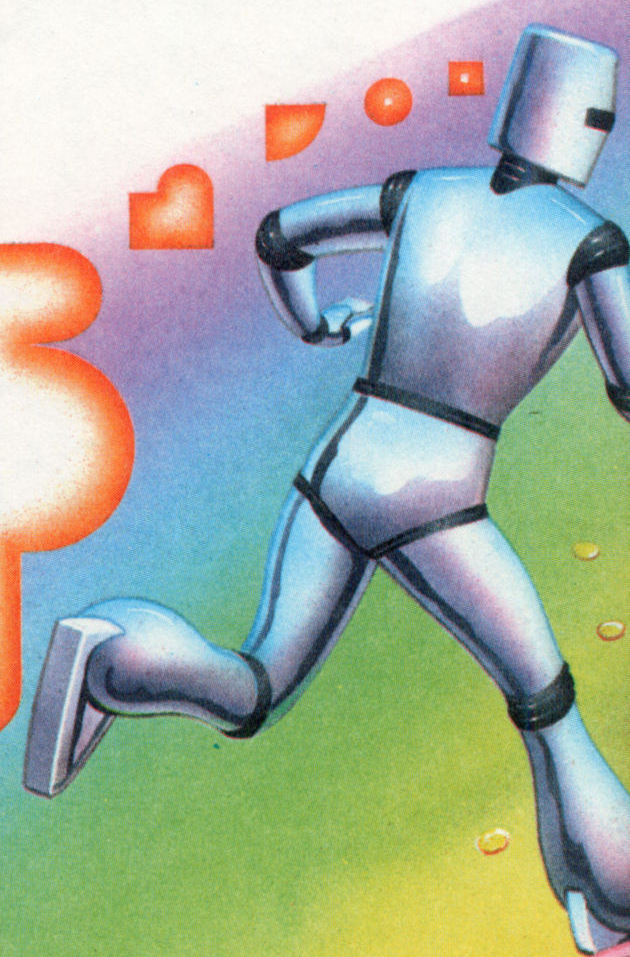
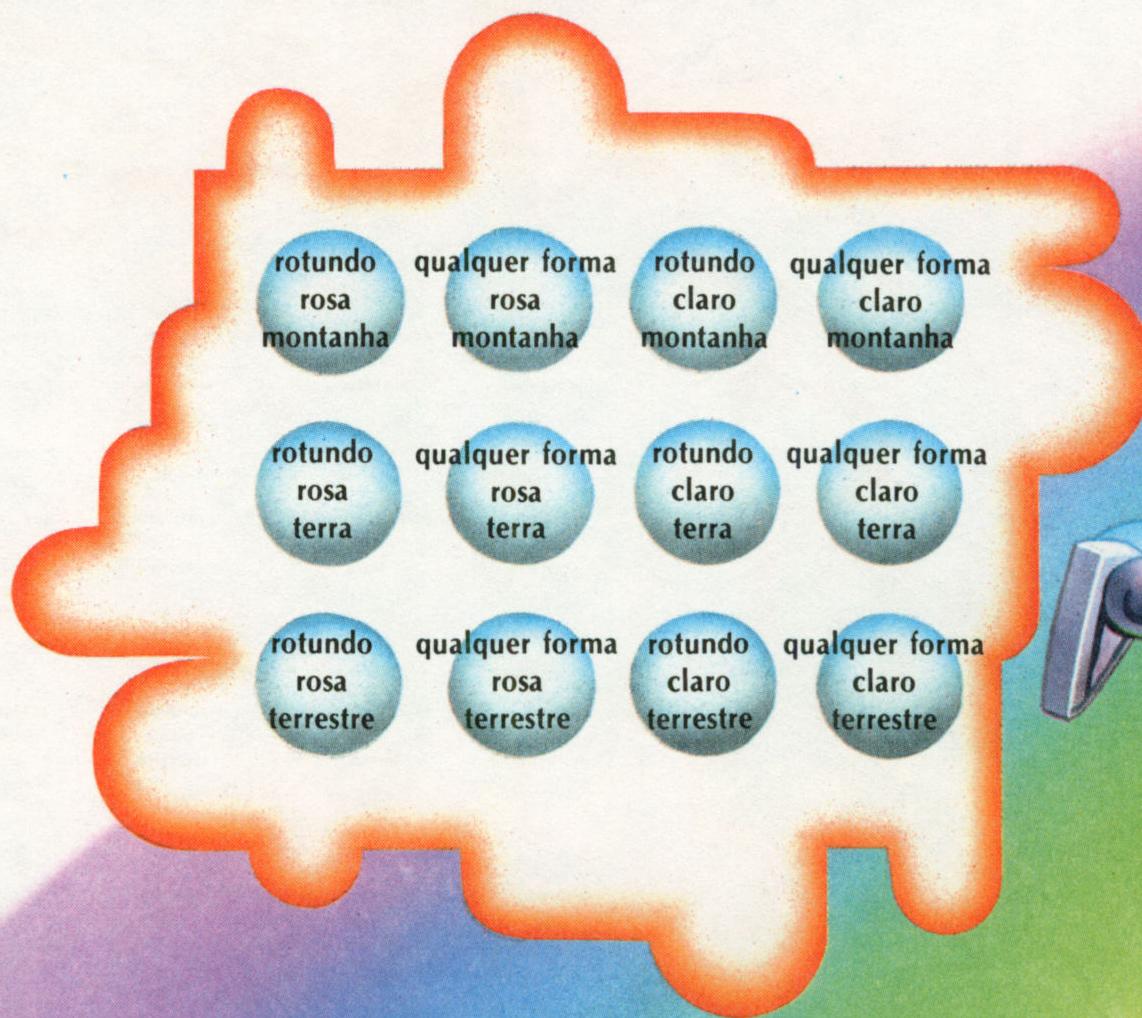


rotundo rosa montanha	qualquer forma rosa montanha	rotundo claro montanha	qualquer forma claro montanha
rotundo rosa terra	qualquer forma rosa terra	rotundo claro terra	qualquer forma claro terra
rotundo rosa terrestre	qualquer forma rosa terrestre	rotundo claro terrestre	qualquer forma claro terrestre
rotundo rosa qualquer hábitat	qualquer forma rosa qualquer hábitat	rotundo claro qualquer hábitat	qualquer forma claro qualquer hábitat

A seguir, o robô aproxima-se de um dragão rotundo, púrpura, habitante de montanha; como a descrição do dragão ajusta-se a várias das generalizações restantes acerca de dragões generosos, é possível que esse dragão seja generoso. Mas, ao invés de dar ouro ao robô, o dragão lhe rouba um pouco. Com base nesse comportamento ganancioso, o robô elimina oito generalizações que descrevem o primeiro dragão mas também descrevem o segundo. Apenas dezesseis generalizações sobre dragões generosos permanecem viáveis.

O robô encontra a seguir um dragão serpentiforme, amarelo, morador dos pântanos, e pede ouro. (Se o dragão fosse azul em vez de amarelo, o robô o teria evitado.) A partir do momento em que o dragão se revelou generoso, o robô descartou todas as generalizações que não o abrangiam, deixando apenas uma — “qualquer forma, claro, terrestre” — que se aplica a ambos os dragões generosos que o robô encontrou, e a nenhum dos dragões gananciosos. Se o método da aprendizagem por exemplos é bem-sucedido, esta será a generalização correta, e o robô deverá obter facilmente ouro em quantidade suficiente para encher o pote no fim do arco-íris.

O próximo dragão que o robô vê é rotundo e rosa e vive nas estrelas. Como esse dragão satisfaz a quatro das generalizações restantes, o robô aproxima-se dele e lhe pede ouro. Mas houve engano: o dragão é ganancioso. Escapando com sua sacola mais vazia, o robô elimina mais quatro generalizações. Das doze generalizações restantes, todas se ajustam ao único dragão generoso que o robô encontrou até agora, e nenhuma se aplica a qualquer um dos dragões gananciosos encontrados.





qualquer forma
claro
terrestre

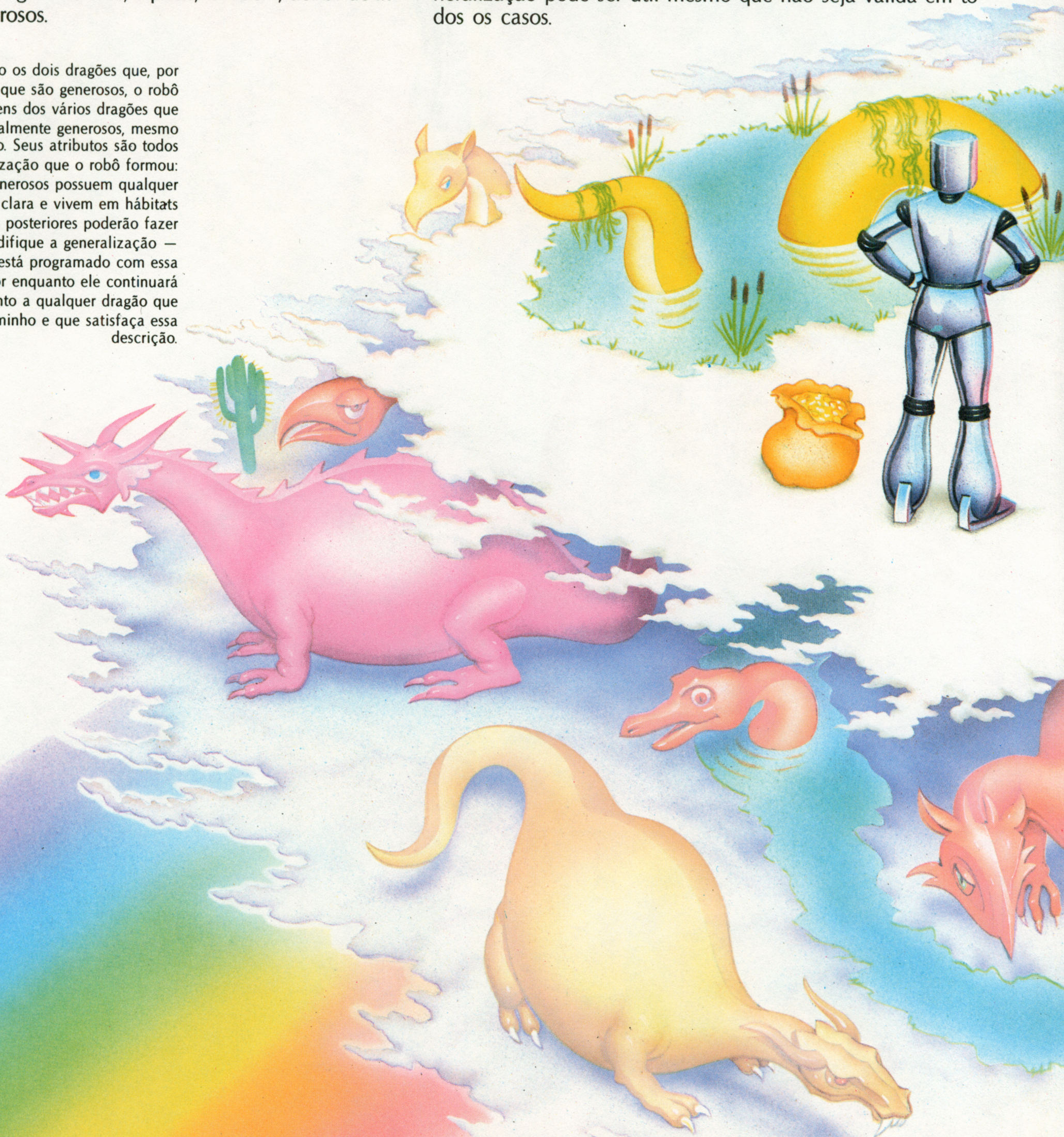
2

Exame da hipótese final

Depois de encontrar quatro exemplos, o robô formou uma generalização sobre dragões generosos. De agora em diante, ele só pedirá ouro a dragões que satisfaçam a generalização "qualquer forma, claro, terrestre" e evitará todas as outras. Entretanto, à semelhança de qualquer outro sistema que aprende por exemplos, o robô corre o risco de ter supergeneralizado ou superespecializado: de fato, a generalização do robô pode cobrir alguns dragões gananciosos; e pode, também, deixar de incluir alguns generosos.

Contemplando os dois dragões que, por experiência, ele sabe que são generosos, o robô é cercado por imagens dos vários dragões que ele presume serem igualmente generosos, mesmo sem tê-los visto. Seus atributos são todos cobertos pela generalização que o robô formou: que os dragões generosos possuem qualquer forma, têm cor clara e vivem em habitats terrestres. Experiências posteriores poderão fazer com que o robô modifique a generalização — uma vez que ele está programado com essa capacidade —, mas por enquanto ele continuará procurando ouro junto a qualquer dragão que encontrar no caminho e que satisfaça essa descrição.

A supergeneralização e a superespecialização podem ser causadas por dados imperfeitos, ou afetados por ruído. Talvez o dragão celeste que o robô encontrou seja normalmente generoso, mas estava de mau humor naquele dia. Seu comportamento ganancioso poderia constituir uma das assim chamadas amostras negativas falsas. Inversamente, um dos dragões que se comportaram generosamente poderia ser um positivo falso. Podem ocorrer outros erros devido a suposições iniciais incorretas residentes no sistema — por exemplo, a de que a ganância e a generosidade admitem ser julgadas com base na forma, cor e habitat. Os algoritmos utilizados nos sistemas de aprendizagem de máquina podem minimizar o risco desses tipos de erros. Frequentemente, uma generalização pode ser útil mesmo que não seja válida em todos os casos.

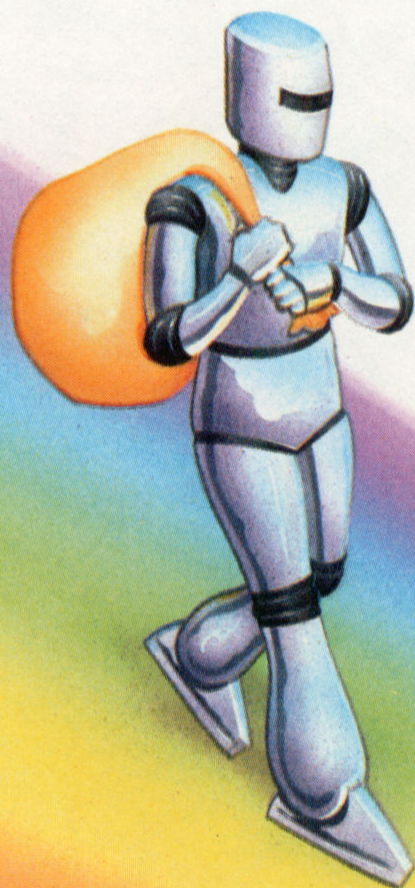




Outros caminhos para chegar ao mesmo fim

Depois de fazer uma generalização viável sobre dragões generosos, o robô está pronto para completar sua tarefa de desempenho, enchendo de ouro o pote no fim do arco-íris. Outras técnicas de aprendizagem de máquina poderiam ser igualmente utilizadas, embora com resultados diferentes. Por exemplo, o robô poderia ter aprendido por rotina, simplesmente memorizando a descrição e o comportamento de cada dragão que encontrasse. Esse método não sofisticado seria útil apenas se o robô encontrasse o mesmo dragão mais de uma vez.

EXEMPLO



ROTINA

ANALOGIA

A aprendizagem por instrução teria envolvido o armazenamento de grandes quantidades de informações sobre o assunto dragão na memória do robô e a programação da máquina para descobrir padrões lógicos de comportamento nos dragões gananciosos e nos generosos. A aprendizagem por analogia somente seria possível se o robô iniciasse sua tarefa de aprendizagem equipado com conhecimento prévio sobre algum assunto semelhante. Se o robô já tivesse aprendido algo sobre, digamos, magos gananciosos e generosos, isso poderia ser aplicado ao estudo dos dragões.

Uma das técnicas mais sofisticadas de aprendizagem de máquina é a aprendizagem por descoberta. Esse método envolve a exploração de um assunto; não é especificada nenhuma tarefa de aprendizagem ou de desempenho em particular. A aprendizagem por descoberta requer a capacidade de fazer inferências, reconhecer padrões, formar e testar teorias e trabalhar com dados inconclusivos. Todas essas são habilidades das quais a mente humana é adepta, mas que não são fáceis de escrever num programa de computador.



DESCOBERTA

INSTRUÇÃO





Vendo e prevendo

Embora relativamente jovem, o campo da inteligência artificial já produziu sua parcela de folclore. De acordo com uma história contada com frequência, em certa ocasião, na década de 1960, Marvin Minsk, um dos pioneiros de AI, convocou um estudante novato do MIT, oferecendo-lhe alegremente um projeto a ser completado durante o verão. A tarefa do estudante: estudar e resolver o problema relacionado com a percepção visual da máquina. Tal empresa poderia parecer bastante simples na época. Os pesquisadores de AI, em sua maioria, estavam preocupados com coisas que, nos seres humanos, requerem um rigoroso esforço mental — tarefas de manipulação de símbolos envolvendo lógica, regras “se-então” e jogos de tabuleiro. A maioria do pessoal que trabalhava nesse campo parecia quase menosprezar o sensorio humano e as habilidades motoras do ver, ouvir, sentir, agarrar e locomover-se. Achavam que essas atividades pouco, ou nada, têm a ver com a inteligência genuína. A visão humana, por exemplo, ocorre instantaneamente e sem ser acompanhada de pensamento facilmente discernível.

Entretanto, como o infeliz estudante da lendária história de AI logo percebeu, replicar a visão por meio de uma máquina é um problema que não conduz a soluções fáceis. Por isso, os pesquisadores da inteligência artificial ainda precisam desenvolver uma máquina que consiga aproximar-se das habilidades sensoriais e motoras até mesmo de um insignificante mosquito quando voa, se alimenta, procria e luta com um ambiente complexo, em mudança constante.

Estender uma ponte que cubra o hiato considerável entre o poder de processamento dos computadores e o das criaturas vivas é crucial para que ocorram avanços efetivos no campo da inteligência artificial. Qualquer inteligência, seja ela natural ou artificial, deve ser capaz de lidar com o mundo real — um mundo que consiste, em sua maior parte, de fenômenos físicos frequentemente ambíguos em vez de símbolos e lógica precisos. Desse modo, uma das mais altas prioridades das pesquisas sobre a AI é a de criar novos métodos e hardwares que capacitem máquinas não apenas a ver, ouvir e sentir, mas a compreender suas percepções. Um sistema de visão computadorizada destinado, por exemplo, a reconhecer certos animais selvagens terá de ser adaptado com câmeras e programado com descrições dos animais que ele precisará identificar. A seguir, o sistema deverá ser programado com conceitos como grande porte, cor cinzenta e quadrúpede — e deverá ser informado sobre quais são os animais que essas características descrevem — a fim de que possa reconhecer um elefante.

Embora os programas que permitem aos computadores ver e ouvir venham sendo aperfeiçoados nas últimas décadas, a tradução de dados sensoriais em conceitos permanece desajeitada e dispendiosa. Algumas máquinas usadas na produção industrial podem reconhecer a silhueta de uma determinada peça que se move ao longo de uma linha de montagem bem iluminada. Microcomputadores simples, equipados com microfones e outros hardwares e softwares, podem, em resposta a um comando oral, ligar lâmpadas, discar números de telefone e até mesmo emitir uma resposta falada. Mas todos esses sistemas são rigidamente especia-

A despeito de todos os esforços dos cientistas em computadores, a criação de uma inteligência artificial que se aproxime, ainda que de longe, do nível humano permanece um objetivo tão remoto quanto as estrelas; a AI aguarda descobertas que abram novas janelas nos vastos domínios do pensamento humano.

lizados; só conseguem perceber objetos isolados, delineados contra fundos vazios e muito iluminados, e reconhecer vocabulários restritos de palavras estocadas e frases simples.

O ESTÁGIO INFANTIL DO RECONHECIMENTO DA FALA

Na década de 1950, cientistas dos Laboratórios Bell tentaram construir máquinas que pudessem compreender a fala — ou pelo menos reconhecer e responder a sons específicos —, num esforço para permitir às pessoas que fazem chamadas telefônicas dizer, em voz alta, num receptor, o número de telefone desejado, em vez de discá-los. Mas os pesquisadores logo perceberam que seu objetivo ainda estava distante e substituíram-no por projetos mais factíveis. Outros pesquisadores construíram alguns sistemas rudimentares para reconhecimento da fala, durante a década de 1960. Entre eles estava um estudante graduado da Universidade de Stanford, chamado Raj Reddy, que chegou aos Estados Unidos vindo de uma aldeia indiana. Sua especialidade básica era a análise numérica, que envolve o uso de computadores para achar a solução de grandes problemas matemáticos; mas ele desenvolveu um profundo interesse pela inteligência artificial.

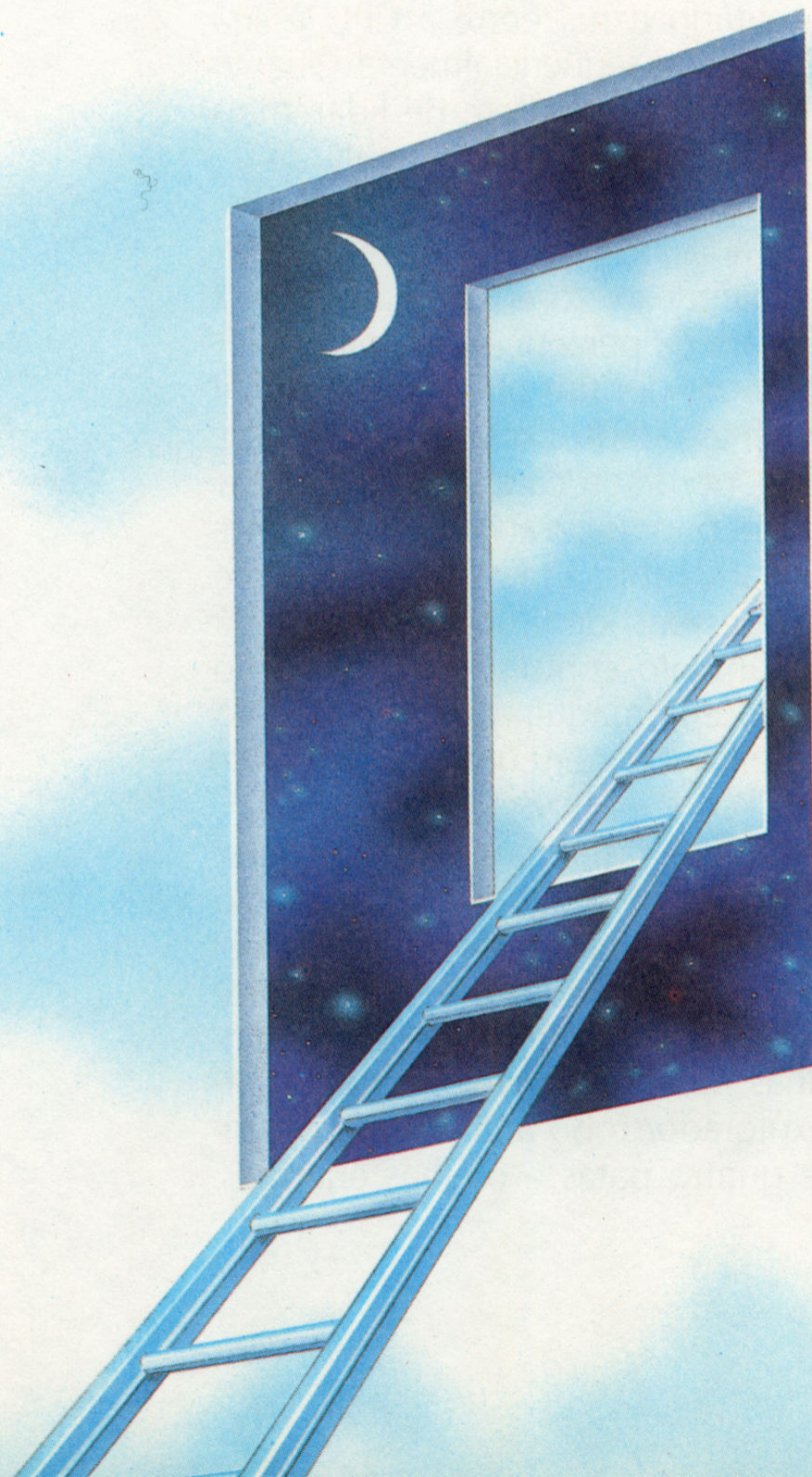
Os esforços de Reddy para a criação de uma máquina apta ao reconhecimento da fala não conseguiram ficar muito acima do nível da conversa entre bebês, mas ele se recusou a desanimar. Em 1970, foi para a Universidade Carnegie-Mellon, a leste de Stanford. Um ano após, ele e vários de seus novos colegas que também pesquisavam a AI concordaram em discutir as qualidades que seriam necessárias a um sistema praticável de reconhecimento de fala por máquina. A sessão foi presidida por Allen Newell, cujo trabalho pioneiro, realizado junto com Herbert Simon e J. C. Shaw, sobre o Logic Theorist e o General Problem Solver insuflara energia na comunidade da inteligência artificial mais de uma década antes. Desse conclave de peritos surgiu um projeto de 15 milhões de dólares, apelidado SUR — sigla para *Speech Understanding Research*, isto é, "Pesquisas para a Compreensão da Fala". Os objetivos foram cuidadosamente sublinhados: demonstrar capacidade de compreender cerca de 90 por cento da fala comum, contínua (ao contrário da compreensão de palavras isoladas), a respeito de algum assunto restrito, usando um vocabulário de, no mínimo, mil palavras. Finalmente, várias organizações ofereceram programas à apreciação, mas apenas dois deles — ambos produzidos por equipes da Carnegie-Mellon, encabeçadas por Raj Reddy — satisfariam os objetivos da reunião.

Um dos programas bem-sucedidos, o Hearsay II, representou um avanço da maior importância em comparação com os esforços iniciais de Reddy, quando ainda trabalhava na Stanford, na década de 1960. O novo programa foi equipado com doze subprogramas para selecionar e coordenar diferentes tipos de informações contidas num sinal da fala. Alguns desses subprogramas eram especializados em fonética e em fonêmica — isto é, em representação dos sons da fala e em regras descritivas dos componentes que constroem a fala numa dada linguagem. Outro era especializado em prosódia, ou regras para interpretar variações na tensão e na entonação; um terceiro dizia respeito à morfêmica, isto é, às regras para as variações das palavras — plurais, por exemplo. Um subprograma de sintaxe lidava com regras para uma satisfatória formação das sentenças; um subprograma pelejava com os significados das palavras. Outro subprograma lidava com a pragmática — dialetos e regras não-escritas do discurso.

Cada um dos especialistas do Hearsay II ocupava-se sucessivamente apenas com pequenas porções de uma sentença — algumas palavras ou, talvez, uma oração simples. Cada subprograma afixava, num “quadro negro”, suas hipóteses sobre aquilo que ele pensava ter ouvido; esse “quadro negro” era um depósito central na memória, que outros subprogramas podiam então consultar para avaliar que hipóteses pareciam mais plausíveis. Por exemplo, o subprograma para fonética poderia presumir que ouvira uma certa combinação de fonemas. Então, um módulo sintático opinaria se era ou não provável que tal palavra ocorresse em tal posição na sentença.

As inovações que Reddy e seus colegas instalaram no Hearsay II — fontes de múltiplos conhecimentos, o “quadro negro” e a alteração entre dados sensoriais e expectativas informadas — tornara-se desde essa época um padrão para muitas outras aplicações da AI. Essas técnicas têm sido particularmente úteis no intimidante campo da visão da máquina; aparentemente, separar um objeto visual de um confuso plano de fundo onde está imerso, e identificá-lo — coisas que os seres humanos e os animais inferiores realizam automaticamente numa fração de segundo — requer, no mínimo, tantos tipos de conhecimentos diferentes quanto a compreensão de uma sentença falada.

Responder aos desafios da percepção sensorial poderá resultar em diversas aplicações inestimáveis, tais como máquinas capazes de converter, com fidelidade, pa-



lavras faladas em escritas, e vice-versa. Um sistema de visão capaz de reconhecer objetos poderia muito bem estar a caminho da autonomia, uma vez que também fosse equipado para aprender a partir do ambiente que o cerca. Mas devido ao fato de que uma parcela tão grande da percepção e da aprendizagem humanas ocorre abaixo do nível da consciência, os pesquisadores têm de enfrentar a tarefa aparentemente insuperável de tentar programar máquinas para simular habilidades que os próprios pesquisadores não conseguem explicar.

No entanto, mesmo que as pessoas que trabalham em AI soubessem precisamente como escrever os programas para a aprendizagem e a percepção sensorial da máquina, é improvável que os computadores convencionais venham a ser tão rápidos quanto as pessoas em certos tipos de processos mentais. Por exemplo, para interpretar uma única cena o sistema visual humano utiliza talvez 10 bilhões de células cerebrais, para que cada uma delas execute 1 bilhão de cálculos com números grandes, a cada décimo de segundo. Isso corresponde a várias vezes a velocidade dos mais rápidos supercomputadores comerciais.

CONGESTIONAMENTO DE TRÁFEGO NA VIA DE DADOS

Para muitas aplicações em potencial da AI, particularmente aquelas que envolvem percepção sensorial e respostas oportunas a eventos do mundo real, tais computadores operam devagar demais. Eles são inerentemente limitados por aquilo que John Backus, cientista em computadores, da IBM — e líder da equipe que desenvolveu a extensamente usada linguagem FORTRAN de programação —, identificou como o “gargalo de Von Neumann”: o solitário canal entre a CPU e sua memória, ao longo do qual os dados e as instruções fluem seqüencialmente. Quanto maior for o número de elementos com os quais um programa tem de lidar, mais tempo tal programa exigirá para encontrar uma solução, pois cada parcela de informação deve viajar nos dois sentidos entre a memória e a CPU, à medida que o computador executa suas operações, uma de cada vez.

A natureza dos computadores seriais cria um paradoxo, como assinalou Scott Fahlmann no fim da década de 1970. Quando se trata de percepção, quanto maior for o número de características identificáveis possuídas por um objeto, mais difícil se torna para o computador obter um reconhecimento rápido, pois a máquina funciona testando todas as possibilidades introduzidas por cada característica. Os seres humanos, ao contrário, parecem achar que, quanto mais características distintivas um objeto possui, mais fácil é identificá-lo. Para Fahlmann, esse fato sugere uma diferença fundamental na maneira como as pessoas identificam coisas. “Há dois tipos de pensamento”, disse ele. “Há o pensamento comum linear, como o que leva à resolução de um problema aritmético, e uma máquina serial modela-o bem. Então, ocorre o reconhecimento. É muito complicado fazer com que isso aconteça numa máquina serial.”

Infelizmente para os pesquisadores que usam máquinas seriais, a maior parte das atividades mentais humanas — o reconhecimento de padrões e o raciocínio acerca deles, a capacidade de fazer rapidamente julgamentos, associações, generalizações, analogias e coisas semelhantes — parece ser o produto, em grande parte inconsciente, desse segundo tipo de pensamento. Presumivelmente, nenhum computador serial jamais será capaz de fazer escolhas num tal aglomerado de informações, com a rapidez que, por exemplo, um explorador robô autônomo necessitaria. Espiando uma forma grande, cinzenta, de quatro patas e orelhas enormes,

andando depressa por um caminho de selva, o robô estaria numa perigosa desvantagem se tivesse de descobrir e combinar listas de propriedades à maneira serial antes de concluir que está a ponto de ser amassado por um elefante.

Além de ter dificuldades no reconhecimento de objetos, computadores de qualquer tipo também não conseguem lidar facilmente com coisas como tempo, causa e efeito, e os assim chamados conceitos indistintos, carentes de significado preciso ou constante. Um exemplo de conceito indistinto é a palavra "longe" que pode referir-se a qualquer coisa, de escalas de distância atômicas e galácticas. O planejamento de um programa que permita a um computador operar sobre regras vagas e definições imprecisas é um dos desafios centrais da AI.

PROJETOS RADICAIS PARA OBJETIVOS AMBICIOSOS

Dada a complexidade dos problemas da AI, a maioria dos pesquisadores, no início da década de 1980, concluiu que qualquer inteligência de máquina digna do nome terá, provavelmente, necessidade de novos tipos de computadores. O Projeto para Computadores de Quinta Geração, japonês, lançado em 1982 e sobre o qual se fez extensa publicidade, baseava-se em algumas idéias emergentes sobre máquinas não-Von Neumann. O objetivo declarado desse ambicioso empreendimento é o de construir computadores que operem vários milhares de vezes mais depressa do que as mais rápidas máquinas Von Neumann, e inventar software que permita a esses avançados computadores compreender as linguagens falada e escrita, aprender, raciocinar e, em geral, executar a maior parte das tarefas que requeiram a inteligência humana.

A idéia principal por trás desse e de outros esforços semelhantes denomina-se processamento paralelo. Dividindo o trabalho de processamento entre muitos dispositivos pequenos, tais como os baratos chips de silício, o paralelismo oferece a esperança de se obter saltos quânticos na capacidade de processamento a baixo custo. Talvez o principal atrativo dessa abordagem seja o fato de que ela imita a própria natureza. Enquanto um único processador de computador eletrônico opera milhares de vezes mais depressa que seu equivalente, o neurônio individual, redes de neurônios executam muitos tipos de tarefas, especialmente não-numéricas, milhares de vezes mais depressa que os computadores. Isso porque os neurônios são ligados uns aos outros de modo que os neurônios individuais podem, simultaneamente, resolver diferentes problemas ou executar funções motoras e sensoriais separadas — em paralelo em vez de serialmente.

Tão brilhantes são as promessas do processamento eletrônico em paralelo que alguns pesquisadores têm até mesmo revivido as idéias dos ciberneticistas — as análises de baixo para cima dos padrões de sinais e de aprendizagem, usando redes de neurônios simulados. Esses esforços foram estimulados no início da década de 1980 pela publicação de um livro intitulado *Parallel Models of Associative Memory (Modelos Paralelos da Memória Associativa)*, coleção de ensaios editados por Geoffrey Hinton, jovem inglês que estudou AI, psicologia e neurofisiologia, e James Anderson, neurofisiologista da Universidade Brown. Hinton, Anderson e os outros autores que colaboraram para o livro apresentaram argumentos persuasivos mostrando que os assim chamados computadores massivamente paralelos — que tomam por modelo as redes de neurônios — poderiam ser muito melhores, em muitas funções da inteligência, que os computadores Von Neumann.

A diferença de capacidade entre os sistemas biológicos de processamento de

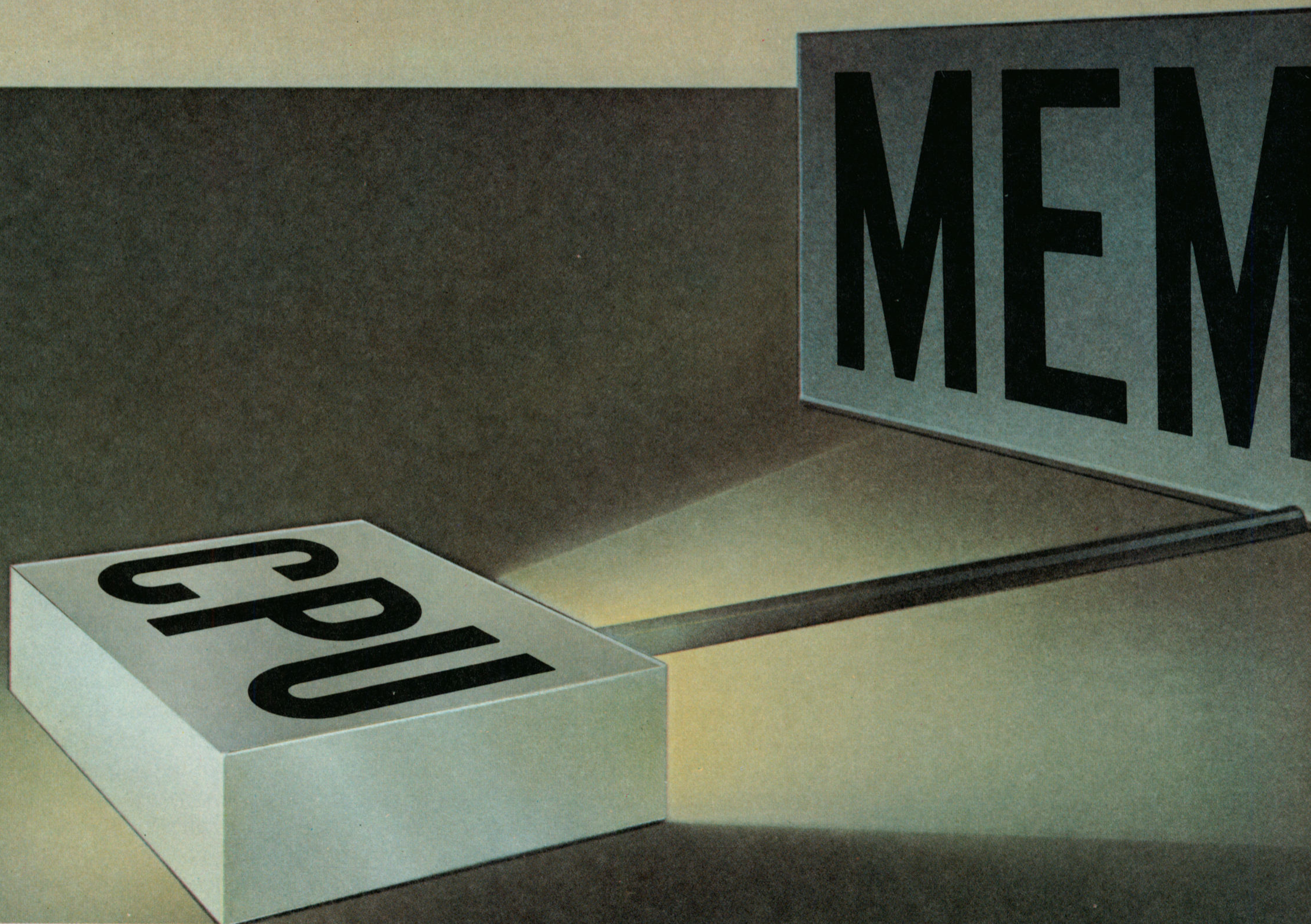
As limitações de uma máquina serial

A despeito dos avanços na tecnologia dos computadores — das válvulas eletrônicas aos transistores, aos circuitos integrados de escala muito grande (VLSI) —, há um aspecto em que os computadores mudaram muito pouco: são ainda, em sua maioria, máquinas seriais, operando de acordo com os princípios definidos por John von Neumann, que asseguram um computador flexível para todos os fins, mas só podem executar uma operação por vez.

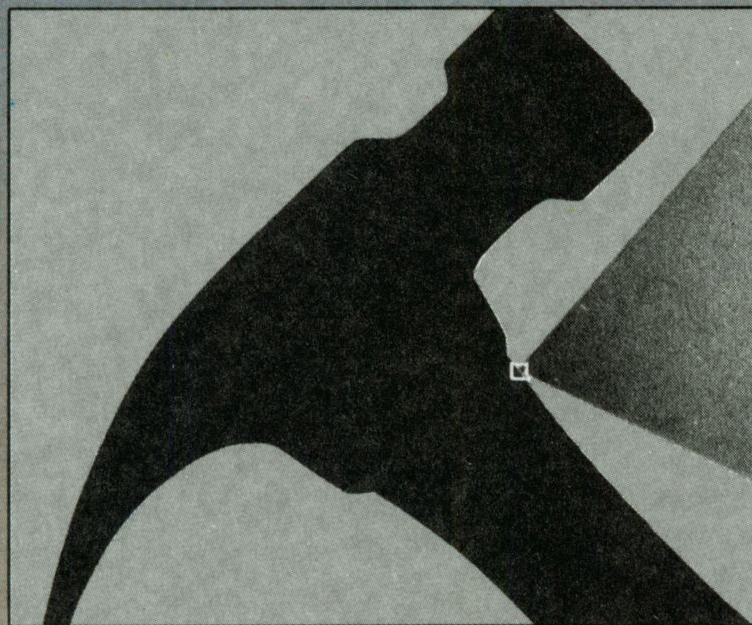
Como é conceitualmente mostrado na figura abaixo, um computador serial consiste numa unidade central de processamento (CPU), que dirige as operações e executa os cálculos; uma memória, que retém instruções e dados pertencentes ao programa corrente; e um condutor eletrônico entre ambos, denominado via de dados. Para executar uma operação simples, como a adição de dois números, a maioria dos computadores tem que efetuar vários passos, que requerem o movimento de dados através dessa via. Entretanto, como ele pode acomodar

apenas uma instrução ou segmento de dados por vez, os passos são obrigados a mover-se seqüencialmente, um após o outro.

Em sua maioria, os problemas de inteligência artificial são muito extensos ou complicados para adequar-se a uma manipulação desse tipo. Independentemente da rapidez com a qual a CPU pode executar operações, o tempo gasto no envio de informações de um lado para o outro através da via de dados é tão grande que a máquina não consegue resolver um problema com a rapidez necessária para que a solução seja útil. Por exemplo, um sistema experimental de visão móvel monitorada por AI, ligado a um computador serial, processa informações a partir de uma imagem digitalizada formada por 250 000 pixels, ou elementos de imagem. Embora use um algoritmo sofisticado para filtrar todos os dados exceto os mais relevantes, o dispositivo só é capaz de contornar um pequeno obstáculo em seu caminho parando por um minuto a cada jarda percorrida, a fim de recomputar sua posição e as dos obstáculos vizinhos.



A grade à direita representa os valores de brilho dos pixels numa área da imagem digitalizada do martelo. Num sistema de visão de máquina, o passo preliminar da detecção de contorno, chamado alisamento, envolve a manipulação dos valores de brilho para eliminar flutuações de pouca importância na intensidade. Aqui, o valor de cada pixel será substituído por uma média dos valores de seus quatro vizinhos. Para executar esse passo sobre os pixels de uma imagem 512 x 480 pixels, um computador serial rápido precisa de cerca de três segundos; a seguir exigirá cerca de sete minutos para executar uma detecção de contorno completa sobre a imagem.



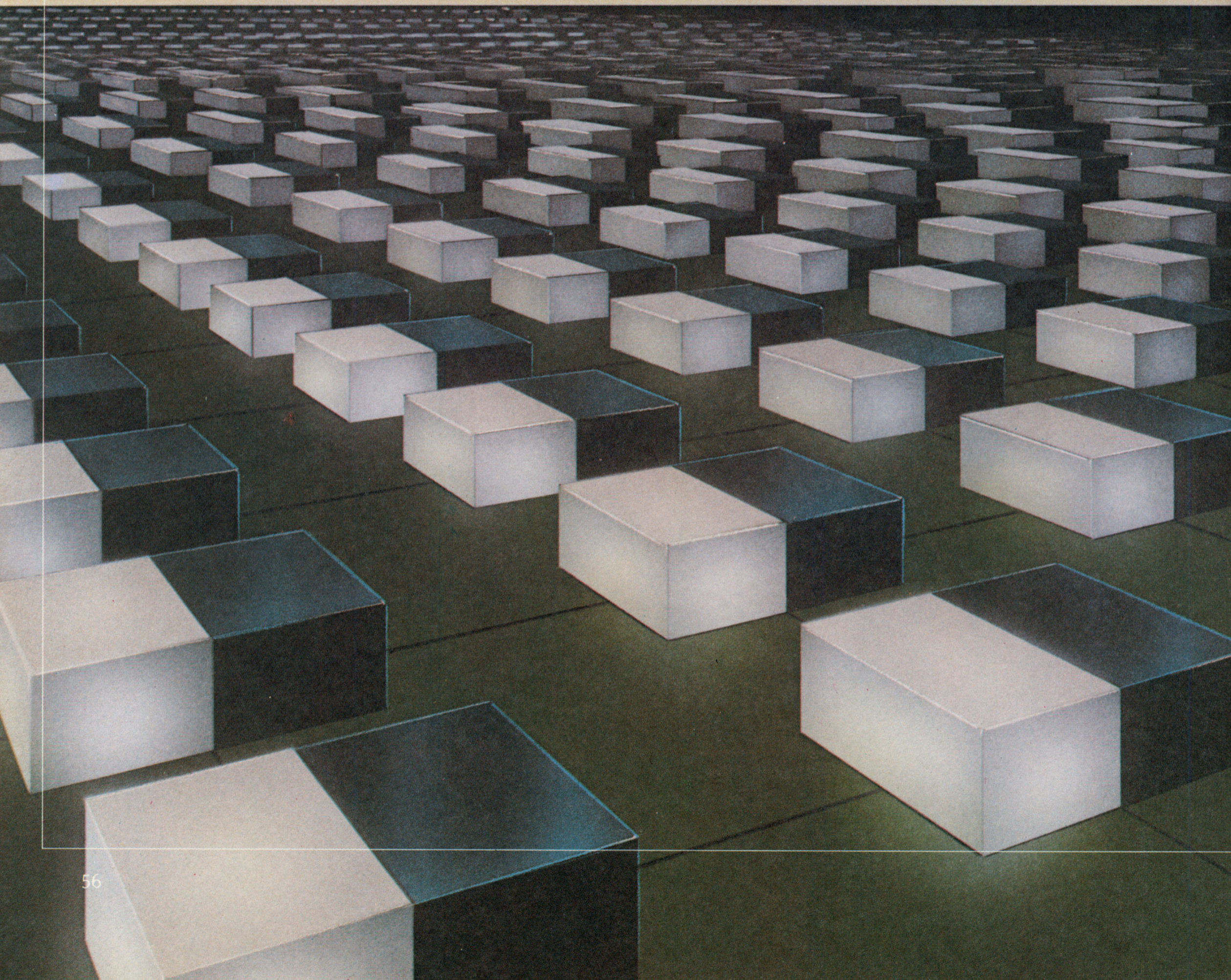
66	76	70	71	76	74
48	55	60	69	71	75
52	60	63	71	70	75
45	45	45	68	67	78
48	49	48	48	65	72
48	48	48	48	48	70

ÓRIA

Paralelismo: uma estratégia para a rapidez

Numa tentativa de romper o assim chamado gargalo de Von Neumann do processamento serial, cientistas experimentaram desenhos de computador radicalmente diferentes, todos classificados sob a definição de "processamento paralelo" — um esquema que permite efetuar muitos cálculos simultaneamente. A figura abaixo ilustra uma configuração possível de um dos chamados computadores massivamente paralelos; tal máquina seria composta de centenas de milhares de pequenos processadores (*brancos*), cada um deles equipado com sua própria memória (*azuis*).

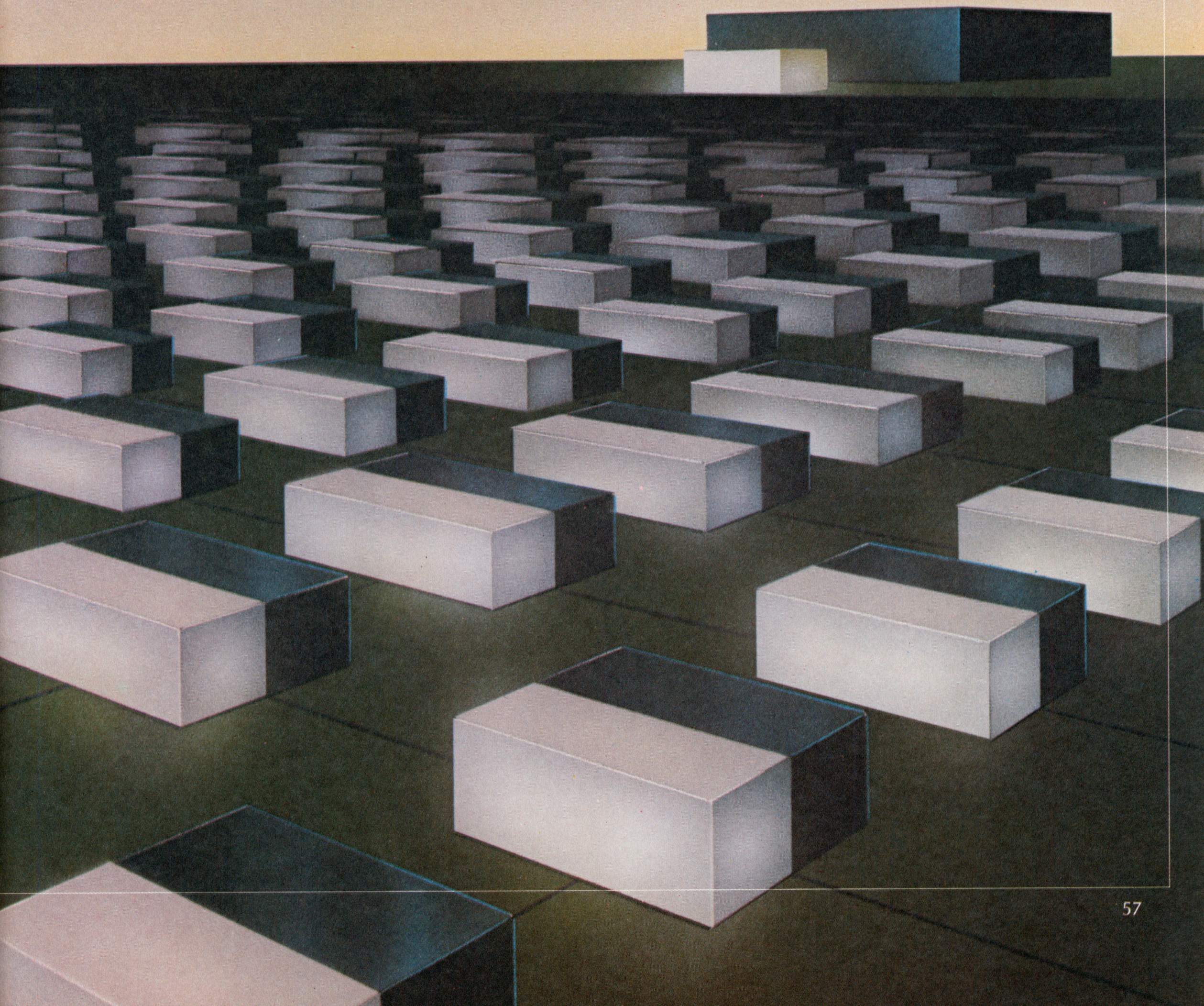
Com tal arranjo, a detecção de contorno sobre a imagem de 512 x 480 pixels descrita na página anterior pode ser efetuada numa fração do tempo gasto por uma máquina serial. Em vez de executar seqüencialmente os cálculos, tal computador poderia fazer 245 760 processadores trabalharem simultaneamente, um por pixel. Cada processador co-



municaria seu próprio valor a seus quatro vizinhos, recebendo, por sua vez, um valor de cada um deles e conservando um total dessa "rodada" em sua própria memória: o processador dividiria, então, seu total por quatro, obtendo o novo valor para seu pixel. Em vez de gastar três segundos no computador dos novos valores para 245 760 pixels, os processadores poderiam completar os cálculos necessários em cerca de meio milissegundo — 6 000 vezes mais depressa que uma máquina serial rápida.

Um aumento tão extraordinário na velocidade certamente será uma bênção para o desenvolvimento da visão de máquina, mas a construção de computadores com mais de um processador levanta muitos problemas de planejamento. Por exemplo: será que é melhor optar por processadores em número relativamente pequeno, mas grandes e poderosos, ou escolher processadores pequenos e simples, e empregar muitos milha-

res deles? É preferível que o controle global resida numa unidade de controle principal — representada aqui como a grande estrutura visível no horizonte — ou que se distribua de algum modo entre vários processadores (ou várias centenas de milhares deles)? Cada processador deve ter sua própria memória ou é melhor reservar uma memória comum para todos — e, nesse caso, o que aconteceria quando dois processadores quisessem o mesmo segmento de dados ao mesmo tempo? A ligação por fios entre tantos componentes também se torna problemática no seu aspecto físico, isso sem falar no problema de como escrever programas para máquinas paralelas. Por último, pode haver diferentes desenhos, cada um apropriado a um tipo de problema, assim como a alocação de trabalho humano varia com a tarefa particular que se tem à mão: pintar uma cerca ilustra o tipo de paralelismo que convém à visão da máquina; pintar uma obra de arte, não.



informações e os computadores seriais começa já nos níveis físicos mais fundamentais. Isso por um motivo: pensa-se que a memória, nas redes de neurônios, esteja distribuída por toda a estrutura, em vez de localizar-se numa região específica, como nas máquinas seriais. Além disso, a memória, nas redes biológicas, usualmente assume a forma de conexões entre neurônios, fortalecidas ou enfraquecidas, e não de símbolos binários armazenados. Os cientistas acreditam que, seja através de propriedades inatas, seja através da aprendizagem, as redes biológicas estão, na verdade, sintonizadas para responder a estímulos específicos. É possível, por exemplo, que cada pessoa aprenda a construir "detectores de elefante", que são ativados pela combinação correta de características tais como a grandeza e as qualidades de ser cinzento e possuir quatro patas.

Em tais memórias distribuídas, cada neurônio é, simultaneamente, seu próprio fabricante de decisões e seu próprio depósito de memória. Uma das vantagens desse arranjo é sua elasticidade: danos causados em alguns neurônios não alteram radicalmente os dados armazenados nem paralisam todo o sistema, como aconteceria num computador serial, se um fio fosse cortado ou se um único transistor fosse danificado. Uma característica mais importante dos sistemas com memória distribuída é seu potencial para aquilo que os pesquisadores da AI denominam endereçamento de conteúdo. Em vez de ser atribuída a uma célula de memória com um endereço numérico arbitrário, cada parcela de informação numa memória distribuída é preenchida e recuperada de acordo com seu conteúdo.

TRUQUES HUMANOS DE ASSOCIAÇÃO

Associações parecem formar a base da maior parte do pensamento humano: recordamo-nos de coisas a partir do conhecimento parcial, por meio de lembranças de conceitos similares e até mesmo usando deliberadamente truques associativos, como o proverbial pedaço de linha amarrada num dedo. As pessoas nem mesmo somam os números da maneira puramente mecânica que os computadores fazem; em vez disso, associamos frases memorizadas como "dois mais dois" à palavra "quatro". Tais associações provavelmente também estão presentes no cerne de percepção, reconhecimento, generalização, aprendizagem e intuição humanos.

Uma nova escola de pesquisadores de inteligência da máquina — cujos adeptos dão a si mesmos o nome de novos conexionistas, porque supõem que o pensamento envolve principalmente conexões semelhantes às dos neurônios em vez de símbolos — argumenta que, se os computadores devem se comportar mais à semelhança de cérebros, eles devem também tornar-se mais parecidos com cérebros. Em essência, eles requerem dispositivos que consistem em simulações de neurônios intrinsecamente conectados.

Os esquemas conexionista e de conteúdo endereçável apresentam certas falhas. Devido a seu tipo de organização em forma livre, eles têm o potencial de confundir diferentes segmentos de dados. Mas outros processos, também do tipo forma livre, tais como a sensibilização e a associação, parecem estar na base das idiosincrasias psicológicas do homem. Roger Schank, da Universidade de Yale, especula que essas idiosincrasias terão de ser incorporadas em qualquer computador com pretensões de inteligência. Schank, um líder do esforço pela criação de máquinas que possam compreender a linguagem natural, disse: "Os computadores atuais podem ler a mesma notícia de jornal dúzias de vezes sem entendê-la; eles somente

aprendem de novo as mesmas coisas. As máquinas não se tornarão inteligentes enquanto não forem capazes de se aborrecer com algo velho, ou de se surpreender com algo novo”.

COMPUTADORES QUE FAZEM CONEXÕES

Muitos pesquisadores esperam que as máquinas de processamento paralelo consigam atingir esse estado. Os cientistas empenharam-se em vários projetos para criar dispositivos que fossem capazes de aprender — por meio de exploração, observação, e tentativa e erro — a fazer coisas não facilmente ensinadas por qualquer programador.

Um dos mais ambiciosos dentre tais projetos foi desenvolvido por uma pequena companhia de Cambridge, Massachusetts, otimistamente batizada com o nome Thinking Machines Corporation, isto é, “Corporação de Máquinas Pensantes”. Esse processador foi concebido por Daniel Hillis, quando ele ainda estudava robótica no MIT sob a orientação de Marvin Minsky.

Em 1985, foi completado o primeiro sistema Connection Machine, isto é, “Máquina de Conexão”, com 65 000 processadores e memórias trabalhando em conjunto. Afinal, a máquina de Hillis abrigava 1 milhão de processadores, ou mais, cada um deles aproximadamente equivalente, em potência, ao processador de um microcomputador, e cada um com sua própria memória. A principal inovação de Hillis é o elaborado e flexível sistema de interconexões e chaves que permite aos processadores difundir informações e pedir ajuda a todos os outros processadores, simulando chamadas associativas semelhantes às do cérebro. O sistema também permite aos usuários rearranjar essas interconexões através de programação e, desse modo, construir realmente diferentes computadores paralelos, projetados especialmente para aplicações que variam da realização de diagnósticos médicos à operação de um veículo-robô.

Um projeto semelhante é a assim chamada máquina de Boltzmann, conceito desenvolvido em Carnegie-Mellon por Geoffrey Hinton, Scott Fahlmann, Terrence J. Sejnowsky, da Universidade John Hopkins, e outros pesquisadores (o nome da máquina é uma homenagem a Ludwig Boltzmann, um dos fundadores da termodinâmica de base estatística, desenvolvida no século XIX). Como no caso das redes de neurônios, a memória na máquina de Boltzmann consistiria, em grande parte, de padrões de interconexões, em vez de símbolos armazenados em células de memória específicas. Em operação, uma máquina de Boltzmann tentaria equiparar padrões de entrada com padrões de conexões armazenados.

Um fato que tem preocupado os modeladores neurais — cujo trabalho, em grande parte, limita-se a simulações em computadores seriais — é que as diferenças de inteligência entre os animais parecem ser, pelo menos aproximadamente, proporcionais ao número de neurônios nos cérebros de cada espécie. Construir computadores paralelos praticáveis com os bilhões de processadores necessários para chegar perto dos 100 bilhões de neurônios de um cérebro humano exigirá, portanto, uma perícia considerável.

Muitos dentre os novos conexionistas parecem reconhecer que vários dos critérios oriundos das abordagens mais tradicionais das pesquisas sobre a AI podem ajudar a responder a esse desafio. Por exemplo, o processamento de símbolos e da lógica poderia ampliar o conexionismo, assim como alargou o pensamento humano graças a invenções como a matemática. Marvin Minsky, por sua vez, prevê

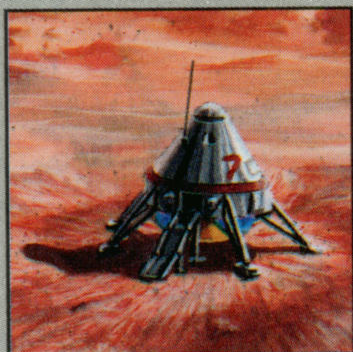
um esforço conjunto no qual muitas máquinas paralelas semelhantes ao percéptron lidarão principalmente com o mundo real, traduzindo dados sensoriais em formas simbólicas para que computadores mais tradicionais os manipulem.

O esforço para se obter inteligência da máquina constitui um dos mais extraordinários empreendimentos humanos. E a breve história dessa busca apenas aumentou o respeito que a maioria dos seus participantes tem pelo poder e pelos recursos do cérebro humano. Considerando os conceitos e abordagens atuais, a realização do sonho de uma máquina para todos os fins, com inteligência em grau humano, exigirá tremendos avanços em hardware e software — muitos milhões de processadores e muitos milhões de regras e fatos. Tudo isso levanta dúvidas sobre até que ponto tais perspectivas são realistas. Alguns pesquisadores, como Hans Moravec, perito em robótica de Carnegie-Mellon, acreditam que sim, que as perspectivas são, de fato, realistas. Na visão de Moravec, o segredo reside principalmente na construção de computadores mais rápidos, seja por intermédio de paralelismo, seja por outros meios. Na verdade, ele acredita que o objetivo da inteligência a nível humano possa ser atingido já no início do século XXI. Mas tal otimismo não é compartilhado por muitos.

De qualquer modo, o obstinado empenho com que a AI persegue seus ambiciosos objetivos continua a influenciar todos os aspectos da tecnologia do computador; já não é tão fácil distinguir a AI das outras pesquisas sobre computadores. Mas nem todas as pessoas que trabalham com a AI concordam que haja motivo de alegria nessa aceitação obtida junto à corrente principal da atividade científica. De fato, corre nesse campo um ditado que, em parte, traduz aceitação, em parte revela queixa: "Se for útil, não é AI". Tradicionalmente, a AI tem atraído pessoas de determinado tipo, mais inclinadas a perseguir metas que se perdem de vista — cientistas que dedicam muito pouca atenção às aplicações práticas de seu trabalho, ou seja, tão logo alguma surpreendente faceta da inteligência humana seja reduzida a uma prática cotidiana de engenharia, ela perde seu mistério e seu atrativo para esses devotados pesquisadores da AI. Como observou Marvin Minsky: "Quando você explica alguma coisa, você explica até o fim".

Talvez sim, mas a inteligência humana está longe de ter sido despojada de seu mistério. Talvez passem séculos antes que a ciência invente máquinas com a gama de compreensões, a versatilidade e as habilidades gerais de uma pessoa de instrução média, quanto mais de um gênio. A curto prazo, surgirão muitos produtos poderosos e úteis da AI para manipular estreitos campos de especialização e tarefas sensoriais e motoras para fins específicos: particularmente úteis às pessoas comuns serão os sistemas que podem comunicar conhecimentos em linguagem do dia-a-dia. Mas num futuro previsível, os computadores provavelmente permanecerão o que sempre têm sido — ajudantes para tarefas especializadas. Durante mais um século, pelo menos, os generalistas — os sonhadores e os planejadores, os que propõem problemas e os que os resolvem — ainda serão pessoas, confiando em seus inigualáveis cérebros biológicos.

A fronteira espacial

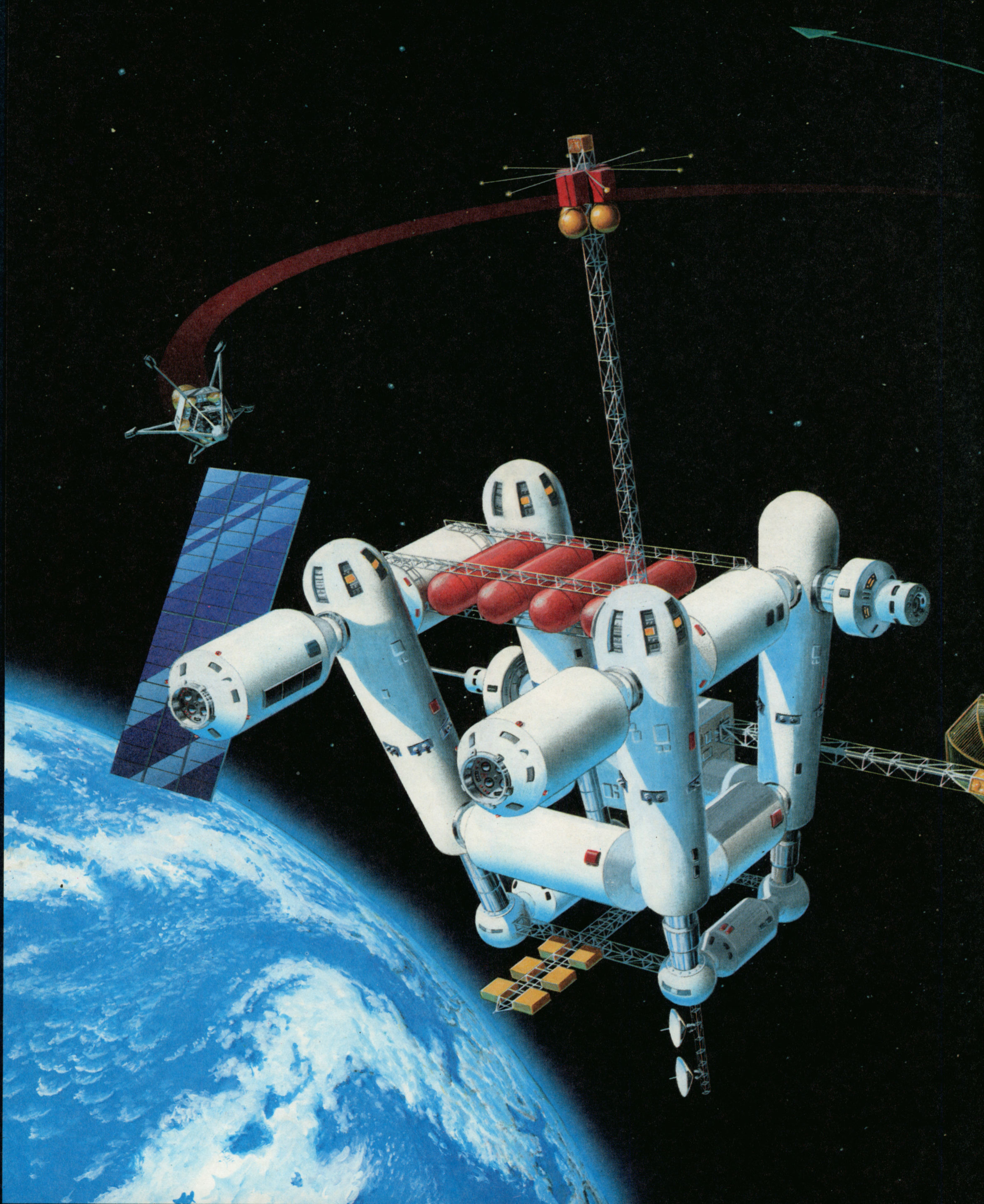


Graças a esforços persistentes e, com frequência, brilhantes, ao longo de mais de três décadas, pesquisadores em várias disciplinas científicas têm conseguido ajustar algumas peças desse enorme quebra-cabeças que é a inteligência artificial. Algumas dessas peças — notavelmente os sistemas expert, a visão computadorizada, a automação, a robótica e programas para a compreensão da linguagem natural — tiveram aplicações práticas imediatas em campos como o dos negócios, medicina e indústrias automobilísticas e petrolíferas.

Mas dentre os que certamente irão beneficiar-se com os progressos futuros da AI, talvez a exploração espacial ocupe o primeiro lugar. Recentemente, a NASA, o Departamento de Defesa dos EUA e várias empresas aeroespaciais comerciais aceleraram as investigações nas áreas da AI que mais promessa abrigam de realizar uma meta ambiciosa: a criação de sistemas computadorizados capazes de substituir o homem no espaço.

As máquinas que resultarão dessas pesquisas deverão ser capazes de atuar com um certo grau de autonomia. No mínimo, tais sistemas necessitarão de bancos de conhecimentos, sejam eles de natureza geral ou específica, e de um meio de manipular esse conhecimento que permita chegar a inferências e julgamentos. Algumas máquinas irão requerer a capacidade de navegar pelo espaço, outras a de cumprir sua missão nas superfícies de planetas ou de luas distantes. E, para serem tão versáteis e úteis quanto possível, esses exploradores computadorizados deverão possuir, naturalmente, não apenas faculdades sensoriais, como visão, audição e tato, mas também a capacidade de aprender a partir da experiência.

Aspectos de cada uma dessas capacidades já foram antecipados (com graus de sucesso bastante variados) em várias máquinas diferentes sediadas no solo, e um sistema expert denominado LES (sigla para Liquid Oxygen Expert System, isto é, Sistema Expert de Oxigênio Líquido) foi testado para uso no ônibus espacial. A partir do que está sendo feito e do que está em estudo, as ilustrações nas páginas seguintes fornecem um panorama de algumas das maneiras pelas quais computadores equipados com inteligência artificial poderão ser postos a funcionar no espaço, em alguma época no século XXI. Esse tipo de extrapolação ainda é, na melhor das hipóteses, arriscada; nas palavras de um observador: "O paradoxo de predizer as ciências está no fato de que conhecer o que será conhecido no futuro é possuir o conhecimento hoje".



Um depósito automatizado

O primeiro posto avançado da Terra, com base permanente no espaço, será completado, segundo se prevê, em meados da década de 1990. Ficará estacionado a 400 quilômetros da superfície terrestre, em LEO (sigla para Lower Earth Orbit, isto é, Baixa Órbita Terrestre). A instalação, que atenderá a vários fins de pesquisa e de produção industrial, será construída prevendo adaptações no software e no hardware, a fim de permitir que novas tecnologias de AI sejam integradas ao sistema.

Esse processo evolucionário foi comparado à criação gradual de um colega eletrônico para a tripulação humana da estação. Diferentemente dos personagens robôs nos filmes, o componente de AI da estação espacial provavelmente não possuirá características físicas humanas. Inicialmente, pelo menos, é mais provável que ele consista numa vasta coleção de sistemas expert, principalmente devotados a monitorar e a regular o ambiente interno da estação e as condições físicas externas. A tecnologia para isso derivará em parte de alguns sistemas experimentais com base no solo, que estão sendo desenvolvidos para o programa espacial. Um desses sistemas é o KATE (sigla para Knowledge-Based Automatic Test Equipment, isto é, Equipamento de Teste Automático Baseado em Conhecimento), um sofisticado localizador e consertador de defeitos, que pode controlar e monitorar o hardware; outro é o IMIS (sigla para Intelligent Management Information System, isto é, Sistema de Informações de Manejo Inteligente), que resgata informações de um banco de dados e responde a perguntas numa linguagem fácil de entender. A estação espacial possuirá um sistema de sensores, que incluirá câmeras de vídeo equipadas com software para reconhecimento de padrões, que vigiará os sistemas de energia, de controle ambiental e de suporte de vida da estação.

À medida que seus sistemas evoluam, a estação espacial atingirá uma espécie de adolescência. Nessa fase intermediária, o componente de AI da estação terá conquistado um moderado grau de autonomia, baseada, em parte, no desenvolvimento de programas de linguagem natural — que permitirão ao sistema responder a perguntas faladas — e, em parte, em sensores visuais e táteis, que melhorarão suas habilidades para monitorar.

Espécie de garagem celeste de ônibus, a estação espacial servirá eventualmente de base de origem para uma frota de veículos autônomos. Sistemas de percepção visual da máquina e de navegação darão a essas estações extraterrestres habilidades como as de abrirem caminho por si mesmas para a Lua, e para fora dela, e de executarem tarefas simples como localizar e recuperar satélites para reparos.

Base de origem para uma força robô

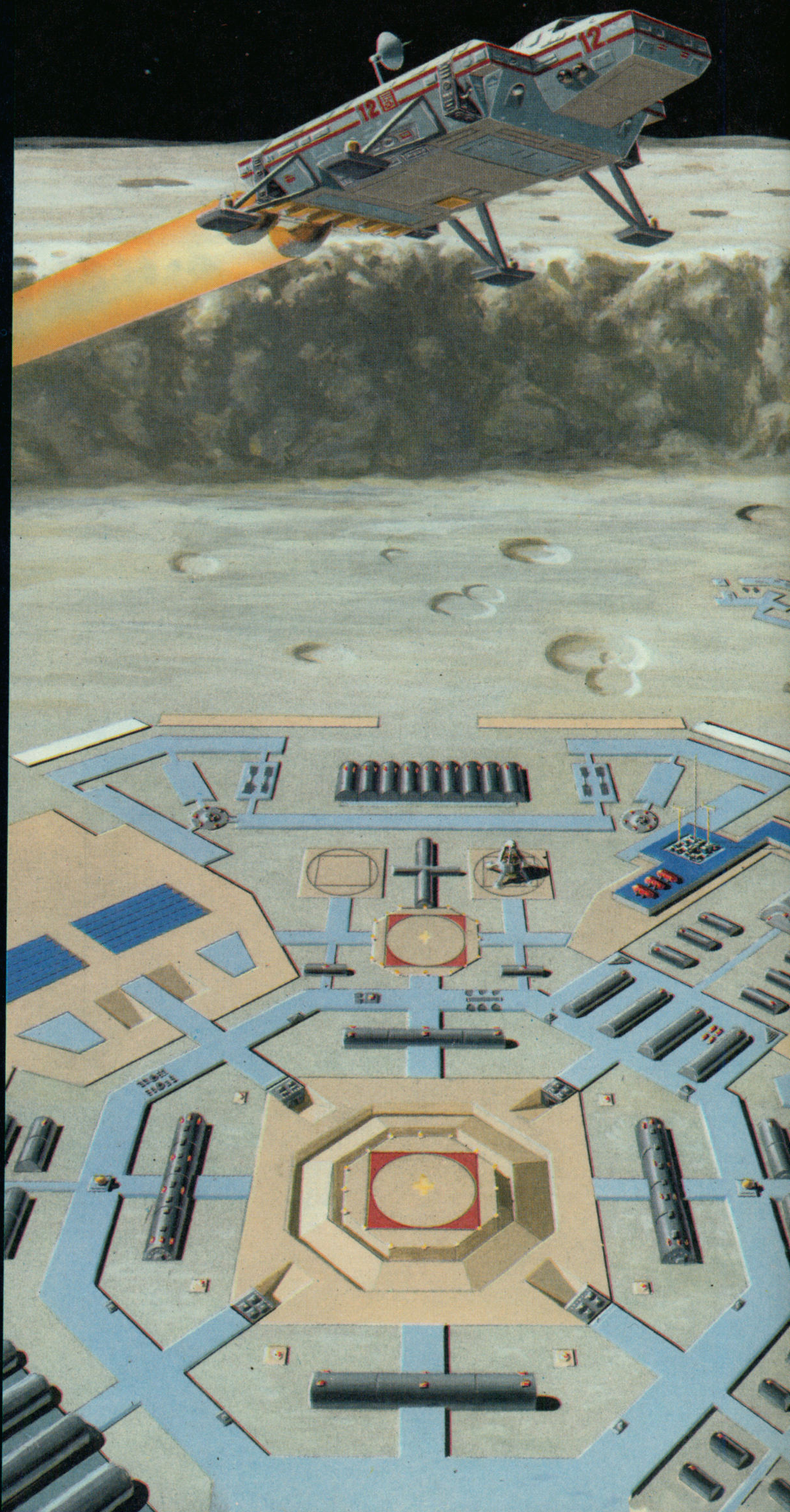
O conjunto geométrico de estruturas à direita é a projeção de uma instalação autônoma do século XXI para pesquisas e processamento de recursos minerais na Lua. De acordo com um esquema, uma comunidade seminal de robôs, equipados com um kit inicial de equipamentos para fabricação, poderá construir essa instalação com materiais lunares e operá-la com um mínimo de supervisão humana.

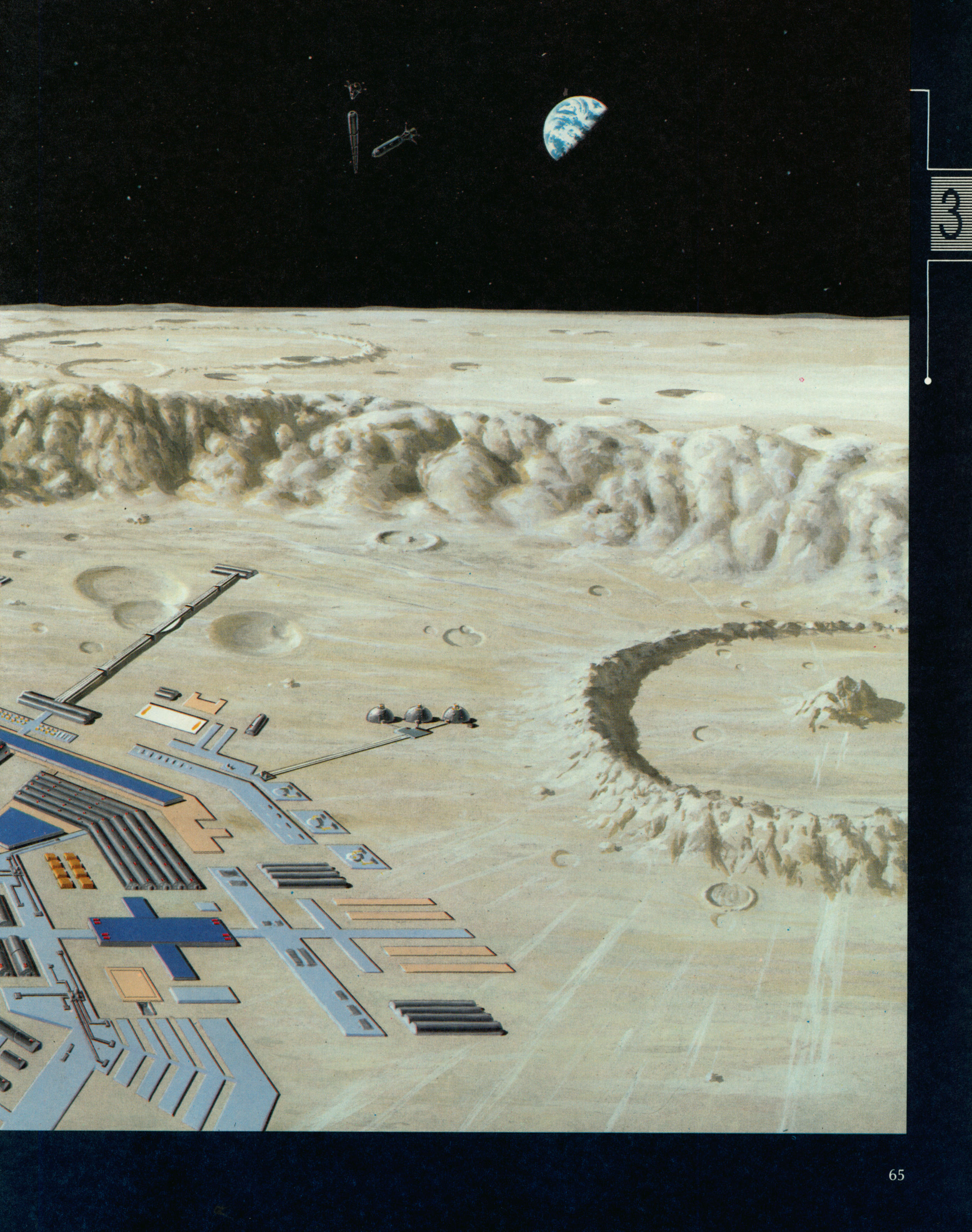
Esse plano ambicioso origina-se, em parte, de uma confusão de propostas sobre os assim chamados sistemas auto-replicantes — autômatos que serão capazes de produzir sua própria prole que, por sua vez, poderá também duplicar-se. Na primeira fase de desenvolvimento lunar, esse método será usado para criar uma grande tropa de robôs exploradores, mineiros e operários.

Mais aptos a manter-se sobre a superfície inóspita da Lua do que os seres humanos, esses nativos mecânicos construirão uma fábrica alimentada com energia solar; elementos como aço, titânio, alumínio, silício e oxigênio serão extraídos do solo lunar. Com eles poderão ser fabricados vários produtos, como componentes de espaçonaves, usados na Lua ou exportados para a estação espacial ou para conjuntos de instalações industriais funcionando em gravidade zero e estacionadas em LEO.

A noção de robôs capazes de construir e operar fábricas — e também de se reproduzirem — sugere a façanha de realizar o tipo de inteligência de máquina de amplo calibre que, desde há muito, tem sido a meta das pesquisas sobre a AI. De fato, os robôs imaginados para a exploração espacial no início do século XXI serão, provavelmente, versões aperfeiçoadas dos atuais sistemas expert baseados em regras e restritos a um domínio específico. É provável que esses sistemas sejam equipados com processadores ultra-rápidos que, conforme se espera, resultarão de pesquisas sobre a arquitetura dos computadores de processamento paralelo. Com desenvolvimentos adicionais na potência, esses sistemas terão uma capacidade muito superior à de seus predecessores no que se refere ao processamento de imensas quantidades de dados colhidos pelos sensores táteis e visuais, e também na manipulação das complexas regras e bancos de conhecimentos que permitem aos sistemas operarem sem a ajuda de seres humanos presentes.

Um versátil veículo de transporte sobrevoa uma instalação lunar de pesquisas e de produção industrial, construída a partir de matérias-primas extraídas de minas abertas nas vizinhanças. Produtos manufaturados, como componentes de espaçonaves feitos de alumínio, poderiam ser enviados, por exemplo, à fábrica em gravidade zero, quase invisível no céu.





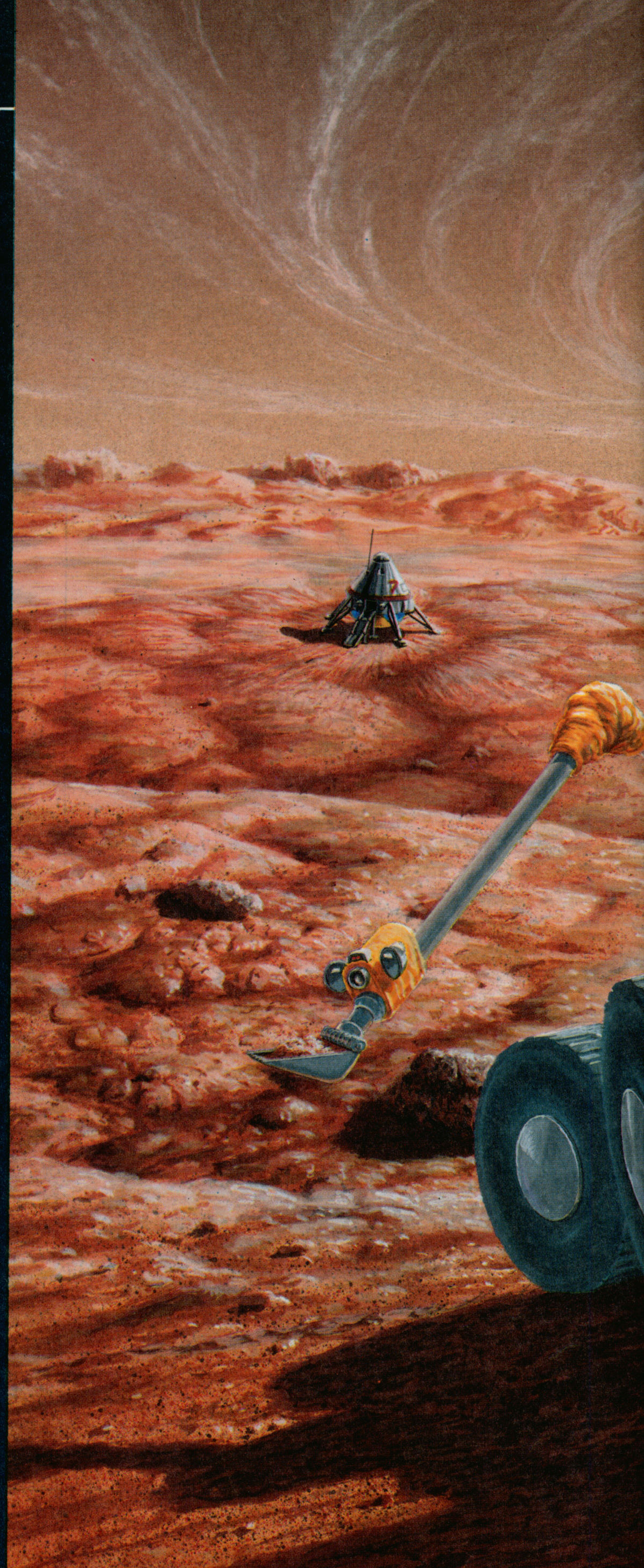
A equipe avançada em mundos distantes

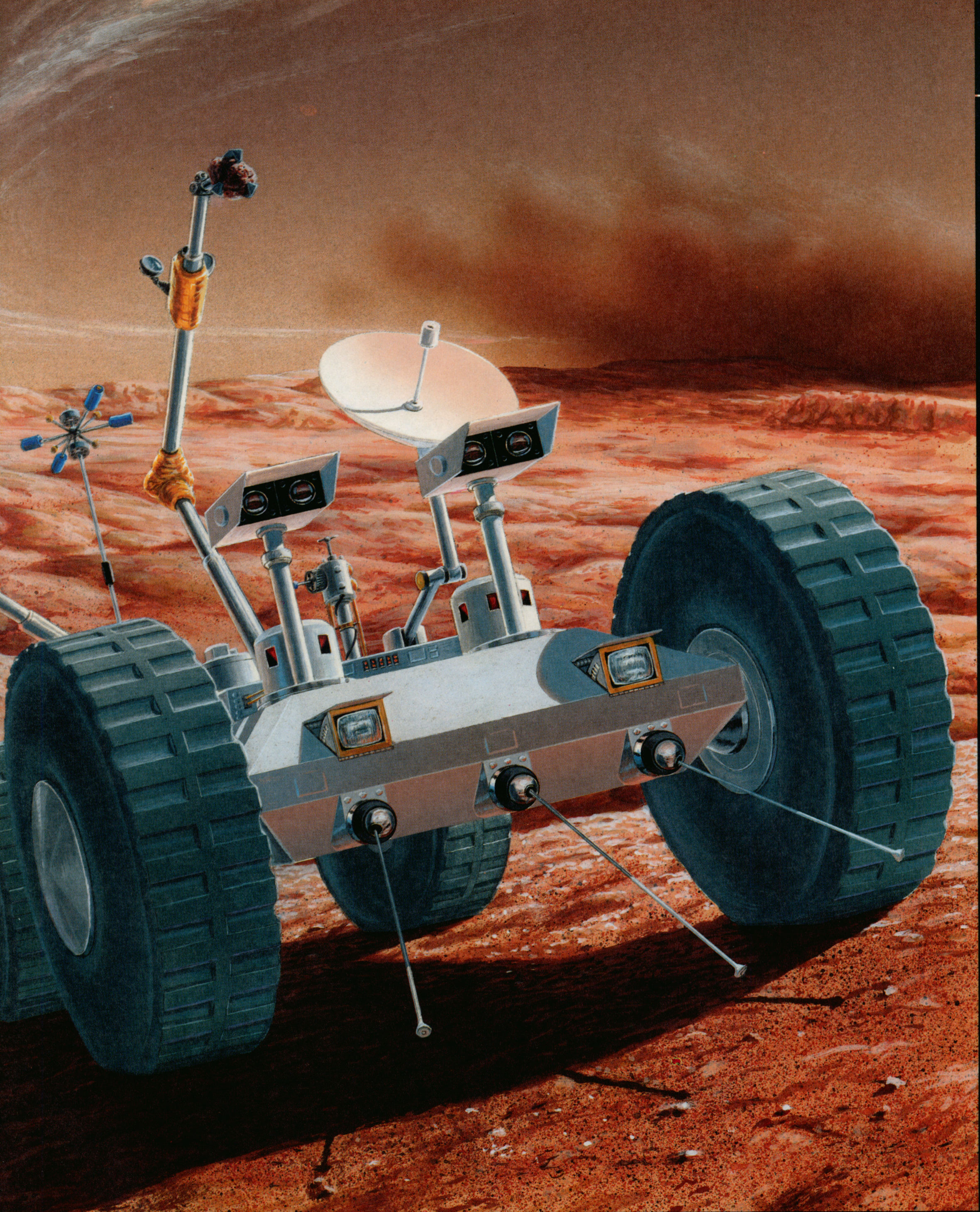
A autonomia das máquinas no espaço provavelmente variará desde o limitado grau de liberdade dos dispositivos a bordo da estação espacial à maior independência do exército de veículos orbitais e planetários capazes de operar por longos períodos sem intervenção humana direta. Os aperfeiçoamentos contínuos em algumas das principais áreas de pesquisa sobre a AI e a robótica — por exemplo, a visão e outros sistemas sensoriais, e sistemas que permitem aos robôs manipular vários objetos — deverão dar a essas máquinas precisão e destreza quase humanas.

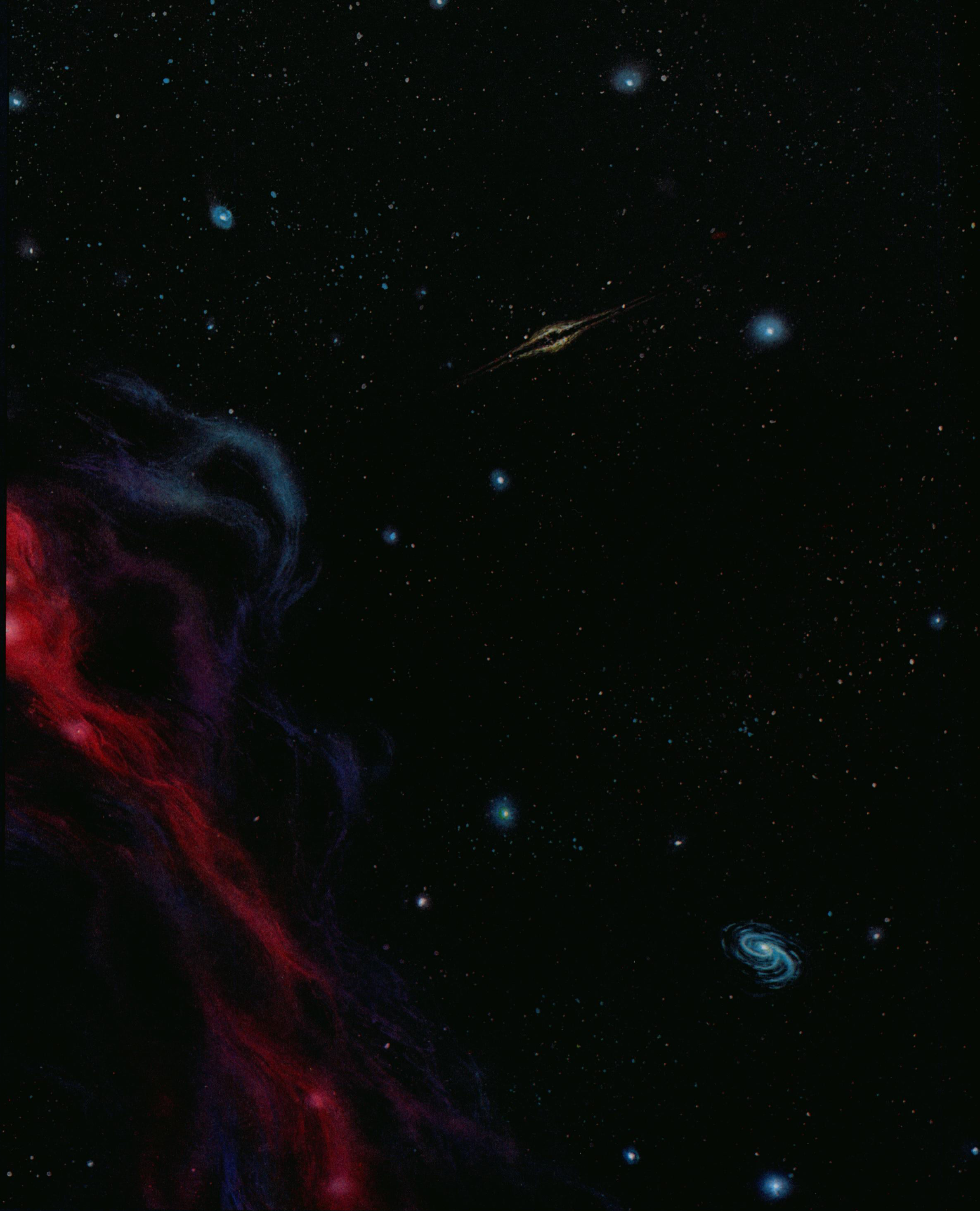
Veículos de operação com base no solo, semelhantes ao explorador ilustrado à direita, representam um estágio avançado do desenvolvimento robótico. Planejados para a investigação sistemática das superfícies da Lua e de Marte, esses veículos incorporarão avanços tecnológicos testados pela primeira vez em máquinas que serão usadas a bordo da estação espacial. Aí, os dispositivos funcionarão sob a estreita supervisão de seres humanos, que os manipularão por controle remoto, virtualmente junta a junta. Mais tarde, com a adição de sistemas para reconhecimento de padrões e de bancos de conhecimentos mais extensos, os robôs terão a capacidade de executar tarefas de manutenção simples, dentro e fora da estação. Desenvolvimentos dos sensores táteis e da capacidade de controle de braços manipuladores (que já começaram a ser usados na Terra para ajudar, por exemplo, os cirurgiões na sala de operação) darão aos robôs a capacidade de executar experimentos científicos delicados e tarefas simples de manufatura de precisão.

Todos esses aperfeiçoamentos — e outros mais — serão necessários para desenvolver espaçonaves autônomas e máquinas com base no solo, em missões não supervisionadas. Veículos terrestres não tripulados terão de ser equipados com núcleos de informações e regras simples e práticas para a exploração planetária. Os veículos-robôs também precisarão possuir sistemas de visão altamente desenvolvidos, que utilizarão câmeras de televisão, radar a laser e software para o reconhecimento de padrões. Esses sistemas de visão também irão requerer processadores de alta velocidade para permitir às máquinas lidarem com dados visuais com a mesma rapidez com que os recebem.

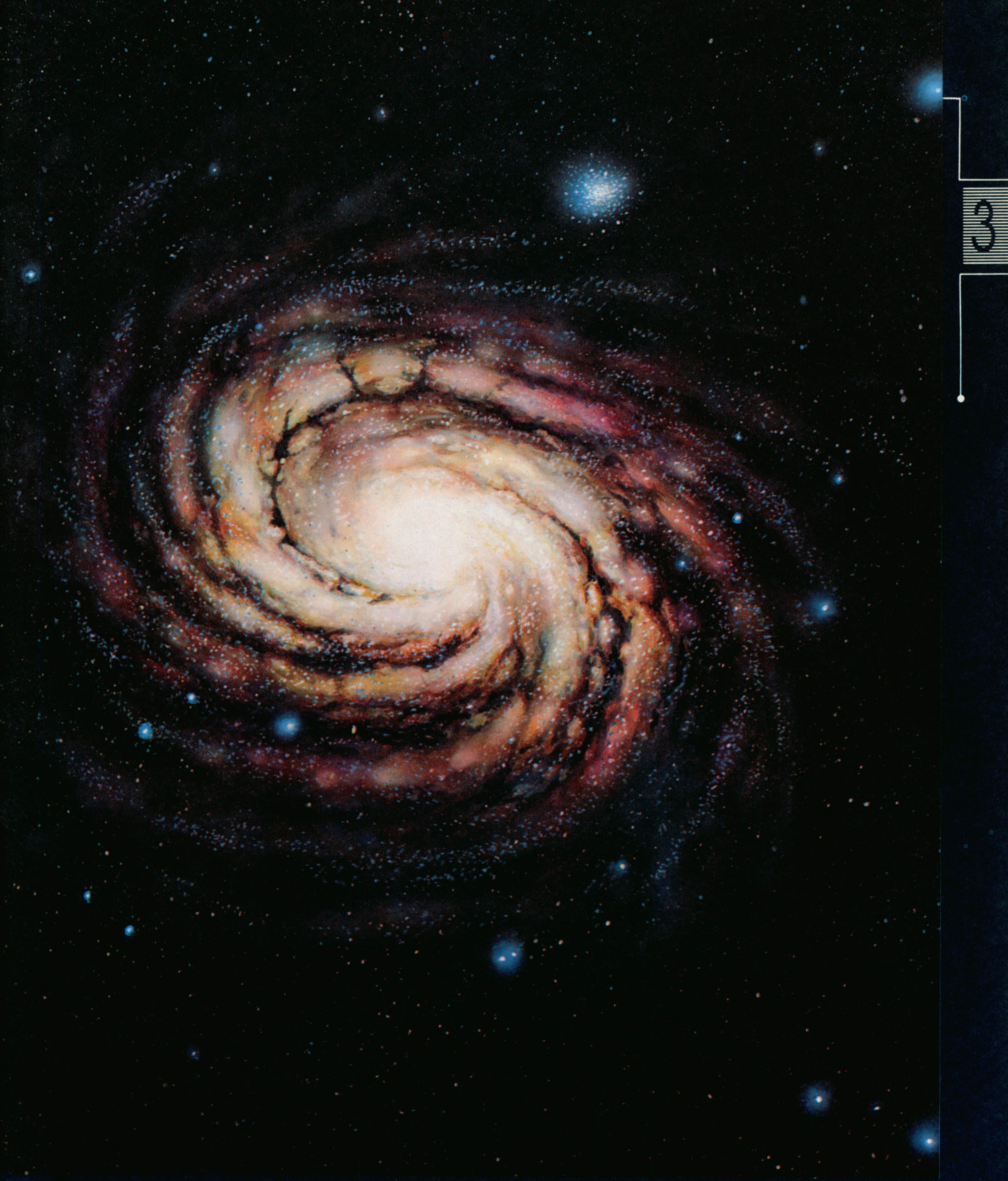
O veículo de operação planetária, de seis rodas, modelado com base num protótipo de veículo terrestre autônomo desenvolvido para missões de reconhecimento militares, explora a superfície de Marte à procura de recursos minerais, água e, possivelmente, locais para se estabelecer. Sensores táteis sondam o solo à frente e braços flexíveis coletam amostras, enquanto duas câmeras retransmitem informações visuais ao sistema de visão do veículo.







No século XXI, robôs exploradores poderão vagar pelo universo, transmitindo informações sobre a vida em outras galáxias.



Os números em *itálico* indicam uma ilustração do assunto mencionado.

A

Albeson, Robert 18-19
Am (Matemático Automatizado) 30-31
Anderson, James 53
Aprendizagem 28-29
Aprendizagem de máquina 23-34, 47; por descoberta 25-27, 30-34, 47; por exemplo 25-26, 35, 36-47
Aprendizagem por descoberta 26-27, 30-34, 47
Aprendizagem por exemplo 25-26; 35, 36-47; conhecimento de domínio 35-37; especialização excessiva 44; generalização excessiva 44; tarefa da aprendizagem 35-37; tarefa de desempenho 35-37
AQ 11, 26
Arches 25-26

B

Backus, John 53
BACON 27, 30
Boltzmann, máquina de 59

C

Carnegie (Universidade de Carnegie-Mellon), Instituto de Tecnologia 27
Cibernética 53-58
Chomsky, Noam 9-10
Colby, Kenneth 9
Computador, arquitetura do: paralela 53, 56-57, 58-60; e percepção 52-53, 58-59; serial 52-53, 54-55
Conceitos indistintos 53
Conexionistas, novos 58-59
Conhecimento, representação do 12-13, 14-15, 16-17, 18-21; MOPs 20; quadros 15, 16-17, 31; rede semântica 12-13, 14-15; scripts 19
Connection Machine 59
Cyc 34
Cyrus 20-21

D

Damas 23-25
De baixo-para-cima, escala 53, 58
Dependência conceitual 19
Detecção de contorno 55-57
DOCTOR 8-9
Domínio de conhecimento 35-37

E

ELIZA 8-9, 21
Endereçamento de conteúdo 58
Espaço, exploração do 61; cenário do 62-69
Especialização excessiva 44
Eurisko 31-34

F

Fahlmann, Scott 14; 52-53, 59
Fala, reconhecimento da 50-52. *Veja também* SUR
Fonêmica 51
Fonética 51

G

Generalização excessiva 44

H

Hearsay II 50-52
Heurística 24-26, 35
Hillis, Daniel 59
Hinton, Geoffrey 59

I

IBM: 701 23-24
Illinois, Universidade de 23
IMIS (Sistema de Informações de Manejo Inteligente) 63
Inteligência artificial: aprendizagem da máquina 23-34; fundadores 66; modelagem neural 53, 58-60; no espaço 61, nome 6-7; pensamento 59-60; percepção 52-53, 58-59

L

Langley, Patrick 27
Lenat, Rouglas 30-34
LES (Liquid Oxygen Expert System) 61
Linguagem natural 5-6, 11
Linguagem para AI, programação de 6
Linguagem pelo computador, compreensão da 6-11, 14-15, 18-21, abordagem sintática 6, 9-11; e semântica 10-11, 14-15, 18-21; regras se-então 21; representação do conhecimento 14-15, 18-21
Linguagem por computador, tradução de 5-6
LISP 7-8, 31
Lua, pesquisas e desenvolvimento sobre a 64-65

M

Matemática, programa de aprendizagem de 30-31, 35, 36-47
McCarthy, John 6-7
Memória: endereçamento de conteúdo 58; e representação de conhecimento 12-13, 16-17; distribuída 58
Michalski, Ryszard 26
Minsky, Marvin 6, 15, 17-18, 49, 59-60
MIT (Instituto de Tecnologia de Massachusetts) 7, 15
MOPs (Pacotes de Organização da Memória) 19
Morfêmica 51

N

Nealey, Robert W. 24
Neurônios, modelagem eletrônica de 53, 58-60
Newell, Allen 6, 50
Nilsson, Nils 20
Nodos 12-13; vínculos entre os 14

O

Oettinger, Anthony G. 5

P

Parallel Models of Associative Memory 53
Parser 10
Partilha de tempo 7-8

Pensamento 58-59

Percepção da máquina 49-50; e arquitetura de computador 52-53, 54-57, 58-60; memória 58; reconhecimento da fala 50-52; visão 52, 54, 55-57

Percéptrons 59

Pragmática 51

Processamento em paralelo 53, 56-57, 58-60

Prosódia 51

Q

Quadros 15, 16-17, 31
Quillian, M. Ross 14-15

R

Reddy, Raj 50, 52
RLL (Linguagem da Linguagem de Representação) 31
Robôs no espaço 64-69
Rogers, Carl 8

S

SAM (Mecanismo Aplicador de Scripts) 19-20
Samuel, Arthur 23-25
Schank, Roger 15, 18-21, 25, 58
Scripts 19-20
Searle, John R. 21
Sejnowsky, Terrence J. 59
Semântica 10-11, 14-15, 18-21, 51; rede 12-13
Shaw, J. C. 50
SHRDLU 10-11
Simon, Herbert A. 6, 27, 30, 50
Sintaxe 51; maneira de abordar a linguagem 6, 9-11
Sistema expert: limitações 25; no espaço 61
SRI International 21
Stanford, Universidade de 7
SUR (Pesquisas para Compreensão da Fala) 50

T

Thinking Machine Corporation 59
Traveller TCS 32-33

V

Vínculos entre os nodos 12-13, 14
Visão da máquina 52, 54; detecção de contorno 55-57
Von Neumann, John 52, 54-55

W

Weaner, Warren 5
Weizenbaum, Joseph 8-9, 21
Winograd, Terry 10-11
Winston, Patrick H. 25-27, 34

Y

Yale, Universidade 19, 20, 25

EDITOR
RICHARD CIVITA



NOVA CULTURAL

Presidente: Flávio Barros Pinto
Diretoria: Anselmo Pecci, Pierluigi Bracco,
Plácido Nicoletto, Walter Thomé

REDAÇÃO

Diretor: Carmo Chagas
Editora Executiva: Stefania Crema

COMERCIAL

Diretor: Plácido Nicoletto
Gerente Comercial: Joaquim Celestino da Silva
Gerente de Circulação: Denise Mozol
Gerente de Propaganda e Promoções: José Carlos Mádio
Gerente de Pesquisa e Análise de Mercado: Wagner M.
P. Nabuco de Araújo

PRODUÇÃO EDITORIAL

Pedro Paulo Poppovic Consultores Editoriais S/C Ltda.
Diretor: Pedro Paulo Poppovic
Editora-chefe: Elzira Arantes
Editora: Elisabeth Justino Carastan
Arte: Massumi Guibu (chefe), Sônia Regina Aversa (diagramadora)
Projeto gráfico (capa): OZ Desenho e Associados
Coordenação de produção: Sônia Maria Pimentel
Colaboradores: Mathias Mendes Wolff (consultoria),
Newton R. Eichenberg (tradução), Luis Reyes Gil (copidescagem)

A Editora Nova Cultural Ltda. é uma empresa do Grupo
CLC — Comunicações, Lazer e Cultura.

Presidente: Richard Civita

Diretoria: Flávio Barros Pinto, João Gomez, Menahem
M. Politi, René C. X. Santos, Stélio Alves Campos

TIME-LIFE BOOKS INC.

DIRETOR DA SÉRIE: Roberta Conlan
Coordenador da série: Loretta Britten
Programador visual: Robert K. Herndon
Editores associados: Russell B. Adams Jr. (texto),
Sara Schneidman (ilustrações)
Pesquisadores: Elisabeth Carpenter, Karen Monks, Pamela L. Whitney
Redação: Lydia Preston
Assistentes de programação visual: Antonio Alcalá,
Christopher M. Register
Coordenação de revisão: Anthony K. Pordes
Coordenador iconográfico: Renée DeSandies
Assistente editorial: Miriam Newton Morrison
Colaboradores especiais: Tom Alexander, Ronald H. Bailey,
Donal Kevin Gordon, Charles Smith, Daniel Stashower,
David Thiemann (texto); Sara Mark (pesquisa)

CONSULTORES

MARVIN MINSKY, RONALD BRACHMAN,
JAIME G. CARBONELL, RANDALL DAVIS,
MIKE DRUMHELLER, CARL EBELING,
RICHARD M. KELLER, DOUGLAS B. LENAT,
CURTIS MARX, ISABEL LIDA NIRENBERG,
STEVE SHAFER, HARRY L. VOORHEES.

Créditos das fotos: Os créditos das fotos e ilustrações estão separados da esquerda para a direita por ponto-e-vírgula, e de cima para baixo por travessão.

Capa: ilustração Steve Wagner. 4-11: ilustrações Steve Wagner. 12, 13: ilustração Aaron Bowles. 14, 15: ilustração Steve Wagner. 16, 17: ilustração Aaron Bowles. 18-22: ilustrações Steve Wagner. 28, 29: ilustração Sean Daly. 35-51: ilustrações Steve Wagner. 54, 55: ilustração William J. Hennessy Jr. 61-69: ilustrações Peter Sawyer.

