

Que inteligência artificial?

EUGÊNIO OLIVEIRA *

«Be caution however! it is as easy to become a rabid believer as it is to remain dogmatically pessimistic.»

Patrick Winston – *Artificial Intelligence*

1. INTRODUÇÃO

O fulgurante êxito da revolução científica e técnica é sem dúvida o dado mais marcante desta segunda metade do século XX.

A sua ligação ao processo produtivo por um lado, e a sua directa penetração nos modos de viver, agir e pensar próprios do homem das sociedades mais ou menos desenvolvidas, alargou horizontes, suscitou esperanças e medos, criou problemas qualitativamente novos.

Como sempre, no epicentro de todo o processo, o homem defronta-se com o homem. Mas desta vez por interposta ... entidade. Não há mística mas, ainda assim, mitificável.

O computador é tido já há muito como instrumento insubstituível nas sociedades industrializadas. A população activa que nos EUA depende directamente no seu trabalho do processamento automático de informação é já cerca de 50%. A importância do respectivo mercado pode mesmo vir a ultrapassar em uma ou duas décadas a importância do próprio mercado da energia.

Nada de mais previsível, normal e aceitável para a mentalidade do homem de hoje educado na (pela e para a) mudança. O que é verdadeiramente novo causando hesitações, reticências ou perplexidade é a queda do velho dogma de que o computador

não seria mais do que uma calculadora rapidíssima e de enorme memória.

E é a partir do momento em que o computador adquire a possibilidade de processar dados sob a sua forma simbólica, e já não exclusivamente numérica, que um novo conceito desliza, não foi o primeiro nem será certamente o último, do universo da ficção para o da ciência: a Inteligência Artificial (IA).

De imediato atacada ou adulada, ridicularizada ou venerada, a disciplina da Inteligência Artificial vai-se demarcando, autonomizando, desenvolvendo, perspectivando e realizando no tempo.

INTELIGÊNCIA?

Mas o que é a inteligência? Inteligência humana? Animal?

Serão os elementos criativos formalizáveis, modelizáveis através da máquina?

Que similitude ou pontos de contacto poderão existir entre uma «inteligência» artificial e uma inteligência humana complexa e diferenciada nos seus vários níveis: informativo-lógico, gnoseológico, psicológico, neuropsicológico, neurofisiológico, bioquímico?

Não será o espírito humano em última instância incompreensível a ele próprio por ser qualquer interpretação dele apenas um seu subconjunto (uma variação do paradoxo de Zenão)?

Poderemos reduzir a inteligência à capacidade (ou à sua medida) de tomar decisões correctas em situações complexas com despesas de recursos aceitáveis? Não haverá nesta definição a introdução abusiva de conceitos de engenharia?

* Núcleo de Inteligência Artificial do Departamento de Informática da UNL.

ARTIFICIAL?

Será o *software* do computador, produto puramente intelectual do homem, verdadeiramente «artificial»?

Será a interface homem-máquina nada mais afinal que uma interface homem-homem?

Quantas questões tantas vezes suscitadas à volta ou *por fora* da IA e não tanto (ainda ou já) por dentro da própria disciplina.

Com efeito, nebulosamente primeiro, meteoricamente nos últimos oito a dez anos, os trabalhadores científicos dos centros de IA têm vindo pertinazmente a conseguir impor à comunidade científica internacional um trabalho realizado com os materiais que da matemática, da lógica e da informática sobretudo, souberam incorporar no seio da sua própria disciplina.

2. EXPLICAÇÃO

DEFINIÇÕES COMPREENSIVA E EXTENSIVA

Não cabe aqui uma definição de inteligência¹.

Digamos apenas, para o que nos interessa, que a Inteligência está intimamente relacionada com as capacidades de processamento e representação da informação.

Será talvez pacífico definir simplesmente IA como o estudo de processos que possibilitam o computador produzir as mesmas coisas que fazem as pessoas ser (ou parecer) inteligentes.

Se compreender melhor a «inteligência» dos computadores é ou não uma maneira de estudar a inteligência em geral é ponto polémico. Provavelmente a cobiça perfeita não chegará a existir. Mas já não haverá dúvidas que os objectivos centrais da IA são tornar os computadores mais úteis e, em paralelo, compreender os princípios que tornam a inteligência possível (o que não será a mesma coisa que compreender a inteligência *elle même*).

Mas se a IA torna os computadores mais úteis,

¹ «In fact, we might say that our present knowledge of the mechanisms of intelligence consists of small islands in a large ocean of speculation, hope and ignorance.» (Nilsson, 1980).

então engenheiros e informáticos (entre outros) quererão certamente saber como a IA pode ajudar a resolver os problemas concretos com que se debatem.

Ou ainda de uma forma mais evidente, que contentará até os detractores da IA: se os computadores são e serão estúpidos, não poderemos tentar fazer (e explicar como se faz) um uso inteligente deles? A IA propõe-se também isso.

Ilustrando um espectro unidimensional da relação utilizador-computador o que deve ser dado ao computador com vista à realização de uma tarefa, tal espectro dito *do-Quê-para-o-Como* apresenta um nítido deslocamento de um dos extremos (o Como) para o outro (o Quê). Isto é, cada vez menos se tem de explicar ao computador *Como* realizar uma certa tarefa e cada vez mais nos aproximamos da situação de definir o *Quê*, ou seja quais os resultados que queremos obter.

A IA é um ramo da ciência («our young science is still more art than science») (Freigenbaum, 1979) que relaciona um fenómeno complexo, a capacidade cognitiva humana, numa analogia com os programas de computadores. De certo modo este paradigma da IA é similar ao da Biónica, disciplina que pede ideias emprestadas às estruturas físicas dos animais e plantas para projectos humanos como pontes, edifícios, aviões, helicópteros, etc. Mas aqui na IA ao contrário da Biónica, não tratamos com um fenómeno desconhecido e outro bem conhecido mas, isso sim, com dois fenómenos apenas parcialmente conhecidos e explorados.

Portanto, se a máquina é um computador digital e está envolvida em tarefas que manipulem raciocínio simbólico em vez de apenas cálculo numérico e armazenamento-retribuição de informação, podemos dizer que estamos em IA.

Shortliffe (1976) explicita 4 metodologias e 8 campos de aplicação para a IA. Resumamos com brevidade.

METODOLOGIAS

A) A modelação e representação do conhecimento. A representação escolhida é de importância crucial para a eficiência de um programa de IA. Pode usar-se o cálculo de predicados para representar factos e regras, os *semantic networks*, sistemas de produção mais ou menos próximos das gramáticas ou ainda as representações procedimentais.

- B) Técnicas de inferência e dedução baseadas no desenvolvimento dos programas de *Problem solving* em computadores.
- C) Pesquisa heurística. Ou seja, estratégias usadas pelos programas para limitar o número de acções alternativas a investigar, evitando que a potencialidade do computador seja desperdiçada ineficientemente. Esta é também uma área particularmente importante para a programação em IA.
- D) Sistemas e Linguagens de muito alto nível a usar pelos investigadores em IA: tornando-se mais evidente que as linguagens tradicionais de cálculo numérico são inapropriadas, outras foram desenvolvidas que enfatizam o processamento de listas (como o Lisp) ou que incorporam facilidades que originalmente pertenciam à iniciativa do programador (como o Prolog) retrocesso sistemático (*backtracking*), certas estratégias de pesquisa, *pattern matching*, etc.²

CAMPOS DE APLICAÇÃO

1. Jogos: É uma tradição o desenvolvimento de técnicas de raciocínio em ligação íntima com os jogos.

Assim, não é de admirar que na mira de muitos investigadores informáticos estivessem os jogos de damas, do xadrez, o póquer, o bridge e outros. De realçar que não se trata nestes casos de fabricar programas invencíveis que explorem exaustivamente todas as alternativas possíveis de jogo, mas dotá-los de grande capacidade de pesquisa inteligente (sem ou com aspas, à vontade de quem lê) sobre a melhor (ou próximo dela) jogada em cada momento, em tempo útil.

2. Aplicações da «engenharia do conhecimento» em ciências como as matemáticas e muito principalmente em domínios científicos não exactos como a medicina. Tais sistemas incorporam conhecimento de especialistas num dado domínio para sobre uma base de conhecimentos e uma base de factos ou asserções, produzirem

² *Pattern Matching* é, sucintamente, o processo de comparar expressões simbólicas e se concluir da similitude entre elas, de modo a unificá-las.

conclusões fundamentadas (explicáveis) como suporte à ajuda de decisões.

3. Demonstração automática de teoremas. Conhecem-se os sucessos evidentes no campo da geometria. O estabelecimento de um plano de acção por um robô para cumprir uma certa tarefa pode ser entendido como um teorema a demonstrar.
4. Na programação automática investem-se grandes esperanças. De facto é aliciante a partir de algumas descrições (especificações) de alto nível e amostras de entradas no programa (dados) e das respectivas saídas esperadas pedir ao computador que crie um programa completo que execute as transformações desejadas.
5. Na Robótica a IA é essencial, dependendo no entanto muitíssimo do desenvolvimento tecnológico de outros ramos da engenharia.

Há quem discuta se qualquer sistema de IA é já por si um robô ou não.

Na verdade uma conhecida definição de robô reza que é uma máquina concreta, capaz de executar tarefas físicas em condições que parecem requerer certas qualidades humanas tais como faculdade de adaptação, autoterminação, capacidade de aprendizagem e sobretudo capacidades de representação do mundo, de predição e planificação.

Não obstante, a presença de captadores «sensoriais» evoluídos e manipuladores que ajam sobre o mundo parece ser em primeira análise suficiente para caracterizar um robô e distingui-lo das funções *software* puras envolvidas nos sistemas mais gerais de IA³. Relacionada com a robótica, a IA torna-se um instrumento imprescindível para o desenvolvimento de técnicas de análise e compreensão de cenas e imagens de televisão.

Ainda a possibilidade de «assemblagem», de visão tridimensional, de desambiguidade das sombras, tudo na base de sinais eléctricos vindos das câmaras de TV são já problemas de complexidade elevada onde a IA está presente. Mas desenha-se a tendência para distinguir em todos os processos aqui envolvidos entre os que

³ Ao mesmo tempo a introdução da IA é o que tem levado à evolução do puro autómato programado para o verdadeiro robô.

se denominam de periféricos e os centrais. Os primeiros, próximos do tratamento dos estímulos exteriores (transdutores ópticos e acústicos, processamento de imagem...) seriam postos fora do campo próprio à IA, enquanto os centrais, processos cognitivos envolvidos no raciocínio e planificação de acções, estariam nela incluídos.

7. A compreensão da linguagem natural escrita ou mesmo falada tem sido para a IA um ponto essencial já que através dessa facilidade todos os outros sistemas se tornarão bem mais acessíveis ao utilizador comum.

É um domínio onde muitos esperam muito. O que não é de todo pacífico. Primeiro pode pôr-se em causa a forma pouco «natural» como o computador aprende e ganha capacidade de linguagem... natural. Depois ainda outros se questionam se será correcto render a máquina a uma linguagem imprecisa e ambígua quando em todos os ramos da ciência se pesquisa em torno de linguagens formais o mais exactas possível.

8. Finalmente, na Psicologia acredita-se que ao criarem-se programas que resolvam problemas com estratégias semelhantes às do cérebro humano se fará nova luz sobre a psicologia do conhecimento, processamento e representação da informação.

As esperanças para o cidadão comum no mundo de amanhã são inúmeras e podem ser encontradas em qualquer livro de divulgação. De Winston (1979) respigamos algumas aplicações actuais ou em perspectiva:

- na agricultura: controlo de pesticidas, controlo do estado dos frutos, selecção dos cereais, colheitas.
- na mina: onde há perigo para o homem, o robô poderá substituí-lo.
- na fábrica: autómatos que fazem a montagem e inspecção de qualquer trabalho nomeadamente os repetitivos e que implicam uma certa precisão.
- na escola: ensino em geral, ensino da música, apresentação de modelos em movimento

como o planetário ou o atómico.

- no hospital: desde os mais complexos diagnósticos e prescrições de tratamento até ao mais simples acto de fazer as camas.
- no escritório: planificação e distribuição de trabalho, correcção de erros em documentos escritos.
- na casa: controlo de toda a cozinha, do *stock* de alimentos, cuidados do soalho, relva, roupa, etc. ...
- na guerra: também ...!

Uma representação diagramática dos assuntos principais na actual investigação em IA, tendo em vista a sua posição relativa na nuclear a periférica, poderá ser

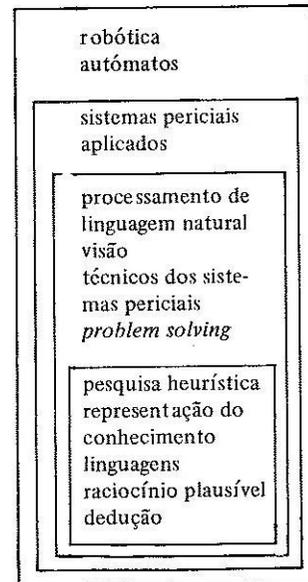


Figura 1

As realizações da ciência nas últimas décadas ensinou-nos a não rir das prospectivas mesmo que possam ser descabidas.

3. IMPLICAÇÃO

Se nos objectivos da Inteligência Artificial se inscreve a compreensão dos mecanismos de um comportamento mais ou menos inteligente seria

fatal que partindo embora da Ciência dos Computadores se fosse por um lado autonomizando desta e por outro esbatendo fronteiras com outras disciplinas que com ela se interligassem proficuamente. Tal é o caso nomeadamente com a Psicologia, a Filosofia, a Linguística.

Aliás os primeiros passos balbucientes da IA (Newell, Simon) foram altamente influenciados pela Psicologia, o que de todo não se tornou muito benéfico. Assim o paradigma do G.P.S. (*General Problem Solver*) de querer construir sistemas com toda a generalidade possível capazes depois de uma aprendizagem mais aprofundada *a posteriori* em vários domínios, à semelhança do comportamento cognitivo humano⁴, viria a ser inflectido por ineficaz.

De notar que querer tornar os computadores inteligentes não é o mesmo que querer que o computador *simule* inteligência. Mas também não se impede o estudo daqueles que se exercitam na procura de princípios que se apliquem a todos os processadores de informação quer sejam construídos de tecidos nervosos ou de *hardware* electrónico ...

Logo, não há em IA nem uma obsessão em minar a inteligência humana nem tão-pouco preconceitos contra o uso de métodos que parecem estar envolvidos no comportamento inteligente dos humanos.

Um aperfeiçoamento do trabalho em IA implica um melhor entendimento das ideias básicas da organização do conhecimento, do controlo de atenção, da pesquisa sistemática e heurística entre outras coisas. Daí que informações dos psicólogos possam ajudar a tornar os computadores mais inteligentes e simultaneamente teorias derivadas dos computadores talvez possam sugerir como ensinar melhor a pensar. Isto para além de haver quem garanta que as metodologias envolvidas em fazer programas mais espertos para os computadores podem ser aplicados para tornar as pessoas mais espertas também (veja-se a generalizada introdução de disciplinas de programação estruturada a vários níveis do ensino).

⁴ Aliás, recentes trabalhos parecem de algum modo querer contrariar essa ideia reconhecida. O sociobiólogo Edward Wilson assevera que estamos longe de poder aprender o que queremos, estando *ab initio* limitados e padronizados pela herança filogenética para mais facilmente aprendermos exactamente certas coisas e não outras.

Continuam aliás a desenvolver-se em paralelo as duas principais perspectivas dos Sistemas de Produção (P.S.)—de que nos ocuparemos detalhadamente a seguir: na criação de modelos psicológicos e por outro lado nos sistemas baseados em conhecimento especializado para tarefas em domínios específicos (*expert systems*). O que parece ser um esforço distinto na prossecução de diferentes objectivos levou, apesar de tudo, muitas vezes à adopção de metodologias similares.

Nada impede que muitos dos conceitos inventados e utilizados pelos investigadores em IA possam ser utilizados pelos psicólogos na intenção de construírem mais poderosos modelos explicativos da inteligência humana.

Se é um facto que a metodologia dos Sistemas de Produção é um veículo efectivo para a expressão e avaliação das teorias do comportamento, também enquanto aplicada aos *expert systems* está na base de programas que evidenciam competência em domínios específicos do conhecimento.

Repita-se que os esforços prosseguidos podem nada ter a ver (mas também podem ter) com similitudes entre os sistemas resultantes e as metodologias humanas. Neste caso o que se pretende é simplesmente realizar uma tarefa eficazmente sem qualquer erro (ou «emoção») sejam erros humanos ou não.

Aliás Newell inverte um pouco o problema ao considerar tais sistemas como importantes não pelo simples facto de com eles podermos aproximarmo-nos de modelos psicológicos, mas ao contrário, como uma metodologia que, por ser semelhante aos mecanismos fundamentais da cognição humana, tem inerente em si uma grande potencialidade de aplicação.

Ambos os campos—Psicologia e IA—estão de algum modo interessados em metodologias de resolução de problemas, compreensão da linguagem, representação e processamento da informação, mas percorrem caminhos diferentes na pesquisa dos mecanismos e funções que baseiam o comportamento inteligente. Enquanto os psicólogos focam sobretudo os problemas de aquisição, armazenamento e transmissão do conhecimento, os cientistas da computação focam de preferência a representação do conhecimento (sem é claro esquecer toda a restante problemática).

Não será descabido nomear aqui a Psicologia como a antidisciplina da IA seguindo de perto as

noções propostas pelo atrás citado Edward Wilson sobre as relações interdisciplinares.

Em conclusão, para a IA em geral e para a engenharia do conhecimento em particular não será fundamental em primeira análise o interesse paralelo, pelos P.S., dos psicólogos. Mas torna-se evidente que tal interesse nos Sistemas de Produção, agora para construir sistemas baseados em conhecimentos especializados altamente eficientes, ultrapassa a simples coincidência.

Sugere que esse fenómeno é o resultado da investigação (re)descobrimo o que fora já aprendido pelos sistemas inteligentes «naturais» através da evolução: — que estruturar conhecimento na forma dos P.S. é uma aproximação efectiva à organização, retribuição e uso de muito grandes quantidades de conhecimento.

ESTAREMOS NO LIMAR DE GRANDES DESCOBERTAS?

É sabido que o cérebro humano (onde podemos identificar o nosso poderoso *software*) não se desenvolveu senão quando encontrou um apropriado suporte físico.

É errado pensar que o *hardware* do computador não é mais que o «veículo tangível para a intangível substância do *software*».

Ambos os conceitos se interligam, se impulsionam ou se retêm.

E se o *hardware* se desenvolveu espantosamente nos últimos tempos nos sentidos da maior velocidade, menor potência requerida, maior integração, novas arquitecturas, o que causou uma certa «esquizofrenia» *hard/soft*, é bem visível agora o salto qualitativo que a IA (e não só) fez dar ao *software*.

Se for tida como definitivamente verdadeira, na teoria da organização e funcionamento do cérebro, a tese segunda a qual ele está funcionalmente dividido em dois hemisférios interligados pelo chamado corpo caloso, fácil se torna concluir que destes dois «computadores» co-residentes no homem, só um deles está parcialmente reflectido na «inteligência» que tentamos produzir.

Assim os nossos sistemas de IA, como computadores de Von Neumann que são e portanto respeitando *ad liminae* a sequencialidade estrita, estão próximos do hemisfério cerebral esquerdo, enquanto que o hemisfério cerebral direito, ao que parece responsável pela produção criativa mais

imaginística e emocional que verbal, continua momentaneamente fora do nosso alcance de «deuses» geradores de máquinas.

Há quem argumente que a evolução da «reconstrução» pelos humanos da «inteligência» teria forçosamente de ser esta, começando pelas funções mais imediatamente formalizáveis do hemisfério esquerdo, crescendo em complexidade e abstracção até à completa reprodução, no futuro, das requintadas características do hemisfério direito.

Os sistemas poderiam então ser logicamente preditos de acordo com a evolução já sofrida pelos seus construtores.

Este «imperativo biológico» é um mandamento predicado em algumas «orações».

É certamente um pouco cedo para discutir profundamente nesses termos. Mas há pelo menos uma característica intrínseca do hemisfério cerebral direito que começa a ser uma preocupação real da investigação na Ciência da Computação: o paralelismo.

4. PARTICULARIZAÇÃO

OS SISTEMAS DE PRODUÇÃO COMO SISTEMAS DE INFERÊNCIA

Estrutura genérica

Em situações onde novos estímulos, novas situações, se levantam continuamente (quer vindas do meio exterior, quer de condições internas geradas pelo próprio programa), um código puramente sequencial, mesmo com variantes mais ou menos rígidas, pode ser, é com certeza, inapropriado.

Em tais situações o programa é melhor idealizado como uma colecção organizada de módulos-guiados-por-padrões⁵ (*Pattern-Directed Modules*) os quais são responsáveis quer pela detecção de várias situações típicas entre os dados quer por lhes responder de forma apropriada.

Uma espécie importante de macro-operação empregada por tais módulos (os PDM's) envolve a leitura de informação numa estrutura de dados para

⁵ Tais padrões (*patterns*) podem ser definidos de muitas formas, desde simples *strings*, grafos mais ou menos complexos, estruturas de redes semânticas até a segmentos arbitrários de códigos capazes de inspecionar elementos de informação.

depois através de vários testes verificar se há ou não compatibilização com algum *molde* (ou arquétipo) particular dos que estão associados ao PDM. Tal actividade é chamada «*pattern-matching*» e justifica o nome dos PDM's.

De uma forma ainda um tanto geral, digamos que qualquer sistema composto de vários PDM's que realizam através de «*pattern-matching*» a leitura e modificação de estruturas de dados, em conjunto com um executivo que selecciona e executa os PDM's é chamado de Sistema de Inferência Dirigida por Padrões (*Pattern-Directed Inference System*).

Neste caso, o programa responde directamente a uma larga banda de dados ou acontecimentos (possivelmente não antecipados) e não opera em tipo de dados já esperados, em formatos pré-definidos e usando uma estrutura de controlo inflexível e pré-especificada como é o caso dos programas convencionais.

Aqui são determinados padrões ocorrendo nos dados que seleccionam segmentos de código no sistema para serem activados. Tais sistemas têm potência suficiente para serem usados quer para inferência dedutiva quer, em menor escala, para inferência indutiva ou aprendizagem.

Sistemas de Produção (*Production Systems*), *Pattern Directed Inference Systems* (PDIS), Sistemas baseados em regras (*Rule-Based Systems*), Sistemas baseados em conhecimento (*Knowledge-Based Systems*), são alguns dos termos muitas vezes usados indiferentemente para classificar os formalismos que enformam os sistemas usados na IA.

Não obstante esta confusão, tais nominativos denominam realidades diferentes se bem que intimamente relacionadas hierarquicamente e por isso partilhando várias características, diferenciando-se noutras. F. Hayes-Roth detalharam, em 1978, uma taxomia elucidativa e apropriada para tornar claro todo este marasmo classificativo.

Porém, antes de a apresentarmos, queremos introduzir aos meandros estruturais (genéricos) dos P.D.I.S., por um relacionamento breve com o exposto na secção anterior.

Um modelo contemporâneo do processamento da informação na cognição humana pode ser esquissado assim:

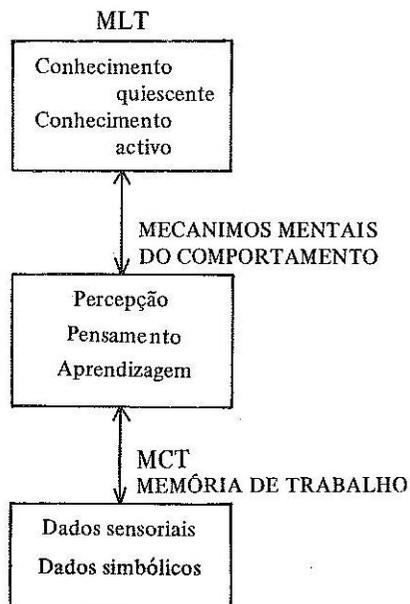


Figura 2

Ou seja: dois tipos de memória e um executivo. A memória de longo termo (MLT) compreende o conhecimento quiescente e o conhecimento activo.

O primeiro consiste na informação de base (inconsciente, subconsciente, hereditário, filogenético) não correntemente acessível ao processo do raciocínio.

O segundo é correntemente relevante e facilmente acessível.

A memória de curto termo (MCT) é memória de trabalho, guardando informação temporária quer de dados sensoriais representando resultados dos estímulos sensoriais (visuais, auditivos, tácteis e outros) quer de dados simbólicos representando resultados temporários e/ou intermédios do processo cognitivo. Normalmente a MCT contém apenas um pequeno número de «posições» (7 ± 2) cada uma das quais pode guardar um pedaço de informação.

Um P.D.I.S. completo pode também ser esquematizado num modelo com dois tipos de memória e um executivo.

O executivo, que é o mecanismo de inferência propriamente dito, é constituído por um monitor (que detecta as modificações efectuadas nas memórias).

Um «*pattern-matcher*» (que compara as situa-

ções observadas com as que estão definidas para permitir o desencadeamento das respostas através das regras de inferência). Este «pattern-matcher» ao sugerir várias acções envia-as para uma agenda que surge como base de ordenação para as futuras computações. Tal selecção é feita por um módulo seleccionador («scheduler») segundo determinados critérios e depois cabe ao processador calcular e implementar tais acções assim seleccionadas.

Ainda um bloco de modificação do conhecimento que efectua as modificações necessárias na base de conhecimento activo e ajuda nas modificações permanentes a serem feitas nas regras ou factos.

A base de conhecimento pode ser comparada com a MLT e nela se destacam três componentes:

O conhecimento quiescente (informação não correntemente envolvida na actividade de inferência);

O conhecimento activo (informação permanentemente relevante para o problema em causa);

O meta-conhecimento (conhecimento sobre o próprio conhecimento; Guias sobre como activar o conhecimento).

A memória de trabalho contendo a já referida agenda com os PDM's passíveis de aplicação na si-

tuação corrente, e a memória de dados simbólicos (repositório das representações do problema, objectivos e resultados intermediários).

Ora este modelo pode ser considerado como uma realização do anterior. Apenas com alguns desvios de focagem como sejam: onde os objectivos de pesquisa Psicológica eram compreender *Quando, Onde e Porquê* o conhecimento é armazenado, os objectivos da IA têm sido compreender *Qual* (conhecimento é preciso para realizar uma dada tarefa) e *Como* tal conhecimento deve ser ligado aos dados apropriados e aplicados para os alterar (arquitectura para os sistemas de inferência).

O que se compreende, pois num caso se trata de analisar o que é feito e noutro se trata de conseguir que seja feito.

Classificação

Para esta visão geral sobre os P.D.I.S. e para melhor ver claro dentro deles passa-se a uma especificação taxonómica necessária.

PROGRAMA (P): Definamos aqui programa como sendo uma colecção arbitrária de instruções que podem ser executadas por um computador, sendo o seu compo. tamento dependente da informação quer externa quer interna.

(Colocou-se esta raiz P no grafo acima para su-

Esquema de um PDIS

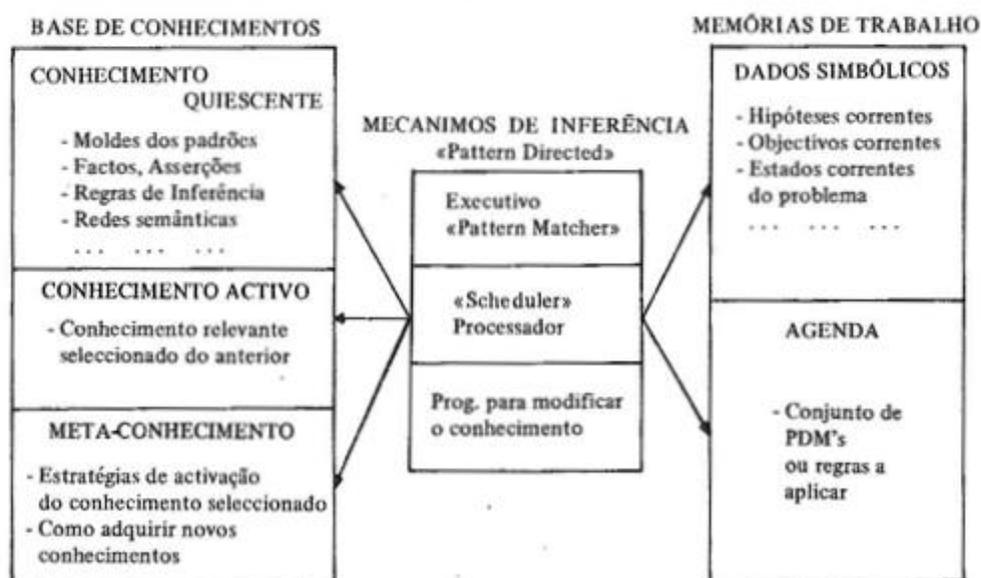


Figura 3

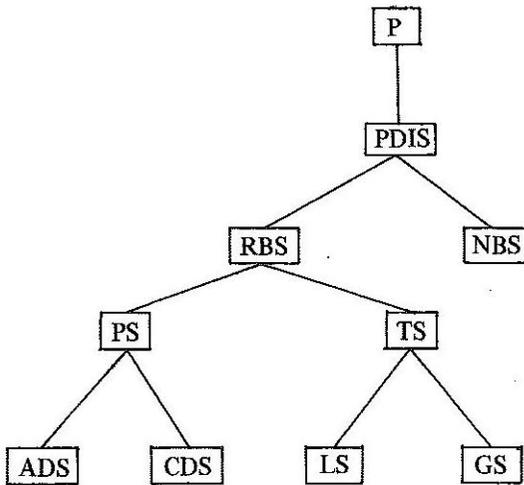


Figura 4

blinhar qual o ponto de vista sobre o qual os sistemas referidos são tomados).

SISTEMAS DE INFERÊNCIA GUIADOS POR PADRÕES – (*Pattern-Direct Inference Systems* – P.D.I.S.): Programa organizado como uma colecção de módulos «Pattern-Directed» operando em estruturas de dados com o objectivo de fazer o «matching» de tais padrões («pattern-matching») e modificar os dados através de um executivo que controla a execução dos PDM's.

SISTEMAS BASEADOS EM REGRAS (RBS): Nestes P.D.I.S. os módulos (PDM) são regras. Cada regra terá em separado um lado esquerdo contendo as possibilidades de acesso (de leitura) e um lado direito contendo as possíveis acções (acessos de escrita).

SISTEMAS TRANSFORMACIONAIS (*Transformation Systems* – TS): RBS em que as operações de «matching» e selecção (*scheduling*) não são necessariamente partes explícitas do sistema.

SISTEMAS DE PRODUÇÃO (*Production System* – PS): São sistemas baseados em regras nas quais as partes de «matching» e selecção (*scheduling*) são explícitas e autónomas, definidas pela operação do executivo.

SISTEMAS GUIADOS PELOS ANTECEDENTES (*Antecedent Drive System* – ADS): Sistemas de

Produção que usam os antecedentes das regras (a sua parte esquerda) como guias para a selecção das regras a activar.

SISTEMAS BASEADOS EM «NETWORKS» (NBS): Os PDM's localizam-se nos nós de um «network» e são activados por sinais nos arcos de entrada.

SISTEMAS GUIADOS PELOS CONSEQUENTES (*Consequent Drive Systems* – CDS): PS que usam os consequentes das regras (a sua parte direita) para seleccionar as regras a activar.

SISTEMAS LÓGICOS (*Logical Systems* – LS): Sistemas transformacionais aplicados em problemas específicos da lógica formal como p. ex. a demonstração de teoremas (*Theorem proving*).

SISTEMAS GRAMATICAIIS (*Grammatical Systems* – GS): Sistemas Transformacionais usados para definir e processar gramáticas (*parsing*, geração, etc.).

Podemos portanto dizer que as ideias base que caracterizam os P.D.I.S., são:

a) Que a atribuição do controlo a um dos elementos-de-código (fornecidos pelo utilizador) é condicionado permanentemente pelo evoluir de uma colecção de dados ou factos separados dos elementos do código. Tal colecção é interpretada como uma descrição da situação corrente do sistema. É o aparecimento de certos tipos de dados que origina a selecção dos elementos de código a serem activados e *não* um mecanismo próprio definido pelo utilizador que passa o controlo de um a outro elemento de código.

b) Por via de a) espera-se que tais sistemas facilitem a representação de processos cuja conduta global dependa em cada instante de situações «específicas, interessantes ou importantes», separando um «conhecimento de situação» de um «conhecimento operativo». O esquema de controlo, também já o referimos, pode ser decomposto em 4 fases:

Seleção: dos PDM's e elementos de informação relevantes.

«*Matching*»: comparando os PDM's activos com os elementos-de-informação também activos, procurando padrões que permitam a correlação (o «*matching*»).

«*Scheduling*»: por algoritmos de prioridades ou apropriação decidir qual o PDM a ser escolhido.

Execução: executar as acções relacionadas com tal PDM.

Sistemas de Produção – estrutura particular

Dos P.D.I.S., aqueles que para as aplicações da IA encerram mais interesse são os Sistemas Baseados em Regras (RBS) e dentro destes o largamente divulgado Sistema de Produção (PS – «Production Systems»).

Nos RBS cada módulo (PDM) é um par (E, A) que no caso dos PS se chama regra de Produção⁶. Em E está consubstanciado o exame a realizar sobre a base de situação antes de activar o módulo. Em A as eventuais acções após os exames. Tais acções podem ser simples modificações sobre a base de situação ou base operatória ou ainda comunicações com o ambiente (p. ex. interrogatórios) ou mesmo intervenções propriamente ditas no meio (caso de actos de um robô). São regras do tipo Situação-Acção.

Os Sistemas de Produção têm uma importância invulgar e são geralmente apontados como sistemas gerais de programação incorporando a potência completa de uma máquina de Turing (mas usando uma codificação do conhecimento homogéneo) (Davis e King, 1977). Aliás a sua carreira tem sido brilhante desde que inicialmente foram usados por Post, em 1943, em lógica simbólica. Aparecendo depois como algoritmos normais de Markov, em 1954, são introduzidos na linguística como regras de reescrita por Chomsky, em 1957 – donde a sua notoriedade. E, como muitas

outras acções nas Ciências dos Computadores, transformam-se qualitativamente e alargam a sua influência quando se tornam operacionais em linguagens de Programação e linguagem de tradução de compiladores.

Refira-se que são precisamente os P.S., e não outros P.D.I.S., os que Herbert Simon considera como sendo os bons candidatos à modelação do processo cognitivo humano. Na verdade, além da aludida generalidade computacional da máquina de Turing universal, contém ainda possibilidades de criação e adição incremental das regras e a sua base de dados (BD) modela a MCT enquanto que o corpo de regras de Produção dão um modelo possível da MLT humana.

Não basta porém continuar a dividir o P.S. em:

- 1 – Conjunto de regras do tipo situação-acção
- 2 – Base de dados
- 3 – Interpretador seleccionador de regras.

Os P.S., são um tipo de RBS nos quais a estrutura de controlo (o interpretador, o executivo) pode ser englobado num paradigma relativamente simples do ciclo Seleção-Execução, ou, dado que seleção se reporta à escolha de regras que se correlacionam com a base de dados corrente, de Conhecimento-Acção (ou também Decidir-Agir)⁷. Este interpretador que diz entre outras coisas quais dos módulos filtrados os que devem ser activados, utiliza tanto a base de dados como a colecção de módulos em modo associativo (portanto com acesso por conteúdo, pelo menos em parte, e não pelo endereço).

É na parte «Decidir» que se englobam as operações de seleção (qual o subconjunto de regras R_1 e factos que vão ser comparados), *matching* (escolha das regras R_2 que resultaram) e resolução de conflitos (qual a Regra entre as de R_2 que vai ser executada).

Esquemmatizando:

⁶ Uma produção P_i é uma instrução condicional, composta por condições e acções, de forma:

$$P_i = (C_1, C_2, \dots \rightarrow A_1, A_2, \dots A_m)$$

⁷ Há aqui continuamente (em cada ciclo Decidir-Agir) uma reavaliação do estudo do sistema e consequente redistribuição do controlo (o que evidentemente é custoso em termos de computação).

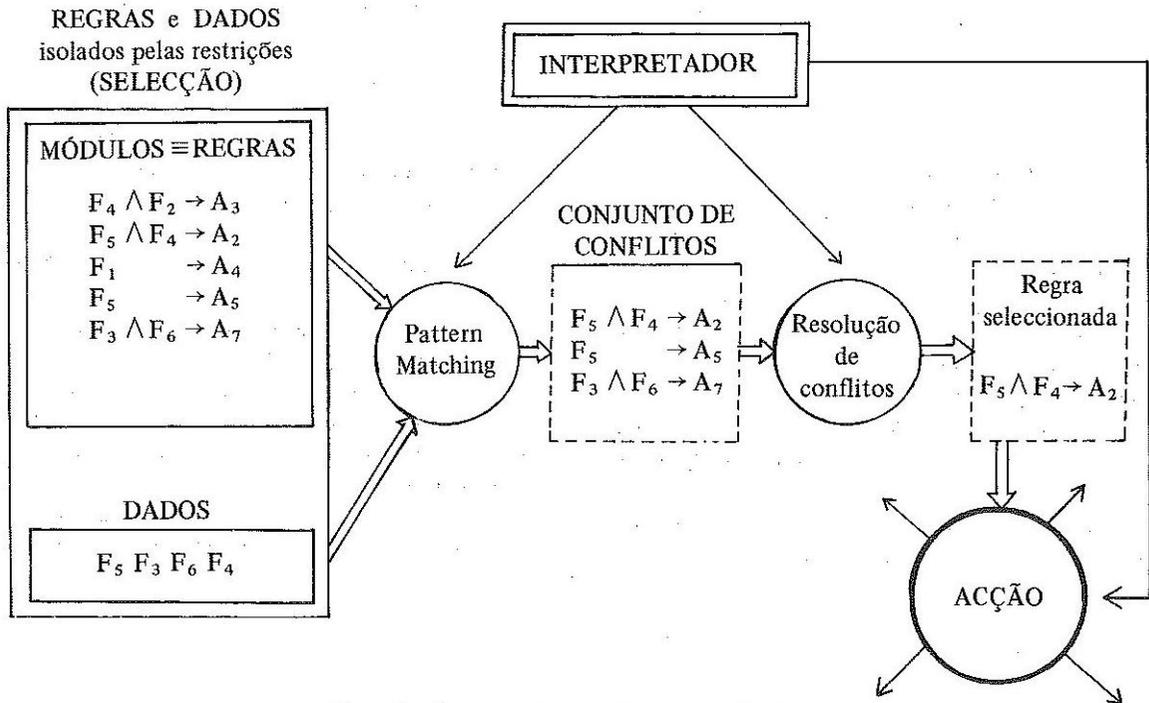


Figura 5 – Execução de um ciclo depois da Restrição

Os critérios seguidos para a resolução de conflitos, i. e. para a escolha de uma regra de entre as do conjunto de conflitos podem ser diversos, como:

- i) *Ordenação das regras*: Há uma ordenação completa de todas as regras já seleccionadas, e é escolhida a de maior prioridade pós-ordenamento.
- ii) *Ordenação dos dados*: Os elementos da base de dados (BD) são ordenados e é escolhida a regra que faz o «match» com o(s) elemento(s) na BD de maior prioridade.
- iii) *Ordem de generalidade*: A regra considerada mais específica (menos geral) é escolhida.
- iv) *Precedência de regras*: Um esquema de precedências contido nas próprias regras (possivelmente com ciclos) determina a hierarquia de resolução.
- v) *Ordem de Recenseamento*: Por escolha quer da regra executada mais recentemente ou da regra contendo os elementos da BD mais re-

centemente escolhidos.

Segundo Davis e King, (1977), o interpretador do Lisp 70 utiliza iii) enquanto que um dos mais conhecidos «expert systems» que a seguir referimos – Dendral – usa iv).

Os sistemas de Produção guiados pelos antecedentes (ADS) são os mais conhecidos como P.S.. O antecedente é uma combinação lógica de proposições acerca da BD ou da memória de trabalho. Quando o antecedente é verdadeiro, o consequente, uma colecção de regras que podem eventualmente modificar a memória de trabalho, é computado e as acções a ele associadas são executadas. Dito doutra forma: o exame E é feito sobre a base de situação (factos ou asserções). Logo que tais factos visados por E são estabelecido prioritariamente à utilização da regra respectiva diz-se que o P.S. é guiado pelos antecedentes. Se os factos que são visados por E não são, eventualmente, estabelecidos senão depois da utilização da regra, diz-se que o P.S. é guiado pelos consequentes.

Neste caso (CDS) tanto o antecedente como o consequente são asserções acerca dos dados. E a selecção e execução das regras são guiadas numa

cadeia inversa envolvendo o «matching» dos consequentes das regras com as asserções a serem provadas.

Nos ADS⁸ o interpretador considera portanto os dados utilizados como conhecimentos estabelecidos a partir dos quais outros serão derivados (pelas chamadas *Forward rules*). Nos CDS os exames põem em relação os padrões das regras com dados considerados como conhecimentos a estabelecer. Se o consequente de uma regra é reconhecido adaptada às hipotéticas asserções (figurando na BD) que nos propomos estabelecer, tal regra poderá ser activada.

A execução de um ADS pode ser representada como uma série de ciclos reconhecimento-acção tal como o evidenciado no diagrama que se segue (que aliás não difere qualitativamente do esquema anterior).

-directed Interactive Transaction Agent) e diagramar a parte capaz de fazer a subtracção inteira (P-Q) (Watterman, 1978). Este exemplo tem a vantagem de mostrar como a base de dados pode não ser uma simples listagem de factos mas estruturas mais complexas. No caso presente a BD é uma colecção de triplos «objecto-atributo-valor» o que aliás acontece em PS dos mais sofisticados e bem conseguidos como é o caso do Mycin que também referiremos.

Não é difícil adivinhar o que se passa se reconsiderarmos que as regras são testadas por ordem. R₁ é a primeira a ser executada apagando da memória o elemento INÍCIO e juntando o elemento (objecto) CONTADOR cujo valor (atributo)⁹ é zero (valor). Seguidamente é R₃ que pode ser seleccionada, e é-o repetidamente incrementando os números associados com CONTADOR e Q até

RECONHECIMENTO

BASE DE DADOS: C₅ C₁ C₃



Figura 6

Como ilustração simples de um Sistema de Produção ADS vamos servir de um Sistema de Produção considerado de alto nível (RITA - *Rule*

que os números associados (valores) a P e Q são iguais. Aí chega a vez de ser R₂ a escolhida. Daí virá o resultado e o fim da operação.

⁸ É interessante notar que PS deste tipo são usados por alguns para descrever processos assíncronos concorrentes: «*Because of event-driver nature of asynchronous concurrent processes, productions are an attractive modeling too.*» (Michael Zisman.)

⁹ Não confundir os dois usos da palavra valor. Aqui, por acaso, o atributo do objecto é o seu valor como poderia ser noutro caso cor ou nome com os respectivos valores de verde, ANA, etc.

Base de Dados: OBJECTO início; OBJECTO_p;
OBJECTO_q; valor É 5; valor É 3;

REGRAS:

R₁: SE: EXISTE início
ENTÃO: APAGAR o OBJECTO início
∧ CRIAR um contador cujo valor É 0;

R₂: SE: EXISTE contador
∧ valor de p É valor de q
ENTÃO: ENVIE valor de contador para o
UTILIZADOR
∧ REGRESSE com SUCESSO;

R₃: SE: EXISTE contador
∧ EXISTE q
ENTÃO: COLOQUE valor do contador em
1 + valor do contador
∧ coloque valor de q em 1 + valor de
q;

Figura 7

Um exemplo abstracto de um CDS é dado a seguir:

Base de Dados: B C

REGRAS:

R₁: B ∧ D ∧ E → F
R₂: D ∧ G → A
R₃: C ∧ F → A
R₄: C → D
R₅: D → E
R₆: A → H

Figura 8

Assumamos agora a seguinte premissa: *H* é verdadeiro. Ora o sistema uma vez que não consegue encontrar na BD o facto *H*, vai tentar deduzi-lo por encadeamento inverso através das regras. É o consequente de R₆ que faz com que ela seja escolhida. O sistema passa a tentar provar que *A* é verdadeiro pois isso implicaria que a hipótese inicial também o fosse. *A* não pertence à BD logo precisa de ser deduzida pelas regras R₂ ou R₃. Se R₂ for escolhida primeiro então há que provar que *D* e *G* são verdadeiras. *D* pode ser inferido de R₄ mas sobre *G* nada pode ser concluído. Auto-

maticamente o sistema regride e tenta aplicar R₃ onde aplicara R₂ para inferir *A*. Isso implica que *C* e *F* sejam verdadeiros. Que *C* é verdadeiro é um facto. A veracidade de *F* (por R₁) depende da de *D* e *E* já que *B* pertence à BD. *D* é provado através de R₄ (*C* pertence à BD) e *E* por R₅. Finalmente ficou provado que *H* era verdadeiro. O encadeamento seguido foi:

```

C D
C → D
  |
  D → E
    |
    B D E → F
      |
      C F → A
        |
        A → H

```

Figura 9

Os sistemas de Produção são vistos por alguns, como Rychener, como linguagens de Programação abstractas e usadas como veículo para explorar as estruturas de controlo.

Se considerarmos os PS «puros» podemos caracterizá-los para além da sua modularidade (regras independentes) por uma grande «abertura».

Esta «abertura» é caracterizada pelo princípio de que qualquer regra pode ser executada em qualquer instante. Portanto, em qualquer ponto da computação todas as regras são em princípio candidatas a serem seleccionadas a seguir, dependendo unicamente, claro está, do conteúdo da BD no fim de cada ciclo.

Se compararmos esta situação com a habitual nas linguagens procedimentais, onde evidentemente há uma certa fixidez na ordenação da chamada dos «procedimentos», compreendemos melhor o alcance de tal característica.

Enquanto que o programador nas linguagens de alto nível procedimentais (p. ex. Algol) escolhe cuidadosamente a ordem de chamada dos procedimentos para criar uma sequência bem escolhida de ambientes, num PS é este ambiente que de certo modo escolhe a próxima regra (e portanto também as acções, os procedimentos) a serem executados.

No entanto pode-se objectar uma certa falta de

visibilidade do comportamento dos PS em comparação com os formalismos procedimentais. Tal opacidade deve-se à representação do conhecimento seguido pelo PS.

Estes são apropriados quando é possível especificar o conteúdo do conhecimento requerido sem contudo especificar a forma como ele vai ser usado. Logo, ler um PS não torna geralmente clara a forma como ele actua¹⁰. Esta situação é geralmente invertida em linguagens procedimentais onde o comportamento do programa é mais evidente que o domínio específico que os procedimentos encerram.

Ainda, ao contrário das linguagens procedimentais não há necessidade para armazenamento separado de informação sobre o estado de controlo, nenhum «Program Counter», pilhas, etc. Nada a não ser a simples BD, e toda a informação a ser guardada irá para lá.

SOBRE A IMPLEMENTAÇÃO

Finalmente o fecho deste capítulo podia ser o início de toda uma outra discussão relacionada com as linguagens de programação em lógica, em especial a Prolog. Digamos que o interpretador Prolog não é mais que um Sistema de Produção guiado pelos consequentes (CDS) com uma informação de controlo muito específica acerca da sequência das acções, incorporando mecanismos de retrocesso sistemático («backtracking») (Nilsson, 1980).

Pode sustentar-se que o Prolog resolve algumas das incapacidades lógicas de outras linguagens nascidas (e algumas abandonadas) para a IA como o Planner, QA₄, Conniver. Também Hayes discute as relações entre a lógica de predicados de conhecimento baseadas em «frames» como as KRL-0, KRL-1 desenvolvidas por Bobrow e Winograd em 1977.

O Prolog tem muitos paralelos com o Lisp (a linguagem tradicionalmente aplicada na IA) sendo ambas linguagens interactivas, projectadas para o processamento simbólico de dados e baseadas em sistemas matemáticos formais — Lisp no Lambda-Calculus de Church, o Prolog num subconjunto da lógica clássica (as clausuras de Horn).

¹⁰ A utilização das linguagens lógicas como o Prolog torna os programas bastante mais legíveis.

5. APLICAÇÃO

«In most cases basic research is only now becoming engineering practice.»
(Winston, 1979)

QUANDO E PORQUÊ

Tínhamos já apontado oito campos de aplicação da IA.

Mas aquele que (segundo Saul Amarel) mais distingue em IA um projecto aplicado de IA da pesquisa básica é o campo da Engenharia do Conhecimento.

QUAL É O ESCOPO?

Como já referimos, nos últimos anos deu-se uma deslocação do antigo paradigma do «General Problem Solver», tentativa de construção de sistemas com a generalização suficiente para serem posteriormente utilizados para apreensão (e aprendizagem) de qualquer tipo de conhecimento e consequente largo espectro de acções possíveis. Paradigma que não convenceu.

Essa lição do passado recente levou a que os investigadores da IA concluíssem que o poder de realização (de solução de problemas) exibido por um agente «inteligente», é em primeiro lugar consequência do conhecimento especializado detido por tal agente e só muito secundariamente relacionado com a generalidade do método de inferência que ele emprega. Numa frase «our agents must be knowledge-rich, even if they are methods-poor» (Feigenbaum, 1979).

A Engenharia do Conhecimento conta, evidentemente, com o transporte dos princípios e métodos da pesquisa da IA para aplicar em problemas, porventura difíceis, requerendo conhecimento especialista para serem solucionados.

— Quando o domínio é visto e interpretado como consistindo de muitos estados independentes (e não unificado por uma teoria precisa e concisa);

— Quando a complexidade do fluxo de controlo se partilha por um conjunto de acções independentes (e não em colecção de processos paralelos envolvendo vários subprocessos dependentes);

— Quando o conhecimento incorporado no sistema pode ser separado da maneira como ele vai ser utilizado, isto é, se o sistema contém uma representação declarativa (e não procedimental,

caso em que o uso do conhecimento está em grande parte predeterminado durante o processo de incorporação do conhecimento no sistema):

Então poderemos dizer que temos um domínio apropriado para sobre ele criar um sistema pericial («expert system»).

A existência de múltiplos, e não trivialmente diferentes, estados independentes, é uma indicação segura da possibilidade de escrever múltiplas, e não triviais, regras modulares. Também um processo composto por um conjunto de acções independentes requer somente comunicações limitadas entre essas acções.

Tudo isto e ainda a citada possibilidade de descrever convenientemente o conhecimento específico no sistema sem impor *à priori* uma forma de uso, são características fundamentais que recomendam os Sistemas de Produção para dar corpo aos «expert systems».

Também podemos dizer que a Engenharia do Conhecimento é o conjunto de processos de construir sistemas periciais. Ou seja, a construção de programas que incorporam conhecimento específico de um domínio vertido de especialistas e que o aplica para produzir conclusões úteis para o utilizador potencial.

Hoje o problema não é tanto o da identificação de uma quantas técnicas mas, e em elevado grau, a questão de como representar grandes quantidades de conhecimento de forma a que a sua manipulação e a sua interacção seja eficiente e rica.

Estas dificuldades incluem em primeiríssimo plano a inquirição ao(s) especialista(s) do seu conhecimento tantas vezes mal estruturado ou deficientemente integrado.

Ao «Engenheiro do Conhecimento» acrescem as dificuldades de:

- Ser geralmente alheio aos domínios e disciplinas cujo conteúdo quer incorporar em programas;
- Esse conhecimento provindo de domínios onde não há teorias bem axiomatizadas mas colecção de princípios e generalizações empíricas ligadas com regras *ad-hoc* contemplando as numerosas excepções, é por isso largamente heurístico, experimental, incerto;
- Geralmente tal conhecimento é privado e dificilmente partilhado quer por não ser fácil

«saber sobre o que se sabe» quer por ciosidade alheia à ciência.

No sentido mais estrito, um PS não é definido plenamente senão quando um conjunto preciso de regras é fornecido, e não somente por uma linguagem de expressão, estruturas de representação e um interpretador. Só assim o sistema especializado por esse conjunto de regras é qualificado de «expert system».

O que não quer dizer, e isso é interessante por parecer contrariar a tendência anteriormente seguida de afastamento do GPS, que não se tenha tentado tornar independente do domínio, certos sistemas particularmente bem sucedidos. Quando tal se consegue qualificam-se de Sistemas Essenciais ou Gerais.

(É o caso do Emycin que ressalta do sistema Mycin depois de se lhe retirarem as regras que o dedicam ao diagnóstico de doenças infecciosas no sangue.)

Há quem assevere que esta interdisciplinaridade permitida e promovida pela IA, mais do que fornecer sistemas eficientes ao serviço de diferentes disciplinas, como a Medicina, terá um impacto ainda muito maior do que o previsto. Isso dever-se-ia à necessidade absoluta injectada em tais domínios da classificação de conceitos básicos, da identificação das relações entre várias partes do mesmo conhecimento, do reconhecimento dos erros, das faltas, dos desconhecimentos.

De qualquer forma reconheça-se que (e paradoxalmente) os tipos de especialidades mais difíceis de adquirir para a gente comum (conhecimentos para-científicos) são mais fáceis de articular para benefício dos sistemas especialistas do que algumas propriedades partilhadas até entre homens e outros animais (como a visão).

ALGUNS CASOS EXEMPLARES

São já bastantes os domínios em que a IA, introduziu sistemas periciais.

Alguns dos mais conhecidos são:

Na Medicina:

Mycin desenvolvidos em Stanford por E. H. Shortliffe (Shortliffe, 1976) é um sistema que assiste no tratamento de doenças infecciosas do

sangue.

Casnet:

Sistema desenvolvido na Universidade Rutgers por Kulikowski e Weiss, que ajuda os diagnósticos e terapias de glaucomas (doença oftálmica).

Internist:

Da Universidade Pittsburgh, realizado por Harry Pople em 1977, e virado para a medicina interna.

Hearsay II:

Sistema construído por Lesse e Ermen em 1977 cujo domínio se situa no reconhecimento e produção de sons e frases.

AM:

Sistema baseado em conhecimentos matemáticos, que conjectura alguns conceitos interessantes na matemática elementar.

Molgen:

Sistema realizado por Stefik em 1978 na Universidade de Stanford, que dá conclhos na genética molecular a propósito da planificação de experiências envolvendo manipulações de DNA.

Crysalis:

Sistema criado por Nii e Feigenbaum, que infere estruturas proteicas a partir de mapas de densidade dos electrões, derivados de dados dos raios X.

Dendral:

Um sistema pioneiro desenvolvido na Universidade de Stanford por Feigenbaum e Bauchanan, que enumera estruturas plausíveis para moléculas orgânicas sendo dados dois tipos de informação: dados analíticos da espectrometria de massa e do espectrómetro da ressonância magnética do núcleo e informações restritivas fornecidas interactivamente pelo utilizador.

Meta-Dendral:

Que infere regras a partir da fragmentação de moléculas no espectrómetro de massa, para utilização posterior pelo sistema Dendral.

Teiresias:

Saído das dificuldades do Mycin para adquirir novos conhecimentos.

Assim permite a interacção como utilizador do Mycin para lhe permitir detectar e modificar regras anteriores assim como acrescentar outras.

Prospector:

Sistema criado por Duda e Hart que permite aos geólogos analisarem e localizarem depósitos minerais.

De realçar ainda o sistema Orbi para o ordenamento biofísico do território desenvolvido por L. M. Pereira, Eugénio Oliveira e P. Sabatier no núcleo de IA da UNL.

Este sistema que comunica em português com o utilizador foi desenvolvido utilizando formalismos lógicos e implementado em Prolog.

Incorpora regras de especialistas em várias disciplinas como geologia, hidrologia, fauna, flora, etc., para tirar conclusões num domínio multidisciplinar como o ordenamento do território, conclusões essas suportadas pela explicitação dos raciocínios efectuados.

Sistema Mycin (consultas médicas baseadas em computador)

Um dos domínios onde mais claro se tornou o benefício introduzido pela IA, é a medicina. Um exemplo da AIM (Artificial Intelligence in Medicine), é o sistema, geralmente apontado como modelo de eficiência e aceitação, Mycin¹¹.

Mycin é um sistema de programas que tem por objectivo assistir médicos que não sejam especialistas no diagnóstico e terapêutica de doenças infecciosas no sangue.

O trabalho começou em Stanford em 1972, estava parcialmente elaborado em 1976 (data do

¹¹ Mycin deriva o seu nome do sufixo comum a grande parte de agentes antimicrobióticos.

livro de Shortliffe) e continuou daí para cá o seu desenvolvimento sempre num ambiente de investigação.

Fundamentalmente as conclusões por ele fornecidas são:

- 1 – Se o paciente tem ou não uma infecção significativa;
- 2 – Qual a identidade do organismo ofensivo;
- 3 – Quais os medicamentos que combatem tal organismo;
- 4 – Qual a droga mais apropriada, dada a condição clínica do paciente.

O sistema contém uma base de conhecimento que na altura da publicação do seu livro era composta por 200 regras e que deppis passou a 400. Segundo as últimas informações continha cerca de

A base de dados é constituída pelas indicações fornecidas pelo médico sobre o paciente e ainda uma parte dinâmica que é criada durante a consulta por estruturação das informações recebidas interactivamente.

O programa propriamente dito consiste de três módulos:

Sistema de consulta, Sistema explicativo e Sistema aquisitivo.

O primeiro faz perguntas, tira conclusões e dá conselhos. O segundo responde a perguntas do utilizador e fundamenta o seu diagnóstico. O terceiro permite a adição de novas regras e alteração das preexistentes.

A linguagem de codificação utilizada foi um dialecto do Lisp, o Interlisp.

A base de conhecimentos sendo constituída por regras (do tipo IF-THEN) é altamente modular. Segue-se um exemplo:

- IF: 1) The stain of the organism is GRAM-POS, AND
 2) The morphology of the organism is Coccus, AND
 3) The growth conformation of the organism is Clumps

THEN: There is suggestive evidence (.7) that the identity of the organism is Staphylococcus

A modularidade vem sobretudo do facto de não haver nenhuma referência explícita a relações entre várias regras.

Embora as primeiras regras tivessem sido incorporadas no sistema pelo informático após longas sessões com a equipa de médicos que discutiu a história clínica de casos representativos, as regras posteriores foram metidas através de interacção utilizador-sistema aquisitivo.

O comando NR (new rule) indica ao sistema aquisitivo essa interacção e este recebe (como se indica abaixo) as permissas e a conclusão em linguagem natural traduzindo-as internamente para Lisp.

(** é o prompt do Sistema; CR indica «carriage return».)

** NR

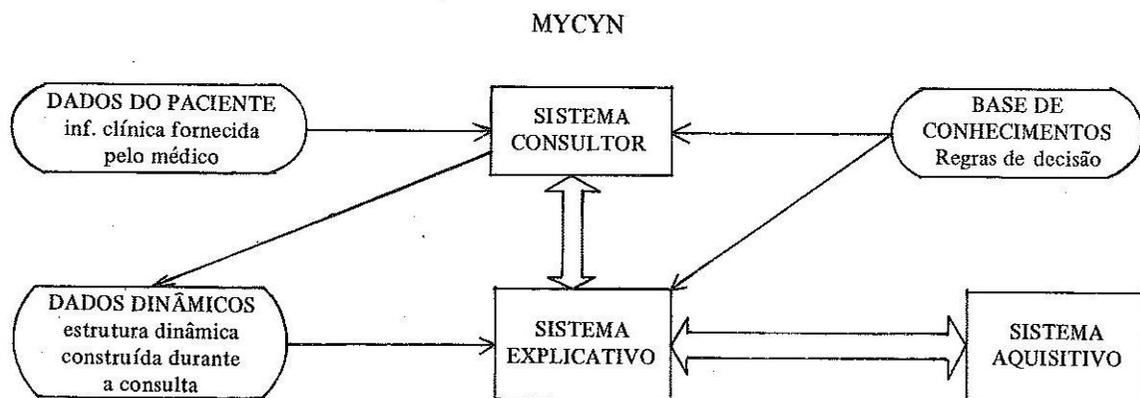


Figura 10

the new rule will be called RULE 200

```

if 1 the organism is a Gram Negative Rod
and 2 it is anaerobic
and 3 it was isolated from the blood
and 4 you think the Portal was the GI tract
and 5 <CR >
then 1 it is probably a Bacteroides.
one scale of 1 to 10, how much you affix to
this conclusion?

9

and 2 <CR >

```

Figura 11

depois da tradução em Lisp o sistema reenvia-a reformulada também em linguagem natural e pergunta ao utilizador se está de acordo.

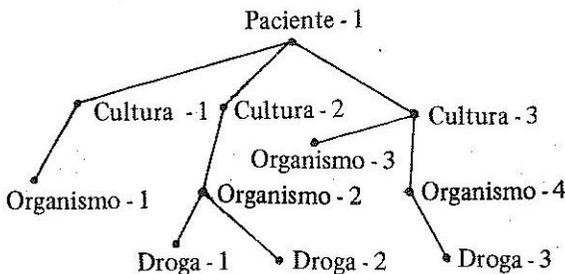
O mecanismo que o Mycin utiliza para compreender a linguagem natural evita a análise sintática ou semântica e faz uma aproximação em termos de palavras-chave.

Da mesma forma o sistema com o subprograma explicativo entende perguntas sobre o conhecimento corrente do paciente, sobre as inferências feitas, sobre as regras relevantes para certas questões, etc.

O mecanismo de controlo do Mycin é guiado pelos consequentes e isso foi apresentado como uma das boas características em «expert systems».

Este sistema pericial tenta portanto fazer inferências partindo, não dos dados, mas da negação de conclusões possíveis.

Outra das suas peculiaridades é a estrutura de dados que é uma árvore de contextos para representar o conhecimento adquirido dinamicamente sobre o paciente. As informações obtidas são sobre o paciente propriamente dito, sobre os organismos isolados, as drogas, e as culturas. Assim se definem outros tantos contextos que vão corresponder a níveis na árvore, p. ex.:



De notar que estes nós podem ou não ser criados e se o forem são-no dinamicamente à medida que a informação vai chegando.

Quanto às regras embora perfeitamente independentes de qualquer ligação numa rede de decisão estão no entanto categorizadas segundo o contexto para o qual são invocadas.

Cada nodo (Parâmetro clínico) da árvore tem um tipo de dados que é um triplo associativo < atributo, objecto, valor >.

Por exemplo o objecto paciente tem o atributo nome com o valor José.

É evidente que esta representação é a usada no cálculo relacional e como caso particular no cálculo de predicados:

(idade, Rui, 10)

(Febre, José, Sim) \equiv Febre (José)

Outro ponto de honra do sistema é o seu tratamento do conhecimento incerto.

Cada parâmetro clínico tem associado ao seu valor um determinado «factor de certeza» (FC) que reflecte a «crença» na correcção desse valor. FC entre 0 e 1 reflecte a evidência na hipótese. FC entre 0 e -1 reflecte a descrença. Estes factores que apareçam nas permissas das regras são depois tratados por funções especiais para fornecerem qual o peso que está associado à conclusão.

O teorema de Bayes que é normalmente usado para tal foi aqui evitado uma vez que ele implicaria o cálculo de numerosas probabilidades condicionadas num campo em que o conhecimento é quase sempre de uma inexactidão confrangedora.

Uma interessantíssima discussão dos melhores métodos para tratar o raciocínio inexacto do mundo real pode encontrar-se no cap. 4 de «Mycin» (Shortliffe, 1976) onde se discutem os teoremas de Bayes a teoria dos «fuzzy sets» de Zadeh, a teoria da confirmação (Canasp, Tornebohm e outros) entre outras.

ALGUMAS REFLEXÕES SOBRE A NOVA PROBLEMÁTICA DOS «EXPERT SYSTEM»

Há vários critérios de julgamento aplicáveis aos «expert systems». Critérios de eficiência, extensibilidade, validade psicológica, compreensibilidade, são alguns deles.

Certamente que haverá sistemas que respondem

melhor a uns critérios e sistemas que respondem melhor a outros. No entanto uma avaliação segura envolverá sempre duas questões fundamentais: a eficiência e a declaratividade.

Várias são também as vias que se tem aberto ao desenvolvimento de sistemas periciais mais perfeitos e mais capazes.

Quer no domínio da Arquitectura dos sistemas em que a questão do processo em paralelo (ou só a sua simulação) é sempre uma motivação presente, quer na adequação dos processos e formalismos de representação mais convenientes para tratar os conceitos em causa, quer ainda na questão da aquisição do conhecimento evolutivo muitas são as pistas ou as tentativas de melhoramentos.

Existem ainda sistemas periciais vocacionados apenas para a aquisição de novo conhecimento, tratando directamente com o perito, sem grande auxílio do informático. Mas essa automatização pode acelerar-se quer não só através de sistemas especiais de edição como pelos avanços nas técnicas de processamento da linguagem natural que permitam com uma maior «naturalidade» (e não em um vocabulário ainda bastante restrito) «ensinar» a base de conhecimentos, quer ainda por desenvolvimento de processos que permitam ao sistema a aprendizagem pela experiência.

Quanto à representação do conhecimento podem equacionar-se as hipóteses dos Sistemas de Produção «puros» (com bases de dados constituídas por factos isolados e base de conhecimentos de regras) as redes associativas como estrutura da BD, ou até um formalismo como as redes associativa estendidas (*extended semantic networks*), capazes de constituir só por si todo o suporte de «expert system». Ou, o cálculo de predicados e nomeadamente as linguagens com ele relacionadas.

Efectivamente um sistema tipo Dendral ou até em parte Mycin estão mais próximos de um sistema de Produção puro enquanto que um sistema como o Prospector estrutura a sua BD com redes associativas. Além disso, Robert Kowalsky propõe a utilização dos «extended semantic networks» que recebem da lógica o seu poder obtendo uma semântica, regras de inferência e uma interpretação procedimental, para além de terem a vantagem de que todas as relações que representam serem binárias e não n-árias.

Finalmente em que medida é que a utilização de uma linguagem de programação baseada na lógica (ou num subconjunto dela) como é o Prolog,

possuindo já incorporado um interpretador que é um Sistema de Produção CDS, (Nilsson, 1980) e onde em última análise todo o conhecimento, desde factos e regras, são cláusulas Horn (unitárias ou não), pode tornar obsoletos todos os esquemas rígidos de arquitectura até aqui apresentados? Um «expert system» pode ser apenas uma conjunto de cláusulas (regras ou dados) guiados por um interpretador que até podendo não ser o da linguagem Prolog (mas escrito nela) seria um meta-interpretador com a versatilidade de aplicar a melhor estratégia para os problemas em causa. É isso aliás o que se passa com Orbi.

Mas independentemente disso, a experiência existente dos PS mostrou obsoleta a noção inicial dos PS puros verificando-se cada vez mais algumas tendências que segundo D. Lenat *vão dos*:

- Princípio das estruturas de dados mais ou menos uniformes (uma ou duas estruturas de dados).
- Princípio do isolamento dos elementos das estruturas de dados (sem apontadores de uns elementos para os outros).
- Princípio das regras simples (p. ex.: preferem-se várias regras extra a uma regra com várias acções).
- Princípio de que todo o conhecimento substancial deve estar em regras.
- Princípio de um interpretador simples. (É essencial que todas as regras possam ser escolhidas em qualquer altura.)

para os

- Princípio das várias estruturas de dados conforme o mais conveniente para cada peça de conhecimento.
- Princípio da inter-relação entre os vários itens da Base de Dados.
- Princípio de que as regras também podem ser consideradas dados para serem guiados por outras regras (meta-regras).
- Princípio do conhecimento implícito (mui-

tas inferências como a implicação lógica, implicação causal, referem apenas relações entre os dados e como tal devem ser codificadas nas estruturas de dados, deixando às regras as inferências cuja conclusão é uma acção procedimental sobre o universo considerado).

Duas outras questões em aberto dizem respeito às melhores técnicas de escolha de activação de regras quando várias são possíveis de unificação com a «pattern» activa (*multiple matching*) e também à possibilidade de unificação quando várias cláusulas satisfazem apenas parcialmente a «pattern» activa nesse momento (*partial and best matching*).

CONCLUSÃO

Filósofos, lógicos, psicólogos, linguistas, informáticos podem eventualmente contribuir para o desenvolvimento de uma chamada Ciência da Cognição («cognitive science») ou Cogitologia (como propõem alguns soviéticos) mas neste momento não parece estar aí o fulcro da investigação e das realizações da IA.

Talvez não seja até muito coerente um domínio de estudos dos processos cognitivos animais e artificiais como não parece ter sido totalmente frutuoso o estudo conjunto dos mecanismos de controlo animais e artificiais na Cibernética.

Não cerceemos contudo a riqueza da discussão da problemática colocada pela IA. São problemas de tal ordem importante que alguns apelidam já a IA (provavelmente com algum exagero) de «Epistemologia Aplicada».

Mas o que parece realista concluir da actual direcção na investigação desta disciplina é a progressiva construção dos meios de desenvolvimento de um global sistema híbrido homem-máquina optimizado no sentido de ao homem permitir a libertação das suas potencialidades mais criadoras.

Tentamos portanto aqui sobrelevar o papel dos Sistemas de Produção descrevendo-lhes as suas principais características.

E se alguns (como Knuth) comentam com estranheza o facto de, se os sistemas de IA têm vindo a conseguir implementar muito daquilo que requer um certo tipo de raciocínio, falham rotundamente naquilo que o homem faz «sem pensar», passando

por cima da lógica, talvez isso não seja uma ligação mas, de certo modo, uma garantia ...

RESUMO

Na busca sobre o que é a Inteligência Artificial abordam-se as suas metodologias e os seus campos de aplicação. E, em particular apresentam-se os sistemas de produção através da sua classificação, estrutura, funcionamento, implementação e exemplos de aplicações. Finalmente, tecem-se algumas reflexões sobre a nova problemática dos sistemas periciais.

SUMMARY

Along the present paper it is achieved an inquiry upon Artificial Intelligence goals, by discussing its methodologies and application fields. In particular, we present the production systems through its classification, internal structure, working, implementation, and application examples. Finally, we reflect upon the problems of the expert systems.

REFERÊNCIAS

Não se reporta aqui a grande quantidade de bibliografia de IA usada mas a alguma da mais salientemente referida no artigo.

DAVIS, R. e KING, J. (1977) — «An overview of Production Systems», *Machine Intelligence* n.º 8.

FEIGENBAUM, E. (1979) — «Themes and case studies of Knowledge Engineering», in *Expert Systems in the microelectronic age*, ed. D. Michie.

NILSSON, N. (1980) — «Principles of Artificial Intelligence», *Tioga publishing*.

NILSSON, N. (1984) — «Artificial Intelligence: Engineering, science or slogan?», *SRI International*, Technical note 248, July 81.

SHORTLIFFE, E. (1976) — «Mycin: Computer-Based medical consultations», Elsevier, 1976.

RAUZINO, (1982) — «Conversations with an intelligent chaos», *Datamation*, Maio 82.

WATTERMAN, D. (1978) — *Pattern directed Inference Systems*, ed. Watterman & Hayer-Roth 1978.

WINSTON, P. (1977) — *Artificial Intelligence* — Addison Wesley 1979.