



A Incidência da Inteligência Artificial na Lógica

Luís Moniz Pereira
Universidade Nova de Lisboa

Disputatio No. 8

May 2001

DOI: 10.2478/disp-2001-0003

ISSN: 0873-626X

A INCIDÊNCIA DA INTELIGÊNCIA ARTIFICIAL NA LÓGICA¹

Luís Moniz Pereira

Universidade Nova de Lisboa

SUMÁRIO

A análise e estudo do impacto e incidência da Inteligência Artificial (I.A.) na Lógica é abordada através do exame dos requisitos e exigências, colocadas à Lógica, pela representação do conhecimento e outras questões abordadas pela I.A. Salientamos algumas das contribuições dadas pela I.A., através da Programação em Lógica, para formas mais dinâmicas de lógica, a fim de tratar o fluxo do conhecimento, nomeadamente: a informação incompleta e contraditória, a formulação e elaboração de hipóteses através da abdução, a argumentação, o diagnóstico a identificação e remoção de erros, a actualização de conhecimento, e a aprendizagem. Ao longo do percurso, aprofundamos as implicações e consequências que acarretam para a filosofia do conhecimento.

A nossa motivação consistiu, essencialmente, em induzir os leitores porventura mais atreitos aos aspectos filosóficos do conhecimento, a aventurarem-se por novos territórios recentemente demarcados no continente da Lógica em resultado da pesquisa contínua da I.A., ao mesmo tempo que lhes oferecemos uma visão de conjunto sobre essas novas regiões.

Os aspectos mais técnicos da nossa exposição têm como objectivo uma compreensão intuitiva, através de exemplos simples, das questões de

¹ Este artigo resulta da tradução, revista em detalhe pelo autor, do artigo original, em inglês, e publicado em 1999 na revista *Grazer Philosophische Studien*, vol. 56 pp. 183-204, com o título "The Logical Impingement of AI", e que pode ser encontrado na página Internet do autor, em <http://centria.di.fct.unl.pt/~lmp/>. O sentido do original prevalecerá sobre o da tradução.

No entretanto, o autor redigiu uma versão bem mais extensa do original em inglês, como capítulo convidado de um livro em honra de Robert Kowalski, a publicar pela Springer. Intitula-se "Philosophical Incidence of Logic Programming", e também consta da página Internet do autor.

natureza mais geral, do modo como são abordadas, resolvidas e ultrapassadas. Os tecnicismos mais elaborados e seus desenvolvimentos encontram-se nas referências².

1. INTRODUÇÃO

Abordamos o impacto e a incidência da Inteligência Artificial (I.A.) sobre a lógica através do exame dos requisitos e exigências, colocadas à Lógica, pela representação do conhecimento e outras questões abordadas pela I.A..

Salientamos seguidamente algumas das contribuições dadas pela I.A., através da Programação em Lógica, para formas mais dinâmicas de Lógica, a fim de tratar o fluxo do conhecimento, nomeadamente: a informação incompleta e contraditória; a formulação e elaboração de hipóteses através da abdução, a argumentação, o diagnóstico a identificação e remoção de erros, a actualização de conhecimento, e a aprendizagem. Ao longo do percurso, aprofundamos as implicações e consequências que acarretam para a filosofia do conhecimento.

A nossa motivação consiste, essencialmente, em despertar interesse nos leitores mais propensos a questões filosóficas, sobre áreas recentemente demarcadas no vasto “continente” da lógica, através das pesquisas contínuas em I.A. oferecendo uma visão de conjunto sobre o tema. As partes mais técnicas da presente exposição têm por objectivo a compreensão intuitiva, através de exemplos simples, das questões de natureza mais geral, do modo como são abordadas e resolvidas. Explicações mais detalhadas sobre os tecnicismos e seus desenvolvimentos podem ser encontradas nas referências e na página Internet do autor³.

A estrutura do presente trabalho é a seguinte: A próxima secção recapitula e dá ênfase ao papel singular da Lógica como conhecimento humano de representação e instrumento de raciocínio dialéctico. A secção após essa dá seguimento à anterior e salienta o benefício da mecanização da lógica nos computadores. Subsequentemente, descrevem-se as incursões da I.A. sobre a lógica, e a última secção constitui um enquadramento em que são realçadas as suas novas incidências para a filosofia, objecto aí de análise mais aprofundada. No seguimento são salientadas algumas das ferramentas de Programação em Lógica utilizadas para este efeito. Depois leva-se a cabo uma descrição geral da utilização destas ferramentas no diagnóstico de teorias lógicas. Na secção seguinte é delineado um enquadramento lógico-dinâmico, após o que se elabora sobre o mesmo descrevendo sumariamente o projecto “MENTAL – Mental Agents Architecture in Logic”, coordenado pelo autor. Fecha-se com *Comentários Finais* e lançando um desafio aos filósofos.

² E também na página Internet do autor já referida.

2. EVOLUÇÃO E ARGUMENTAÇÃO

A evolução proporcionou aos seres humanos um discernimento simbólico, bem como capacidades de comunicação abstracta.

O conhecimento objectivo comum requer que a mente acompanhe o pensamento abstracto, contemplando regras (independentes do conteúdo) de raciocínio dialéctico e de argumentação, as quais não devem ser inteiramente subjectivas, sob pena de tornarem a comunicação objectiva e o esforço de raciocínio colectivo praticamente impossíveis. Tais regras acabaram por se entranhar na essência do próprio pensamento humano, constituindo um enorme valor acrescentado conjunto para a sobrevivência³.

No quadro da evolução do conhecimento humano, tanto o conhecimento mimético (como por exemplo o conhecimento inerente à elaboração de mapas e modelos), como o conhecimento puramente imitativo (por exemplo o relacionado com os rituais de observação, ou com a reprodução de objectos), constituíram passos iniciais fundamentais na direcção do comportamento socialmente situado, e seguidor de regras conjuntas, exigido, por exemplo, pelo planeamento das caçadas. Subsequentemente, emergiu em todas as culturas e sociedades humanas o domínio do abstracto como resultado de jogos sociais baseados no seguir de regras.

As regras dos jogos encerram em si mesmas padrões definidores de situações, e sequências causais concretas do tipo situação-acção-situação, que espelham a obediência à causalidade da realidade física. A partir dos jogos, resultou mais abstracção, daí emergindo finalmente as noções de conceito definidor de situações, de regras gerais sobre o pensamento e seu encadeamento, bem com quais as jogadas de legítima argumentação e contra-argumentação. Conjuntamente, constituem aquilo a que se pode chamar de meta-jogo cognitivo⁴.

A ubiquidade da lógica no que diz respeito à representação do conhecimento e do raciocínio, sendo uma verdadeira *lingua franca* em todas as línguas e culturas humanas, reside na sua capacidade real de encorajar e promover a compreensão racional e a objectividade prática. Decisivamente, a própria evolução dinâmica do conhecimento objectivo, quer individual quer colectivo, segue leis e padrões de raciocínio.

Além disso, e mais recentemente, as próprias regras de raciocínio podem e são efectivamente empregadas no discernimento sobre o próprio raciocínio. Apesar de alguns métodos de raciocínio serem sobejamente conhecidos, contudo alguns continuam sendo-o apenas inconscientemente. Porém, tal

³ Ver por exemplo: Terrence Deacon, *The Symbolic Species*, W.W. Norton, 1997; Merlin Donald, *Origins of the Human Mind*, Harvard University Press, 1991; Steven Pinker, *How the Mind Works*, W.W.Norton, 1997.

⁴ Cf. John Holland, *Emergence*, Addison-Wesley, 1998.

como acontece com as regras da gramática, eles podem vir a ser descobertos pela investigação. Por outro lado, novos métodos de raciocínio vieram a ser inventados e aperfeiçoados ao longo da história civilizacional da evolução humana. Exemplos disto são: a indução transfinita, a *reductio ad absurdum* (prova por contradição), a recursão, a abdução, e a remoção da contradição, para apenas citar alguns tipos.

Novos e supostos métodos genéricos de raciocínio podem ser alvo de controvérsia, tal como o poderão ser quaisquer passos ou esquemas concretos de raciocínio ou dedução. Porém, o raciocínio apenas pode ser alvo de disputa através de raciocinação adicional sobre ele, se se pretender alcançar um mínimo de consenso objectivo!⁵ Algumas pessoas argumentam que a discussão científica e filosófica se circunscreve necessariamente a uma argumentação de natureza cultural *ad hoc*, informal, persuasiva, culturalmente sensível e relativa, aliada à retórica do vale tudo⁶. Muitas delas ignoram que a argumentação constitui, também ela, uma forma de raciocínio que se tornou objecto próprio da formalização lógica, e esquecem que se a retórica pode ser óptima para pregadores, não oferece a comunicação bilateral indispensável para se poder atingir o consenso requerido pela rigorosa *praxis* científica, e conducente à acumulação do conhecimento.

3. A LÓGICA E O COMPUTADOR

A lógica, como acabamos de ver, nasceu como sendo a formulação, independente do conteúdo, das leis de pensamento. As suas predicções expressam quaisquer relações conceptuais que possamos desejar sobre os mundos interior e exterior. Estes blocos de construção predicativos podem então ser combinados em fórmulas, por intermédio de quantificadores e conectivos lógicos, manipuláveis de acordo com as regras de inferência caracterizadoras das variadas formas de raciocínio.

A lógica tem a capacidade de articular a visão exterior e comportamentalista das predicções, encaradas como caixas negras ou relações de entrada/saída, concomitantemente com uma visão intencional dos predicados, encarados agora como funcionalmente decomponíveis em outros predicados, como caixas negras interligadas no interior de outras caixas negras⁷.

⁵ Cf. Thomas Nagel, *The Last Word*, Oxford University Press, 1997. (Trad. port: Gradiva.)

⁶ Cf. Paul R. Gross, Norman Levitt, *Higher Superstition*, The Johns Hopkins University Press, 1994.

⁷ Cf. Luís Monteiro, Luís Moniz Pereira, "Aspectos Cibernéticos da Epistemologia", in *Novas Perspectivas das Ciências do Homem*, pp. 13-27, Biblioteca de Ciências Humanas 4, Editorial Presença, Lisboa, 1970.

Uma vez que a linguagem da lógica é puramente simbólica, e as suas regras operativas são independentes do conteúdo semântico, as elaborações lógicas podem ser especificadas através de procedimentos gerais e abstractos. Estes permitem-se, por sua vez, ser alvo de programação em computador. Via a sua mecanização, as teorias lógicas e as diversas formas de raciocínio lógico-dedutivo atingem hoje, pela primeira vez, uma existência *in vitro*, e uma capacidade e disponibilidade para a execução repetível, independente de apoio de *hardware* mental humano⁸.

Os dois pontos de vista atrás mencionados, intencional e extensional, são conciliados no computador através da insistência em que a semântica declarativa do “saber”, ou do “que” é especificado, seja equivalente à semântica do procedimento computacional do “fazer”, ou do “como” a especificação vai ser derivada. Esta equivalência semântica entre especificação e derivação constitui a verdadeira pedra basilar do uso do paradigma da Programação em Lógica na Ciência da Computação, ao qual voltaremos⁹.

4. A FILOSOFIA E A INTELIGÊNCIA ARTIFICIAL

A Filosofia tem sido berço e suporte do desenvolvimento do pensamento racional ao longo da história intelectual do ser humano. O seu estudo da linguagem nutriu e animou o refinamento da lógica como abstracção da linguagem natural, e como ferramenta de questionamento linguístico. As investigações sobre os fundamentos da matemática atribuíram igualmente à lógica o papel meta-instrumental que desencadearia a sua disposição no terreno como ferramenta genérica de manipulação de símbolos *par excellence*. A ciência da computação adoptou também a lógica como seu instrumento fundacional intrínseco, enquanto que a I.A. tornou viável e credível a proposta de a tornar numa linguagem de programação *bona fide*.

Simultaneamente, a I.A. desenvolveu a lógica muito para além dos limites da cumulatividade monotónica, característica dos domínios precisos, completos, duráveis, condensados e fechados da matemática, abrindo-os ao domínio não monótono do mundo real, do conhecimento impreciso, incompleto, contraditório, discutível, susceptível de revisão e de evolução. Em suma, a I.A. acrescentou dinâmica à estática reinante.

⁸ Cf. Luís Moniz Pereira, “Inteligência Artificial: mito e ciência,” in *Revista Colóquio-Ciências*, 3:1-13, Outubro 1988, Fundação Calouste Gulbenkian, Lisboa. Este artigo está disponível na revista electrónica filosófica *Intelecto*, n.º 5, no URL <http://www.geocities.com/revistaintelecto/>, juntamente com uma entrevista recente ao autor, “Conversando sobre Inteligência Artificial.”

⁹ A Ciência e a Tecnologia doravante, apoiadas pela Informática e especialmente através da Inteligência Artificial, verão cada vez mais automatizados, de forma simbiótica com o Homem, os processos da sua própria criação, com retroacção positiva!

Na I.A., como em qualquer ramo da ciência, a lógica comum tem um importante papel a desempenhar. Sem dúvida que qualquer ciência, implícita ou explicitamente, de uma maneira ou de outra, arregimenta a lógica e o raciocínio ambos ao seu serviço, para construir argumentos, contra-argumentos, e para resolver disputas. Contudo, pode esperar-se da I.A. um papel bem mais amplo na lógica!

- Em primeiro lugar, a I.A. pretende mecanizar a lógica, uma ferramenta absolutamente essencial para tantas actividades racionais;
- Em segundo lugar, a I.A. intende tornar explícita e bem definida a lógica sub-consciente que utilizamos, testando-a objectivamente através da sua automação por via informática;
- Em terceiro lugar, a I.A. emprega a lógica como uma linguagem de genérica de comunicação, de conhecimento e de procedimentos, entre seres humanos e computadores, bem como entre os próprios computadores;
- Em quarto lugar, em I.A., mesmo quando procedimentos e artefactos não são implementados utilizando directamente a lógica, esta pode assumir o papel de linguagem precisa de especificação dos requisitos daqueles, bem como o de um formalismo através do qual estudar as respectivas propriedades semânticas;
- Em quinto lugar, a I.A. contribuiu significativamente para o equacionamento e exame do problema da identificação dos limites do raciocínio simbólico próprio corporizado em computador, bem como de avaliação sobre se estes limites se aplicam aos seres humanos;
- Finalmente e em sexto lugar, a I.A. auxiliou os investigadores na exploração de novas questões e de novos métodos de raciocínio, bem como a combinar modalidades díspares de raciocínio num enquadramento uniforme e unificado, de modo a lidar com informação incompleta, imprecisa, contraditória e em mudança.

Este último papel da I.A., em contraste com o papel predecessor, muito apapricado pelos filósofos, quase não tem chamado a atenção deles por enquanto. A epigénese recente da lógica por ele anunciada, e o seu sustento computacional, apontam para um papel inovador do empreendimento logístico. A saber, o papel de suporte, justificação e mecanização da dinâmica dos processos epistemológicos e argumentativos abstractos, subjacentes ao próprio esforço científico. E isto constitui, em nossa opinião, boas notícias para os filósofos da ciência.

Consequentemente, é para esse aspecto que agora nos viramos, na esperança de que existirão filósofos que auxiliarão no processo de amadurecimento de uma epistemologia computacional de corpo inteiro.

5. FERRAMENTAS LÓGICAS DA INTELIGÊNCIA ARTIFICIAL

A lógica clássica foi desenvolvida de modo a estudar objectos matemáticos imutáveis, bem definidos e consistentes. Adquiriu, por esse motivo, um carácter estático. A I.A. necessita de tratar com o conhecimento em fluxo, em condições longe da perfeição, por meio de formas mais dinâmicas de lógica. Estes aspectos têm constituído o cerne da pesquisa em Programação em Lógica, um ramo da I.A. que utiliza a lógica directamente como linguagem de programação¹⁰, fornecendo para esse efeito métodos específicos de implementação e sistemas funcionais eficientes¹¹. A Programação em Lógica é igualmente bastante utilizada como veículo de base para a implementação sobre ele de outras formas de abordagem da I.A. à lógica.

Quais são as questões e requisitos da dinâmica da lógica que a I.A., nomeadamente através da Programação em Lógica, tem contribuído para resolver?¹²

5.1 A Programação em Lógica

Existe uma ligação estreita entre a implicação lógica e a causalidade física. A lógica, afinal, permite-nos elaborar modelos sobre os acontecimentos no mundo. Isto verifica-se de um modo mais premente no caso da Programação em Lógica (PL), em que todas as conclusões verdadeiras devem ser devidamente suportadas em, ou "causadas" por, premissas, e em que a implicação é unidireccional, ou seja, não contraponível: as "causas" não têm efeitos nos seus antecedentes. A notação com cláusulas ou regras Horn é aqui utilizada para expressar esta direccionalidade.

Para produzir um resultado ou conclusão C , é necessário verificar-se a conjunção de suficientes condições positivas P que a apoiem, conjuntada por "e" com a ausência ou negação de uma conjunção $\neg N$ de todas as condições negativas que a evitariam, dado P :

$$C \leftarrow P, \neg N$$

¹⁰ Cf. Robert Kowalski, *Logic for Problem Solving*, North-Holland, 1979, para os fundamentos, e José Júlio Alferes, Luís Moniz Pereira, "Reasoning with Logic Programming", *Lecture Notes in Artificial Intelligence*, vol. 1111, Springer, 1996, para um estado da arte recente.

¹¹ Para o mais avançado, que incorpora os mais recentes desenvolvimentos teóricos, veja-se o sistema XSB em: <http://www.cs.sunysb.edu/~sbprolog/xsb-page.html>.

¹² Para um tratamento técnico detalhado dos assuntos que se seguem consulte-se o nosso livro, indicado na nota de pé de página número 10, e as referências que nele constam, bem como a página Internet do autor, em <http://centria.di.fct.unl.pt/~lmp/>, para os trabalhos mais recentes.

Caso exista mais de uma maneira de obter C , temos então diversas fórmulas do tipo:

$$C \leftarrow P_i \neg N_i$$

Se as nossas informações sobre cada fórmula, bem como sobre o conjunto das mesmas para C , forem completas, teremos então:

$$C \leftrightarrow \bigvee_i P_i \neg N_i$$

Mas que acontecerá se tal não for assim? Primeiro, talvez não possuamos suficiente informação sobre cada $\neg N_i$. Poderemos conceberivelmente conhecer cada P_i , pelo menos sem um dos quais C jamais poderá ser obtido. Porém, no que diz respeito a $\neg N_i$, ou seja, todas as condições que, caso presentes, tornariam impossível a dedução de C em presença de cada P_i , isso seria mais difícil.

5.2 *Mundos Abertos*

Demasiadas coisas poderão correr mal num mundo aberto, não matemático, de algumas das quais nós nem sequer suspeitamos. Tanto quanto sabemos, poderia até rebentar uma bomba que destruísse toda a configuração física que estamos a procurar modelar de uma maneira lógica (a distância segura). Teremos também de modelar isso? No mundo real, qualquer conjuntura é suficientemente complexa para que dificilmente a possamos definir exaustivamente em cada ocasião. Teremos de levar em consideração a ocorrência de excepções imprevistas, baseadas em informação nova que adquiramos. Deste modo, em vez de termos de assegurar, ou provar, que alguma condição N_i não esteja presente, poderemos assumir que não está efectivamente, na condição de que estejamos preparados para aceitar informação que possamos receber susequentemente em sentido contrário. Quer dizer, poderemos assumir uma regra mais geral do que o justificado, porém deveremos estar a partir de então preparados para tratar as excepções que possam depois surgir.

Tomemos por exemplo, com a leitura óbvia, o pedaço de conhecimento expresso pela seguinte regra Horn, i.e., predicacões positivas e negativas conjuntadas, que implicam apenas uma predicacão positiva:

$$\text{fiel (H,K)} \leftarrow \text{casado (H,K), } \neg \text{ amante (H,L)}$$

Não possuímos, normalmente, suficiente informação explícita sobre quem é, ou não, amante de uma determinada pessoa, apesar de que tal espécie de informação nos possa chegar totalmente inesperadamente, especialmente quando se trata de Presidentes¹³. Conseqüentemente, poderemos codificar o nosso conhecimento na presunção de que se alguém denominado H estiver casado com K , e não existir informação ou base para concluir que existe L tal que *amante* (H,L), então temos que H é fiel a K :

fiel (H,K) \leftarrow casado (H,K), not amante (H,L)

Isto é expressável através da chamada negação supletiva aplicada a uma proposição P , *not* P , a qual poderá ser lida como “ P não é demonstrável”. Ou seja, já não necessitamos de provar \neg *amante* (H,L) qualquer que seja L dado H , mas assumiremos tal a menos que possamos estabelecer *amante* (H,L) para algum valor L dado um H . Por outras palavras, se não possuímos evidência que nos leve a concluir *amante* (H,L) para um L dado H , assumiremos tal falso para todos os valores L dado H .

5.3 Hipóteses Anuláveis

A negação supletiva (também dita por omissão, ou por defeito) foi introduzida pelos investigadores de I.A. através da Programação em Lógica, e poderá ser lida alternativamente como “ P não é demonstrável”, ou “a falsidade de P pode ser assumida”, ou “a falsidade de P pode ser abduzida”, ou “não existe evidência para P ”, ou “não existe argumento para P ”. A negação supletiva permite-nos lidar com a falta de informação, uma situação muito comum no mundo real. Introduce a não-monotonicidade na representação do conhecimento porque se, mais tarde, tivermos conhecimento da existência de relacionamentos amorosos, entre um L e H , teremos que retirar a nossa conclusão anterior acerca da fidelidade que tenhamos presumido como existente. Sem dúvida que as conclusões podem não se revelar sólidas em consequência da anulabilidade de hipóteses nas regras que lhe estiveram subjacentes. Textos de natureza jurídica, regulamentos, e tribunais empregam esta forma de negação supletiva abundantemente, uma vez que forçosamente lidam com mundos abertos, apesar de ela não ter sido previamente formalizada pela lógica, até a I.A. ter descoberto igualmente a necessidade de a utilizar.

¹³ Na ocasião em que escrevia, em Fevereiro de 1998, estava nos EUA e o Presidente americano era então pressionado a propósito de um suposto caso amoroso na sala oval.

5.4 Hipótese do Mundo Fechado

Assinale-se que *not* deve outorgar igual estatuto à informação positiva e à negativa. Ou seja, deveremos então poder escrever a seguinte:

$\neg \text{fiel}(H,K) \leftarrow \text{casado}(H,K), \text{not } \neg \text{amante}(H,L)$

para caracterizar um mundo em que as pessoas são infiéis por sistema ou costume, sendo portanto explicitamente necessário provar (ao invés) que alguém *H* não tem nenhum amante *L* antes de poder concluir que *H* não é infiel. Caso contrário presume-se *H* infiel por negação supletiva. Esta questão coloca-se porque, frequentemente, e em especial no caso de base de dados ou bases de conhecimento, se torna necessário impôr a hipótese do mundo fechado (Closed World Assumption ou CWA). Ou seja, a hipótese ou assumpção de mundo fechado é feita quando se presume que a base de dados ou de conhecimento possui toda a informação positiva pertinente, de modo que apenas é verdadeiro aquilo que está explicitamente declarado, ou é explicitamente daí derivável; caso contrário presume-se que seja falso. Porém, e de modo geral, como será possível possuir toda a informação pertinente num mundo em constante mutação?

A informação é expressa normalmente de um modo positivo, por força de mecanismos economizadores, de natureza mental ou linguística. Por isso, através da CWA, a informação ausente, não explicitamente adquirível, é comumente a negação de informação positiva. O que significa que, sempre que não exista qualquer informação disponível sobre amantes, $\neg \text{amante}(H,L)$ é verdadeiro por CWA, ao passo que *amante* (*H,L*) não o é. Esta assimetria é indesejável, uma vez que os predicados escolhidos são puramente convencionais e poderíamos, justificadamente, colocar a hipótese de empregar o predicado positivo *não-amante* (*H,L*) e a sua negação $\neg \text{não-amante}(H,L)$, para modelar o nosso conhecimento. A linguagem lógica deveria ser indiferente no que diz respeito a estes acidentes de representação do conhecimento.

5.5 Negação Explícita

Por outro lado, quando não possuímos informação factual ou qualquer outra informação dedutível, de natureza positiva ou negativa, gostaríamos de poder afirmar que ambas os casos são falsos epistemologicamente, isto é do ponto de vista do “conhecimento que possuímos”. Consequentemente, o princípio do terceiro excluído é inaceitável devido à possibilidade de certas predicções e suas negações poderem ser simultaneamente falsas.

A CWA e os requisitos de simetria e epistemológicos podem ser reconciliados através da leitura do símbolo “ \neg ” acima referido, não como sendo uma

negação do tipo clássico, a qual obedece ao princípio do terceiro excluído segundo o qual qualquer predicação ou é verdadeira ou falsa, mas como sendo uma ainda outra nova forma de negação, apelidada em Programação em Lógica como “negação explícita”¹⁴ (a qual ignora a provisão do terceiro excluído). Podemos impôr a CWA apenas aos predicados P ou $\neg Q$ que desejarmos, escrevendo simplesmente:

$$\neg P \leftarrow \text{not } P \quad \text{ou} \quad Q \leftarrow \text{not } \neg Q$$

Alternativamente, a utilização de uma assumpção $\text{not } P$ ou $\text{not } \neg Q$ pode ser feita apenas para aquelas ocorrências predicativas que a ela necessitem de recorrer, tal como fizéramos anteriormente.

5.6 Revisão de Hipóteses

Examinemos de seguida a necessidade de revisão de hipóteses e de introdução no nosso esquema de raciocínio de um terceiro valor de verdade, a que chamaremos de “indefinido”.

Surge alguma confusão quando combinamos os pontos de vista de ambos os mundos atrás referidos:

$$\begin{aligned} \text{fiel}(H,K) &\leftarrow \text{casado}(H,K), \text{not } \neg \text{amante}(H,L) \\ \neg \text{fiel}(H,K) &\leftarrow \text{casado}(H,K), \text{not } \text{amante}(H,L) \end{aligned}$$

Assumindo que $\text{casado}(H,K)$ para certos H e K , aparentemente tanto $\text{fiel}(H,K)$ como $\neg \text{fiel}(H,K)$ são contraditoriamente verdadeiros. Já que não temos evidência para $\text{amante}(H,L)$ nem para $\neg \text{amante}(H,L)$, uma vez que simplesmente não temos qualquer informação sobre tal, fazemos duas hipóteses, uma acerca da falsidade de cada. Porém, quando uma qualquer hipótese conduz a uma contradição deveremos retirá-la. Trata-se do venerável princípio de *reductio ad absurdum*, ou seja de “raciocínio por contradição”. No nosso caso, duas hipóteses conduziram-nos a uma contradição. Qual delas deveremos retirar? Estão em pé de igualdade!

5.7 Indefinição

Se não possuímos qualquer outra informação preferencial, retiramos ambas uma vez que não estamos em condições de decidir entre delas. Ou seja, assumimos que nem $\text{amante}(H,L)$ nem $\neg \text{amante}(H,L)$, são falsas.

¹⁴ Cf. José Júlio Alferes, Luís Moniz Pereira, Teodor Przymusiński, “‘Classical’ Negation in Nonmonotonic Reasoning and Logic Programming”, *Journal of Automated Reasoning*, 20:107-142, 1998.

Uma vez que nenhuma delas é demonstravelmente verdadeira, classificamo-las de *indefinidas*. Quer dizer, introduzimos um terceiro valor de verdade de modo a melhor caracterizar a falta de informação mais precisa sobre uma determinada situação amorosa, tornando assim também *fiel* (H,K) e \neg *fiel* (H,K) igualmente indefinidos. Esta imposição de indefinição poderá ser alcançada simplesmente acrescentando ao nosso conhecimento:

\neg amante (H,L) \leftarrow not amante (H,L)
amante (H,L) \leftarrow not \neg amante (H,L)

Não possuindo mais nenhuma informação, não podemos provar quer a condição de *amante* (H,L) quer a de \neg *amante* (H,L) nem verdadeiras nem falsas. Qualquer tentativa de o fazer cai num círculo de auto-referência sem saída envolvendo a negação supletiva. Contudo, se por hipótese considerarmos uma delas verdadeira a outra torna-se falsa em consequência, e vice-versa. Ambas estas situações possíveis não são, portanto, postas fora de hipótese. Mas a terceira opção, sendo mais segura e céptica, será não tomar partido de qualquer delas nesta disputa marital, abstendo-nos de acreditar em qualquer das possibilidades.

De facto, a semântica bem fundada (Well Founded Semantics ou WFS) dos programas lógicos atribui aos literais referidos das duas cláusulas acima o valor de verdade *indefinido*, no seu modelo bem fundado, céptico em relação ao conhecimento, mas permite manter em aberto os outros dois modelos mais crédulos, não minimizadores de conhecimento.

Porque não poderemos adicionar simplesmente:

amante (H,L) \vee \neg amante (H,L) ?

Porque não funcionaria. Continuaríamos a não poder provar se *amante* (H,L) ou se \neg *amante* (H,L) eram definitivamente verdadeiras, mantendo-se a contradição!

Sempre que lidamos com não demonstrabilidade, necessitamos realmente de um terceiro valor de verdade, de modo a expressar a nossa incapacidade epistemológica de apresentar mais informação. Mesmo partindo do princípio de que o mundo é ontologicamente bipolar, o nosso acesso à informação codificada acerca dele poderá sempre ser tripolar em geral. Além do mais, o mundo pode muito bem não ser totalmente bipolar ou, em todo caso, não "capturável" de um modo inteiramente bipolar. Podemos alguma vez afirmar, acerca de qualquer partícula/onda, que está neste sítio ou que não está neste sítio? Estará realmente aqui ou então não estará aqui, apesar de não o podermos afirmar? Esta discussão a esse propósito não pode aparentemente ser resolvida experimentalmente.

De qualquer modo, necessitamos de um terceiro valor lógico ainda por outras razões. À medida que construímos a nossa base de informações e conhecimentos imperfeita sobre o mundo real, podemos muito bem criar, não deliberadamente e inconscientemente, dependências circulares tal como a atrás referida. Por exemplo, o Legislador bem poderá elaborar e implementar leis conflituantes e logicamente circulares. Ainda assim, envidamos esforços no sentido de prosseguir nos nossos raciocínios, quer essas circularidades exprimam legitimamente o que se pretende modelar quer não. E, sempre que não o façam, pretendemos detectar e funcionar com elas, ao invés de deitar for a criança juntamente com a água do banho!

Mas a indefinição pode também ser legitimamente posta ao serviço para exprimir conhecimento comum nas suas diversas leituras, crédulas e cépticas. Ora, ainda ontem li num livro¹⁵ aquele conhecido provérbio marinho que diz «*Resiste às tempestades que não podes evitar e evita as tempestades a que não consigas resistir!*», ou, traduzido de uma maneira lógica:

resistir ← tempestade, not evitar
evitar ← tempestade, not resistir

O que acontecerá a um marinho experiente durante uma tempestade? Resistir-lhe-á ou evitá-la-á? Existem três soluções possíveis para esta questão. Incluindo aquela em que existe indecisão sobre o resultado final. Para nós, significa que teremos de resistir ao conhecimento circular, não o evitando nas águas calmas dos paraísos artificiais da matemática...

5.8 Expressividade da Programação em Lógica

A introdução desta semântica a três valores na Programação em Lógica, constitui uma importante inovação da I.A., com diversas consequências. Nomeadamente, e temo-lo visto, permite-nos ser cépticos e não presumir mais de o que é sustentável.

Outra inovação, recapitulando, é que a negação \neg utilizada acima, denominada negação explícita, tem também três valores, pelo que difere da negação clássica. A negação explícita, como vimos, não se conforma ao princípio do terceiro excluído. As predicções e suas negações podem ser ambas falsas («*a cadeira está furiosa*», «*a cadeira não está furiosa*»). Numa base de conhecimentos, poderemos definitivamente não ser capazes de provar uma determinada afirmação ou a sua negação (explícita), pelo que ambas se tornam falsas supletivamente.

¹⁵ Cf. John Barth, *On with the Story*, p. 55, Back Bay Books, Little Brown and Company, 1996.

Adicionalmente, a introdução da negação explícita e, em especial, a sua combinação com a negação supletiva, proporciona uma representação mais expressiva do conhecimento. Consideremos, a título de exemplo, a seguinte recomendação

\neg atravessar \leftarrow comboio

versus a recomendação alternativa

\neg atravessar \leftarrow not \neg comboio

A última constitui, claramente, uma opção mais segura.

De acordo com ela, temos realmente de provar que não vem um comboio antes que \neg *atravessar* se possa tornar falsa, ao passo que com a primeira opção, \neg *atravessar* torna-se falsa bastando para isso não conseguirmos provar que um comboio vem aí!

Outra inovação a referir ainda é que a seta de implicação (\leftarrow), quando combinada com a negação supletiva ou com a negação explícita, não significa implicação material. As contrapositivas ou antíteses não são pois sancionadas. Consideremos o seguinte:

$A \leftarrow B$
 $\neg A$

$\neg B$ não se segue, pois não existe presunção da antítese $\neg B \leftarrow \neg A$. Sempre que as antíteses sejam desejáveis, devem ser explicitamente adicionadas. Pelo contrário, \leftarrow deverá ser encarado como exprimindo uma regra de inferência, que pode ser utilizada procedimentalmente tanto “de baixo para cima” de modo a concluirmos A dado B , ou então “de cima para baixo”, na tentativa de provar o corpo B de modo a provar a cabeça A . Os factos são apenas procedimentos com corpo vazio. Esta leitura inferencial das cláusulas como regras, faz de \leftarrow um operador direccional, o que o torna bastante apropriado para modelar não apenas a dedução mas também a causalidade.

Efectivamente, as semânticas dos programas lógicos enfatizam este aspecto ao insistir na admissão de apenas modelos de mínimos. Os modelos mínimos são aqueles para os quais qualquer literal positivo ou explicitamente negado é suportado por alguma regra cuja cabeça ou conclusão é o literal, e cujo corpo ou conjunto de premissas são, por sua vez, suportadas. Assim, os factos são automaticamente suportados uma vez que possuem um corpo vazio.

Adicionalmente, qualquer literal positivo ou explicitamente negado é *falso* somente se todas as regras para o mesmo tiverem corpo falso. Consequen-

temente, se não existirem quaisquer regras relativamente a um literal, ele é automaticamente falso.

Todos os outros literais positivos ou explicitamente negados são considerados *indefinidos* (isto poderá acontecer apenas se eles estiverem envolvidos em círculos auto-referenciais não resolvidos, através da negação supletiva). Finalmente, um literal P negado supletivamente, $\text{not } P$, é *verdadeiro/falso/indefinido* exactamente no caso de P ser, respectivamente, *falso/verdadeiro/indefinido*.

5.9 Actualização

Por último, mas de não menos importância, será de referir que a Programação em Lógica se tem debruçado sobre a actualização de uma base de conhecimentos por uma outra. Esta noção de actualização de conhecimentos, em oposição à simples actualização de factos, abre uma nova dimensão à dinâmica da lógica, contrastando com a estática da velha lógica. Dada uma base de conhecimentos existente, contendo factos e regras, bem como alguns novos contendo igualmente factos e regras, porventura contraditando até conhecimentos adquiridos anteriores, qual será a base de conhecimentos resultante da actualização de uma por outra? Poderá esse processo ser repetido?

Trabalhos recentes mostraram-nos como tal pode ser atingido com generalidade¹⁶. Algumas áreas de aplicação são, por exemplo, quando as duas bases de conhecimentos consistem em peças de legislação, ou regulamentações, normas de segurança ou regras de conduta de robôs.

6. DIAGNÓSTICO DE TEORIAS LÓGICAS

Existem muitos exemplos da utilização de negação supletiva, negação explícita, remoção de contradições e de valuações tripartidas, tornados possíveis pela pesquisa em Programação em Lógica, nomeadamente nas áreas relacionadas da abdução, argumentação, revisão de crenças, actualização de conhecimentos, aprendizagem e diagnóstico¹⁷. Ilustraremos seguidamente as últimas, ou seja, o diagnóstico ou depuração de uma base de

¹⁶ Cf. José Júlio Alferes, João Leite, Luís Moniz Pereira, Halina Przymusinska, Teodor Przymusinski, "Dynamic Logic Programming", *Int. Conf. on Knowledge Representation and Reasoning (KR'98)*, Trento, Italy. Stuart Shapiro (ed.), Morgan Kaufmann, 1998. Versão ampliada: J. J. Alferes, J. A. Leite, L. M. Pereira, H. Przymusinska and T. C. Przymusinski, "Dynamic Updates of Non-Monotonic Knowledge Bases", *Journal of Logic Programming*, vol. 45/1-3, pp 43-70, September/October 2000.

¹⁷ Cf José Júlio Alferes, Luís Moniz Pereira, "Reasoning with Logic Programming", *Lecture Notes in Artificial Intelligence*, vol. 1111, Springer, 1996.

conhecimentos genérica. As suas ligações com as outras áreas mencionadas permitem-nos também aflorá-las.

Suponhamos que temos as seguintes regras, expressando um programa lógico que nos permite computar o predicado C na base dos predicados P e N :

$$C(X) \leftarrow P(X), \neg N(X)$$

$$P(a) \quad \neg N(a)$$

$$P(b)$$

$$P(c) \quad \neg N(c)$$

Preveamos a possibilidade de a regra para C estar possivelmente errada, ou seja, que podem existir excepções à mesma, através da seguinte expressão:

$$C(X) \leftarrow P(X), \neg N(X), \text{not excepção}(C(X))$$

Se nada mais for dito sobre o predicado reservado *excepção*(_), as conclusões do programa lógico permanecem inalteradas, pelo que o programa poderá conter tais regras desde o início, para qualquer predicado que desejemos.

Se a seguir soubermos que $C(a)$ é falso, poderemos salvaguardar a regra para C , uma vez que deixa de ser válida para todo $X=a$, bastando para isso acrescentar ao nosso conhecimento o seguinte:

$$\text{excepção}(C(a))$$

A regra para C continuará a funcionar para os restantes casos, apesar da possibilidade de se poderem subsequentemente acumular mais excepções.

Este método toma conta das regras que não sejam fiáveis, ou seja, as que produzam resultados errados.

É o que acontece ao outro caso problemático, aquele que envolve a incompletude, ou seja, em que resultados considerados correctos não são no entanto produzidos? Suponhamos que ficávamos a saber que $C(b)$ é verdade, o que as nossas regras não produzem de todo e está em falta. Bem, é suficiente introduzir, para o conjunto das regras respeitantes a cada predicado, uma regra mais geral e abrangente. No presente caso:

$$C(X) \leftarrow \text{em_falta}(C(X))$$

$$P(X) \leftarrow \text{em_falta}(P(X))$$

$$\neg N(X) \leftarrow \text{em_falta}(\neg N(X))$$

Caso nada mais seja dito sobre o predicado reservado *em_falta*(_), as conclusões do programa lógico permanecem inalteradas, de modo que o programa pode conter tais regras desde o início. Agora, para fazer com que $C(b)$ seja verdadeiro, basta abduzir, ou adoptar, uma das duas hipóteses: que *em_falta* ($C(b)$) ou que *em_falta* ($\neg N(b)$) seja verdadeiro.

Este método toma conta dos conjuntos de regras que sejam incompletos, ou seja, não produzem todos os resultados.

Atingimos a generalidade. Através de considerarmos os dois predicados, *excepção*(_) e *em_falta*(_) abduzíveis, podemos diagnosticar e depurar uma base de conhecimentos expressa em lógica.

A abdução é um processo muito conhecido de raciocínio, através do qual adoptamos hipóteses visando que certas instâncias de predicados (ou as suas negações) sejam verdadeiras, a fim de provarmos uma conclusão pretendida. Trata-se de raciocinar a partir dos objectivos para as necessidades, e tem sido exhaustivamente estudada no contexto da Programação em Lógica¹⁸. Claro que as afirmações ou hipóteses adoptadas deste modo podem vir mais tarde a provar-se estarem erradas, se e quando conduzam a contradições. Teremos então de prever a aplicação de um processo de remoção de contradições baseado na revisão de hipóteses, adoptando possivelmente então outras hipóteses. Este aspecto foi também extensivamente estudado em I.A. e em Programação em Lógica, tendo tido como resultado sistemas de raciocínio automatizado que fazem para nós esse trabalho¹⁹.

7. UM ENQUADRAMENTO LÓGICO DINÂMICO

Não é muito difícil imaginar como poderá ser utilizado um processo combinado de geração de regras, de diagnóstico sistemático, e ainda de revisão de regras por actualizações, para alcançar-se uma aprendizagem automatizada de teorias, de um modo integrado, adentro do enquadramento uniforme da Programação em Lógica.

Para se iniciar a aprendizagem, começa-se com algum tipo de conhecimento prévio e preceitos fixos ou já adquiridos, ou seja, com uma teoria, acompanhados de princípios condutores que acrescentem novos conhecimentos com algum significado, a fim de explicar observações abductivamente conhecidas, sejam elas positivas ou negativas, na forma de factos explicitamente negados.

¹⁸ Para uma resenha ver: Anthony Kakas, Robert Kowalski, Francesca Toni, "Abductive Logic Programming", *Journal of Logic and Computation*, 2:719-770, 1993.

¹⁹ Cf. José Júlio Alferes, Carlos Viegas Damásio, Luís Moniz Pereira, "A logic programming system for non-monotonic reasoning", *Journal of Automated Reasoning*, 14:93-147, 1995.

O objectivo consiste na geração de regras que definam um conceito positivo, bem como a sua negação, de modo a que cubram todas as situações conhecidas observadas. Esta geração automática de novas regras encontra-se sujeita a um critério de apreciação pré-definido, ou seja, apenas determinadas regras e atributos são permitidos no processo generativo. Regras novas podem entrar em contradição entre si em determinadas circunstâncias, pelo que devem ser sujeitas a diagnóstico, de modo a identificar as possíveis revisões alternativas.

Para decidir que tipo de revisão deverá ser adoptado, é desejável que se façam novas observações sobre as teorias correntes actuais, cujos resultados, caso sejam conhecidos, permitirão decidir qual a melhor revisão a adoptar. Os resultados destas denominadas observações cruciais são então obtidos, quer seja através de solicitações do programa quer através de um processo de planeamento subsequente, executado de modo a levar a cabo as acções que conduzem aos resultados dessas observações.

Uma vez seleccionadas as revisões desejadas, com base nos resultados e em critérios de preferência programados, serão estas validadas através de um procedimento actualizado. De notar que as próprias regras revistas podem ser, por sua vez, submetidas a revisões posteriores caso tal se venha a revelar necessário.

Sem dúvida que o processo completo será repetido na base de novas informações, ou através de confrontação de postulados entre as diversas teorias envolvidas, com diferentes históricos, pontos de vista, geradores de regras, diagnósticos, revisores, preferências, planeadores, observações, e procedimentos actualizados incluindo o seu agente racional.

A confrontação epistemológica fia-se em argumentação e resolução mútuas. Para além do domínio legislativo e legal, o campo da discussão científica também se apoia em tais procedimentos, podendo beneficiar da sua automação.

Lidar com argumentação envolve um conjunto de ferramentas semelhante às utilizadas para diagnosticar e resolver problemas. A argumentação pode atacar as afirmações de outra corrente argumentativa directamente através da prova da negação de uma afirmação ou, indirectamente, através da contradição da conclusão de outra argumentação que se apoie nas suas afirmações. Contudo, tais ataques entre argumentações podem, por outro lado, ser contra-atacados do mesmo modo, podendo estes, por sua vez, contra-atacar, e assim sucessivamente. A Programação em Lógica mostrou de que modo este processo pode ser estudado, e que conclusões podem ser tiradas sobre argumentação mutuamente contraditória que compita entre si, e de que modo cada um desses tipos de argumentação pode ser revisto, de modo a ser encontrado um acordo.

8. O PROJECTO MENTAL

O esquema atrás descrito de um enquadramento lógico dinâmico, foi materializado dentro do contexto do nosso projecto corrente de agentes mentais²⁰, realizado no Centro de Inteligência Artificial da Universidade Nova de Lisboa.

Tanto na universidade como na indústria se sente cada vez mais que os agentes inteligentes serão uma tecnologia chave, à medida que os sistemas informáticos se tornam cada vez mais distribuídos, interligados e abertos. Em tais ambientes, a capacidade de os agentes autonomamente planearem e acompanharem as suas próprias acções e metas, de cooperarem, coordenarem e negociarem com terceiros e, deste modo, responderem de um modo flexível e inteligente à dinâmica e imprevisibilidade das situações, conduzirá a melhorias muito significativas na qualidade e sofisticação dos sistemas informáticos que podem passar a ser concebidos e implementados, bem como alargará as áreas alvo e os problemas que possam ser por eles abarcados.

O objectivo do projecto MENTAL consiste no estabelecimento, sobre uma sólida base teórica, de um desenho de uma arquitectura global para agentes mentais (ou seja, compreendendo conhecimentos, crenças e intenções) baseado e construído sobre os pontos fortes da Programação em Lógica.

Os agentes compartilharão um ambiente comum em mutação constante, ele próprio modelado em termos de Programação em Lógica. O enquadramento geral está a ser conceptualizado em Programação em Lógica, e testado através de Programação em Lógica distribuída. A utilização de agentes permite a distribuição e a partilha de informação, apenas quando exista necessidade. Deste modo, a complexidade inerente à manipulação do conhecimento pode ser reduzida. Também se ganha em eficácia em resultado da execução concorrente dos agentes.

Um agente deverá ser capaz de gerir o seu próprio conhecimento, crenças, intenções (objectivos) e planos, à medida que recebe informação e instruções novas, bem com de reagir às mudanças nas condições ambientais. Deverá ser também capaz de interagir com outros agentes, através do intercâmbio de conhecimentos e crenças, e de reagir às solicitações que receba de outros agentes. Quaisquer pares de agentes poderão cooperar entre si, quer através do diagnóstico de erros e da informação em falta nas bases de conhecimentos de cada um, quer cooperando na utilização de recursos comuns, no evitar da interferência mutua nos respectivos planos, e no planeamento conjunto a fim de atingirem um objectivo comum.

²⁰ Trata-se do projecto "MENTAL – Mental Agents Architecture in Logic", financiado pelo programa PRAXIS XXI, e identificado por 2/2.1/TIT/1593/95.

Cada agente é composto de subagentes especializados, eventualmente activos em paralelo e funcionalmente relacionados, que vão executando os diversos objectivos e instruções que lhes são transmitidas pelo respectivo agente. Exemplos destes subagentes são aqueles que implementam as funcionalidades de reactividade, de raciocínio, de planeamento, de revisão de crenças, de argumentação, de explicação, de aprendizagem, de gestão de diálogos, de recolha de informação, de avaliação de preferências, de estratégia e de diagnóstico. Os subagentes são coordenados por uma camada meta-nível que assegura a distribuição interna de tarefas e o respectivo planeamento, a comunicação entre os subagentes, as decisões finais e toda a interacção com o exterior, ou seja, não só com o ambiente como com os outros agentes, incluindo a comunicação, a gestão de interrupções, os pedidos e as observações.

Apesar de cada agente ter uma coordenação de meta-nível, a colecção de agentes interactivos não possui uma tal cobertura coordenadora, e o seu comportamento colectivo manifestará dificilmente previsíveis propriedades emergentes. Porque conhecimento e crença são em geral incompletos, contraditórios e atreitos a erros — e tanto mais num cenário de multi-agentes — recorremos à semântica, a procedimentos e a implementações que lidam com o contraditório, o incompleto, o erróneo, o imperfeito, a informação vaga e supletiva, a abdução, a revisão de crenças, a depuração e a argumentação.

Propomo-nos atingir a aprendizagem por reforço através da utilização de técnicas da revisão de crenças, de modo a conseguir um efeito semelhante à da propagação recessiva (back-propagation). Tal aprendizagem permitirá o evoluir da força de evidência associada ao conhecimento e às crenças de um agente, em resultado de respectiva contribuição para o surgimento de conclusões quer correctas quer incorrectas. Além disso, os agentes podem comparar e combinar os seus graus de evidência, seja para argumentarem seja para partilharem ou intercambiarem informações, ou para atingirem consensos. A utilização de algoritmos genéticos para fazer evoluir “genes” codificadores de crenças (ou “memes”), é também uma técnica que vale a pena ser explorada para a aprendizagem, uma vez que abre um novo campo: o da ligação, em agentes, entre a abordagem com algoritmos genéticos e a da evolução do conhecimento baseado em lógica. Além de que a troca de material genético permitirá aos agentes a realização de fertilização cruzada das suas experiências.

A utilização da Programação em Lógica para este empreendimento justifica-se na base de que ela fornece um suporte teórico abrangente para os tópicos atrás mencionados, bem como um veículo de implementação para processamento paralelo e distribuído. Adicionalmente, a Programação em Lógica constitui-se num instrumento formal flexível e de alto nível para a especificação e experimentação rigorosas de concepções computacionais, e

portanto bastante útil para a elaboração de protótipos, mesmo quando se pretendam utilizar outras linguagens de implementação de mais baixo nível.

9. COMENTÁRIOS FINAIS

De que modo poderão as questões gerais sobre o raciocinar, e exploradas aqui, serem abordadas segundo um enquadramento uniforme, a não ser dentro do contexto da lógica? Sustento que a lógica é o único ponto de encontro concebível para albergar tal empresa. E não o teremos que fazer? Não teremos porventura de mecanizar os vários procedimentos de raciocínio de modo a que tanto robôs como computadores possam executar, para nós ou connosco, os trabalhos por vezes monótonos por vezes excessivamente complexos inerentes a esses raciocínios? Certamente que sim. Só persistindo nessa direcção conseguiremos lidar com os cada vez mais exigentes problemas de raciocínio que temos pela frente.

Espero ter-vos convencido de que a I.A., muito especialmente através da Programação em Lógica, continuará a prestar uma preciosa contribuição para a identificação, formalização e implementação das leis do pensamento. Mais notavelmente, a I.A. aceitou o desafio de abrir a lógica à dinâmica do conhecimento em fluxo. E, ao fazê-lo, tem vindo progressivamente a satisfazer as nossas expectativas e requisitos. Dar continuidade a este trabalho é essencial se quisermos enfrentar com êxito, mesmo com o auxílio dos computadores, os desafios de um conhecimento cada vez mais acumulado e simultaneamente distribuído, neste mundo em mudança.

A participação dos filósofos poderá assumir quer a forma de tomarem parte no desenvolvimento dos substratos conceptuais das necessárias ferramentas lógicas a implementar em computador, quer ainda no emprego de tais ferramentas para a reconstrução racional, a funcionar em computador, da argumentação epistemológica histórica²¹, ou ainda para a duplicação da evolução das teorias científicas através da aprendizagem automática. A classificação taxionómica das espécies naturais poderia ser um bom ponto de partida.

10. AGRADECIMENTOS

Este trabalho foi parcialmente apoiado pelo projecto do programa PRA-XIS XXI, MENTAL 2/2.1/TIT/1593/95, e por uma bolsa de estudos da NATO durante o período em que o autor esteve de licença sabática no departamento de Ciências da Computação da Universidade Estadual da Califórnia,

²¹ Tal como pode encontrar-se por exemplo em: Imre Lakatos, *Proofs and Refutations: the Logic of Mathematical Discovery*, Worrall, J. and Zahar, E.G. (eds), Cambridge University Press, 1976.

LUÍS MONIZ PEREIRA

em Riverside, E.U.A.. Agradecimentos a Luís Correia, João Leite e Halina Przymusinska, pelos seus comentários perspicazes.

Luís Moniz Pereira
Dep. de Informática
Faculdade de Ciências e Tecnologia
Universidade Nova de Lisboa
Monte de Caparica
2825-114 Caparica
Imp@di.fct.unl.pt