

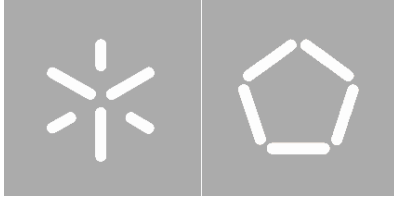
Universidade do Minho

Departamento de Informática

Estruturas Hierárquicas com Herança

Cesar Analide

José Neves



Universidade do Minho

Departamento de Informática

Estruturas Hierárquicas com Herança

Cesar Analide

José Neves

Texto Pedagógico

Grupo de Inteligência Artificial

Centro de Ciências e Tecnologias da Computação

Índice

1 ESTRUTURAS HIERÁRQUICAS COM HERANÇA	2
2 GRAFOS	7
3 ENQUADRAMENTOS	8
4 REDES SEMÂNTICAS	10
5 OUTRAS ESTRUTURAS	15
5.1 Guiões	16
5.2 Grafos Conceptuais.....	17
5.3 Objecto-Atributo-Valor.....	17
6 ESTRUTURAS HIERÁRQUICAS	18
6.1 Descrição da Estrutura do Sistema	19
6.2 Implementação do Raciocínio Não-Monótono.....	24
6.3 Representação de Informação Incompleta	36
6.3.1 Nulos do Tipo Desconhecido	38
6.3.2 Nulos do Tipo Desconhecido, de um Conjunto de Valores	44
6.3.3 Nulos do Tipo Não Permitido	48
7 RESUMO	51
8 REFERÊNCIAS	52

1 Estruturas Hierárquicas com Herança

Neste trabalho orientar-se-á a discussão para a representação de conhecimento segundo o modelo hierárquico, com herança, no contexto mais lato da representação de informação incompleta.

Apesar deste trabalho considerar, principalmente, uma aproximação aos modelos de representação de conhecimento baseados em redes semânticas, far-se-á a distinção entre uma rede semântica e a estruturação hierárquica do conhecimento. Uma rede semântica fornece uma esquematização plana da informação, enquanto que uma estruturação hierárquica caracteriza, em particular, a noção que se pretende defender sobre herança.

O trabalho inicial sobre modelos baseados em redes semânticas foi desenvolvido para abordar a representação de frases em Linguagem Natural (LN) [Fil88]. Por exemplo, a frase “*o João deu um livro à Maria*”, poderia ser representada através da rede semântica esquematizada na Figura 1.

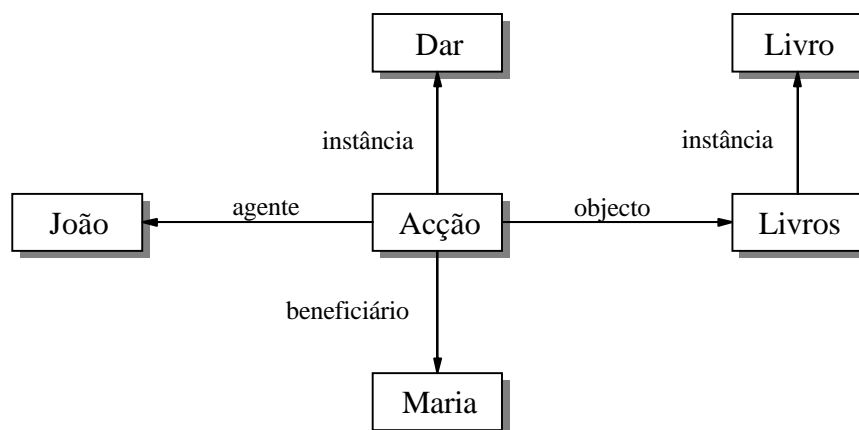


Figura 1: Representação de uma frase numa rede semântica

Neste sentido, uma **rede semântica** é uma rede por ser um “*emaranhado de coisas ou de circunstâncias*” [CoMe76], e é semântica, por tratar da “*parte da linguística que se ocupa da significação das palavras e da evolução do seu sentido*” [CoMe76], ou seja, uma rede semântica não é mais do que uma representação, considerada em termos gráficos, de um emaranhado de entidades, com o intuito de representar o significado das palavras.

As redes semânticas são, então, estendidas, dando corpo a **redes semânticas hierárquicas**, de modo a que se possa representar informação hierarquizada, com vista à aplicação de estratégias de raciocínio que permitam tirar proveito dos mecanismos de herança. Assim, as redes semânticas são passíveis de utilização, não só em sistemas de tratamento de LN, mas, também, em outros sistemas de representação de conhecimento, que utilizem a herança como referência. Um exemplo de uma estrutura deste tipo é dada pela na Figura 2. Este tipo de estruturas permite dividir a informação por níveis de especialização, mais genérico o conhecimento no topo da rede, mais específico na base.

Nesta classe de formalismos para a representação de conhecimento há ainda

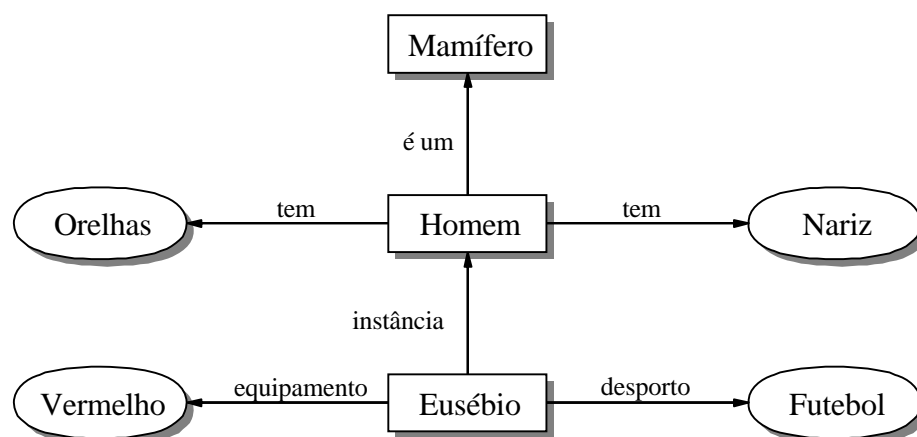


Figura 2: Rede semântica hierárquica

que considerar os *enquadramentos*¹ [Min81], [RiKn91]. Os enquadramentos não são mais do que colecções de atributos, aos quais podem ser associados valores, apontadores para outros enquadramentos, regras ou procedimentos, ou outras entidades, que possibilitem descrever o mundo real. Geralmente, os enquadramentos são agrupados entre si, formando redes ou estruturas hierárquicas. Na Figura 3 apresenta-se um esquema exemplificativo deste tipo de estrutura.

Tal como *Marvin Minsky* refere, um enquadramento não é mais do que uma rede de nodos e relações, em que os níveis superiores são fixos e os inferiores compreendem atributos, os quais deverão ser concretizados com instâncias específicas ou dados [Min81]. Esta descrição indicia uma estruturação hierarquizada da informação representada por um *sistema de enquadramentos*.

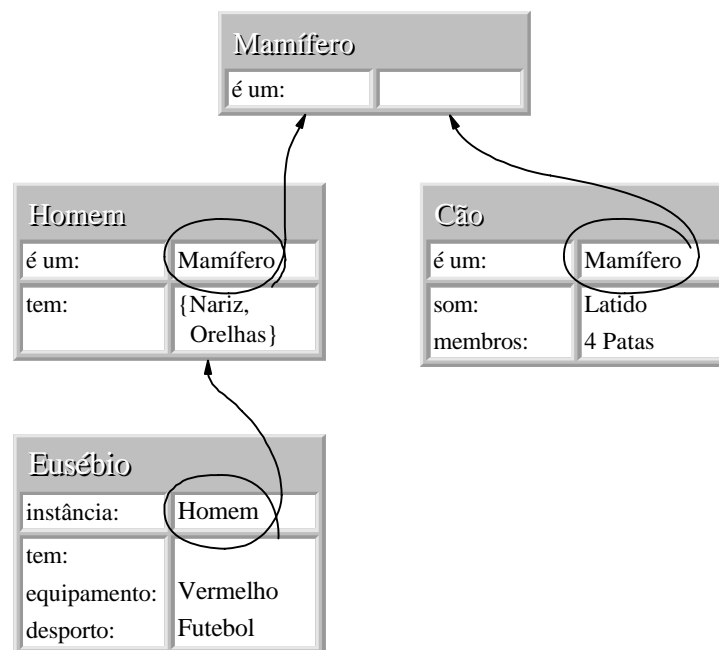


Figura 3: Exemplo de um esquema de enquadramentos

¹ Na bibliografia, utiliza-se, normalmente, o termo “*frame*” para designar este tipo de estruturas que aqui são designadas por “*enquadramentos*”.

A principal diferença entre enquadramentos e redes semânticas está em que, num esquema de enquadramentos, os atributos são fixos para uma determinada entidade, podendo ou não ter valores associados, enquanto que numa rede semântica, cada nodo pode comportar várias propriedades, e não necessariamente as mesmas entre instâncias particulares de um mesmo nodo.

Todos estes formalismos têm um denominador comum: a representação de informação é realizada de uma forma estruturada, ou seja, quaisquer dos modelos utilizados são, nem mais nem menos, do que particulares instanciações de um arquétipo, o grafo.

Um **grafo** materializa uma estrutura de dados constituída por dois tipos de elementos: **nodos** e **arcos**. Conforme os arcos representem ligações dirigidas ou não, o grafo diz-se orientado ou não orientado, respectivamente. Um arco une dois nodos, representando uma relação entre a informação que estes contêm. Os nodos representam entidades ou conceitos e podem estar, ou não, ligados entre si. A Figura 4 ilustra dois tipos de grafos: orientados e não orientados.

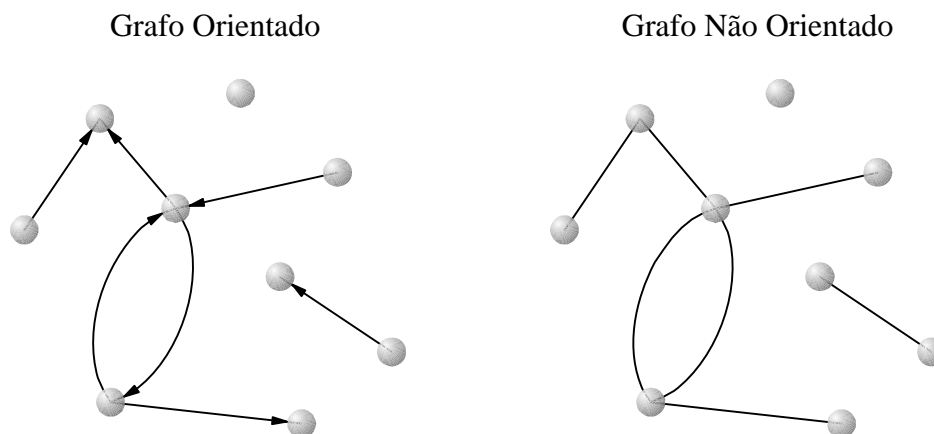


Figura 4: Esquematização típica dos grafos

Note-se que podem existir diversos arcos a ligar dois nodos, e isto porque,

um arco, representa a existência de uma relação entre dois nodos.

Apesar de a Figura 4 apresentar em esboço, simples grafos, não se deve partir do princípio que um grafo é, essencialmente, uma estrutura simples, dependendo, a complexidade da estrutura, das entidades que se estejam a representar ou das relações que se pretendam estabelecer.

De entre as estruturas que têm vindo a ser analisadas, aquela que se pretende adoptar como arquétipo, neste trabalho, será uma amálgama de todas estas, enfatizando uma estruturação hierárquica da informação (Figura 5).

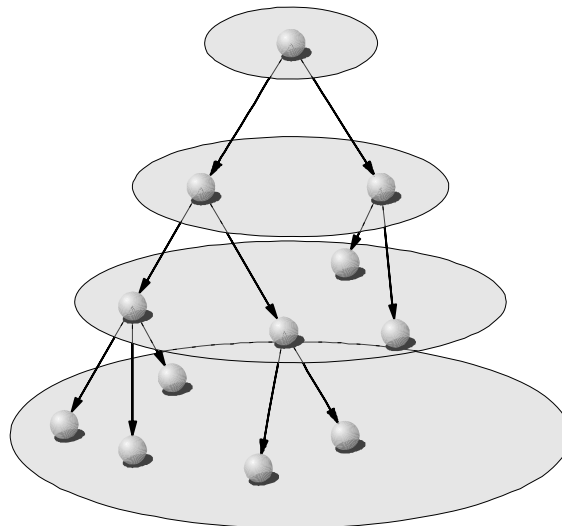


Figura 5: Estruturação hierárquica da informação

Esta estruturação hierárquica da informação não deverá ser:

- tão simples na esquematização quanto um grafo; pretende-se que a partir da apresentação da estrutura seja dado o primeiro passo para a identificação da informação em jogo;
- tão rígida na elaboração quanto os enquadramentos; pretende-se desenvolver um sistema que seja mais flexível do que a descrição da

informação através de estruturas que se assemelham a registos;

- tão densa quanto uma rede semântica; devido à grande variedade de tipos de ligações presentes numa rede semântica, esta pode-se revelar ilegível, o que dificulta quer o seu desenvolvimento, quer a sua compreensão, quer, ainda, a sua evolução.

Uma rede semântica hierárquica é a estrutura mais próxima daquela que se pretende desenvolver. Contudo, a expressão “*rede semântica*” revela-se demasiado “*pesada*” por estar associada, basicamente, a sistemas de tratamento e processamento de LN. Daí que se tenha adoptado a expressão “*estrutura hierárquica*”, por forma a denominar uma estrutura para a representação de conhecimento que, acima de tudo, contemple uma hierarquia de entidades num sistema capaz de tirar proveito de possíveis mecanismos de herança.

A estrutura hierárquica referida surge como uma evolução natural de um grafo, em que os nodos se organizam hierarquicamente, tal como acontece nos enquadramentos, segundo ligações (arcos) que estabelecem a hierarquia entre os nodos e lhes associam propriedades ou valores, como no caso das redes semânticas e dos enquadramentos.

A motivação para a introdução de uma designação distinta de qualquer das mencionadas, está no facto de não só se pretender dissociar uma estrutura hierárquica (como a referida) da de uma rede (estrutura plana) e da organização de um sistema de enquadramentos, mas também por esta ser uma arquitectura especializada de um grafo. Por outro lado, uma estrutura hierárquica permite o tratamento do conceito de herança de uma forma natural.

2 Grafos

Basicamente, um *grafo* é uma colecção de *nodos* e *arcos*, com os arcos ligando

nodos entre si. Tudo quanto se possa descrever por aplicação destas entidades, pode ser caracterizado como um grafo.

Apesar de, abstractamente, um grafo ser uma estrutura bastante simples, baseada em duas entidades de representação, a complexidade destes dependerá da informação que se fizer representar nos nodos e nos arcos.

Um nodo pode representar uma entidade tão simples quanto um ponto de passagem ou um repositório de informação, como algo mais complexo, por exemplo, agentes cognitivos, com todo um manancial de informação e raciocínio associados.

Por sua vez, também um arco pode servir de ligação, simplesmente, entre dois nodos, ou representar complexas relações ou condições de união entre estes.

3 Enquadramentos

Os enquadramentos são arquétipos propostos inicialmente por *Marvin Minsky* [Min81], como estruturas de dados para a representação de situações estereotipadas. A motivação por si encontrada para a introdução deste tipo de estrutura deveu-se ao facto de que, em sua opinião, quando se está perante uma nova situação, recorre-se à memória, efectuando-se enquadramentos com outras situações similares, adaptando-as de modo que se encaixem na realidade, por modificação, apenas, dos detalhes necessários.

Tendo começado por aplicar estes conceitos a sistemas de visão, os vários enquadramentos descrevem uma mesma cena segundo diferentes ângulos de visão. Torna-se, assim, necessário que a estrutura dos enquadramentos seja idêntica entre si, por forma a que seja possível relacionar e coordenar a informação obtida.

Um enquadramento é uma estrutura fixa, semelhante aos registos de tabelas numa base de dados relacional, composta por uma colecção de alvéolos (ou *slots*),

que se podem identificar com os atributos de um registo, mas que podem conter, para além de valores, referências a outros enquadramentos, regras ou procedimentos, ou outras entidades.

Um enquadramento tanto pode descrever uma entidade abstracta como representar uma instanciação particular dessa entidade.

Raramente um enquadramento isolado tem qualquer significado ou utilidade. Em vez disso, constroem-se sistemas de enquadramentos a partir de colecções de enquadramentos, que se relacionam em virtude do facto de que um valor de um alvéolo de um enquadramento, pode ser um apontador para outro enquadramento.

Considera-se a existência, basicamente, de dois géneros de enquadramentos: os que representam uma entidade abstracta ou uma classe de objectos, e os que representam uma instância ou elementos particulares dessa classe de objectos. Também numa classe se podem considerar dois tipos de alvéolos: os que representam informação específica sobre a classe e os que contêm dados que podem ser herdados pelas suas subclasses ou instâncias. Atente-se no sistema de enquadramentos representado na Figura 6.

Nesta figura, considere-se o enquadramento que caracteriza a classe dos papagaios. Neste enquadramento existe um alvéolo com informação específica sobre a classe, a de que um papagaio é uma ave, bem como três outros que representam propriedades que se verificam para a classe dos papagaios, a saber, o som que estes emitem, a cor que apresentam e o tipo de alimentação. Nestas propriedades foram especificados valores por defeito para duas delas.

Admitindo que uma classe seja caracterizada por estes dois tipos de informação, está-se a permitir que a classe defina uma estrutura para os seus elementos e a descrever um protótipo de possíveis instâncias, através da introdução dos valores por defeito.

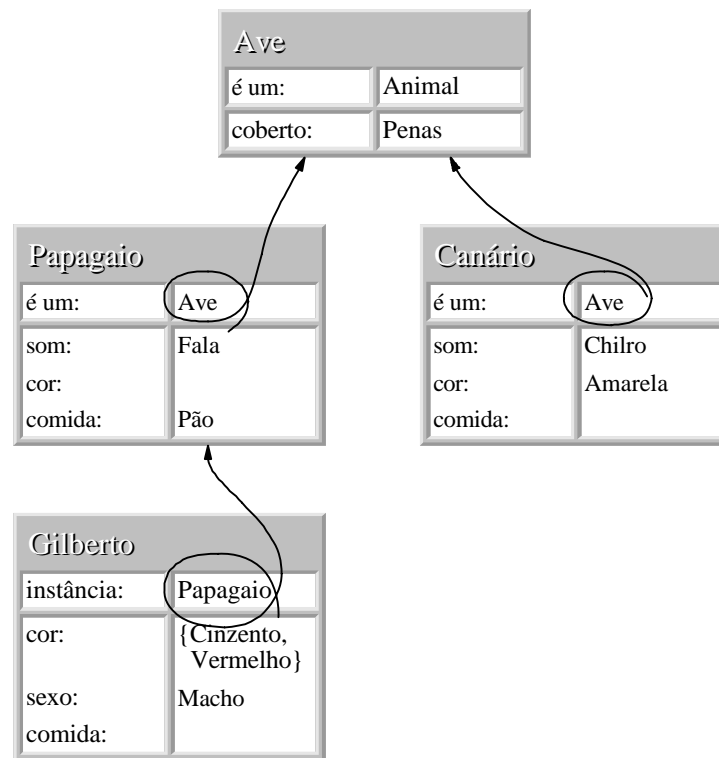


Figura 6: Sistema de enquadramentos sobre aves

Os alvéolos que se encontram por preencher suportam a representação de informação incompleta ao nível desse enquadramento ou atributos cujos valores devem ser herdados das correspondentes classes a que se associam.

Apesar deste tipo de modelo de representação de conhecimento não ter sido isento de críticas, a sua aceitação ficou a dever-se ao facto de o conhecimento se apresentar estruturado, quer ao nível da informação do domínio do problema a tratar, quer ao nível da forma como o conhecimento se relaciona.

4 Redes Semânticas

As *redes semânticas* fazem parte de uma classe de formalismos de representação

de conhecimento que tem vindo a adquirir grande aceitação na área científica da Inteligência Artificial (IA). Existem diferentes formas de redes semânticas em IA, mas todas se agrupam segundo um denominador comum, por partilharem uma mesma génese, consistindo num conjunto de *nodos* e num conjunto de *arcos* [RiDu88], [Way91].

Os nodos representam objectos, conceitos ou situações, com os arcos a representarem as relações entre aqueles. Embora não exista nenhuma regra que defina, em absoluto, o que cada um destes elementos deve representar, esta é, em princípio, a associação que se segue e que resulta mais natural. Isto permite reduzir as redes semânticas a simples especializações de grafos, em que as ligações (arcos) representam utilidades e os nodos denotam entidades.

É geralmente atribuído a *Ross Quillian* o pioneirismo na utilização de redes semânticas, em particular no tratamento de LN, em que o modelo adoptado não era mais do que um conjunto de nodos, ligados por diferentes tipos de arcos. Os nodos correspondiam a conceitos em LN e representavam o significado das palavras, directa ou indirectamente. Não dando qualquer justificação para a utilização dos arcos, para além da necessidade de relacionar os conceitos descritos pelos nodos, introduziu diferentes tipos de ligações, dos quais há a destacar os “*apontadores*”, como forma de referenciar nodos especiais de tipificação da estrutura. O resultado foi uma estrutura tão difícil de entender quanto difícil de implementar.

Na verdade, não é esta aplicação de redes semânticas ao tratamento da LN que interessa discutir. Está-se interessado, isso sim, em descrever conhecimento que possa ser organizado hierarquicamente, pelo que a discussão se dirigirá no sentido da abordagem às redes semânticas hierárquicas.

A base teórica não se altera: à semelhança do que acontece com as redes semânticas, uma rede semântica hierárquica é constituída por nodos e arcos. A diferença está em que, neste último caso, os arcos agrupam os nodos de modo a

organizá-los hierarquicamente. Pode-se considerar esta forma de rede semântica como uma particularização da anterior, em que os arcos são de um conjunto mais restrito de tipos, permitindo relacionar os conceitos descritos pelos nodos em termos de uma estruturação hierárquica da informação.

A representação gráfica de um sistema de conhecimento hierárquico é bastante mais apelativo, tal como se pode constatar pela Figura 7, por obrigar a uma organização mais estruturada da informação. Embora esta figura, que representa, a título de exemplo, uma rede semântica hierárquica, apresente uma notação gráfica ligeiramente diferente da Figura 2, os grafismos usados nestas duas representações são equivalentes, uma vez que a diferença está, essencialmente, numa maior riqueza gráfica da Figura 2, em contraste com uma maior simplicidade da Figura 7.

Uma rede semântica hierárquica pode ser vista como uma estrutura de organização de tipos, agrupados por níveis de generalidade, em que, no topo da hierarquia, os conceitos são mais genéricos e na base estão representados elementos individuais ou instâncias. Por este motivo, se diferenciam os nodos da rede que representam classes ou tipos de dados, daqueles que identificam indivíduos específicos dessas classes, como é o caso, no exemplo dado pela Figura 7, do indivíduo Gilberto, que é uma instância particular da classe dos papagaios, apresentando, inclusivamente, características próprias daquelas que se encontram, genericamente, nos papagaios.

Este tipo de estruturação da informação de acordo com níveis de generalização, suporta eficazmente o mecanismo da herança. Deste modo, as propriedades e relações de qualquer classe podem ser *herdadas* por todas as suas subclasses. Por exemplo, tendo em conta a Figura 7, sabendo que os papagaios falam e comem pão e que o Gilberto é uma instância da classe dos papagaios (logo, é um papagaio), para além das propriedades específicas do Gilberto, fica-se a saber que este também fala e também come pão. Poder-se-ia ir mais longe na hierarquia e

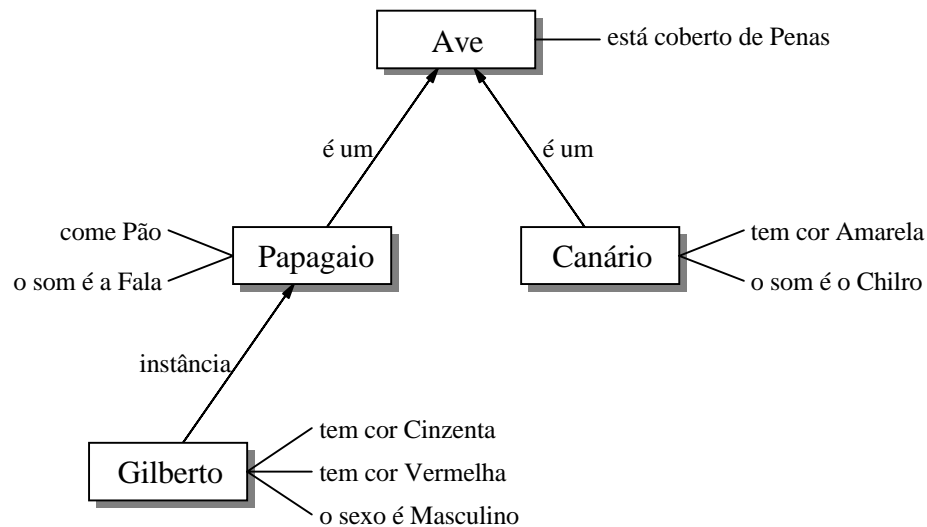


Figura 7: Rede semântica hierárquica sobre as aves

concluir que o Gilberto tem o corpo coberto de penas, uma vez que é um papagaio, que os papagaios são aves e que as aves têm o corpo coberto de penas.

A hierarquia permite inferir estes factos, em termos de procedimentos idênticos aos subjacentes, por exemplo, na regra de derivação *modus ponens*: o Gilberto é um papagaio, os papagaios são aves, logo, o Gilberto é uma ave. Desta forma, não é necessário, ao nível da entidade Gilberto, descrever explicitamente toda a informação respeitante a esta, como, por exemplo, estar coberto de penas, alimentar-se de pão ou emitir um som conhecido. Essa conclusão é obtida pela forma como a informação se encontra estruturada.

Pode-se, por conseguinte, afirmar que as redes semânticas hierárquicas implementam uma forma particular de raciocínio por defeito, uma vez que as propriedades herdadas por uma instância são do mesmo tipo da informação que se associa a todas as instâncias dessa classe.

Ainda no que respeita à classificação das aves, é, também, uma propriedade

comum a quase todas, a circunstância de se movimentarem através do voo. Contudo, nem todas as aves voam, como é o caso dos pinguins e das avestruzes. Nesta situação, entidades das subclasses poderão herdar a propriedade do voo, a menos que exista informação que defina o contrário para as entidades das subclasses que são exceção a essa regra. Esta forma de cancelamento de herança traduz uma forma particular de raciocínio por defeito, tal como foi abordado em [Ana96].

Contudo, a herança de propriedades por defeito apresenta alguns óbices, no que respeita, principalmente, à verificação de *exceções* e à *herança múltipla*. Esse é o caso, como se viu, das aves que não voam. Quando se aplicam estratégias de raciocínio por defeito, uma determinada conclusão surge como função de um conjunto de premissas, e em ausência de informação explícita que defina o contrário. O problema surge neste ponto!... Numa rede semântica hierárquica pode não existir informação explícita que defina a exceção, uma vez que essa informação se pode encontrar algures na estrutura hierárquica da rede.

O problema da herança múltipla surge quando uma subclasse, em particular, está relacionada com mais do que uma superclasse, das quais pode herdar propriedades que entrem em conflito. Atente-se na Figura 8.

Da Figura 8 pode-se concluir que os vampiros são aves. Uma vez que as aves voam, os vampiros também voam. No entanto, os vampiros também são animais mortos e os animais mortos não voam. Em função desta última observação, os vampiros não deveriam voar. Este exemplo ilustra o problema de uma determinada subclasse poder herdar propriedades que entram em conflito, em virtude de estar relacionada com duas superclasses.

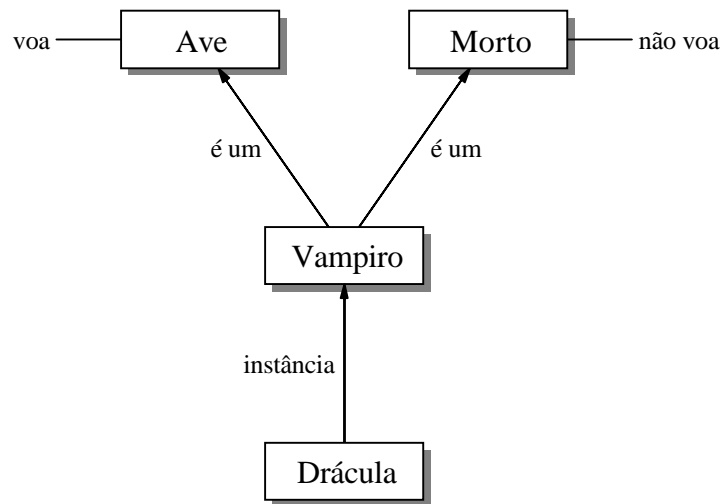


Figura 8: Herança múltipla em redes semânticas hierárquicas

Apesar destes problemas, para os quais têm sido apresentadas diferentes soluções, endossar uma estrutura hierárquica à qual se podem aplicar mecanismos de herança, tirar partido de esquemas de raciocínio por defeito, atender à simplicidade de representação de informação incompleta, são atributos que abonam a favor da utilização deste modelo de representação de conhecimento.

5 Outras Estruturas

Existem outros modelos para a representação de conhecimento que, independentemente da importância que tenham adquirido ou da relevância que hajam atingido, podem sempre ser vistas como especializações das estruturas que foram abordadas até ao momento.

5.1 Guiões²

De entre as formas para a representação de conhecimento, em que há a considerar o tratamento de sequências de acções ou eventos, tem-se os *guiões*.

Um *guião* define-se como uma estrutura que descreverá uma sequência estereotipada de eventos num determinado contexto, constituída por um conjunto de alvéolos (*slots*), aos quais pode estar associada informação sobre o tipo de dados que podem conter. Existe, também, a possibilidade de definir valores por defeito, para o caso de não existir disponível outra informação. Nestes termos, ainda são ténues as diferenças relativamente aos enquadramentos, na medida em que a característica que diferencia os guiões dos enquadramentos está em que os primeiros são utilizados para descrever sequências de eventos.

Neste sentido, os guiões são aplicados, essencialmente, quando a descrição do mundo real que se pretende abordar apresenta padrões de ocorrência de eventos, que surgem devido à forma como a ocorrência de um evento pode condicionar o acontecimento de terceiros.

A estrutura base de um guião apoia-se no encadeamento de eventos. O encadeamento tem início com a verificação das *condições de entrada*, que determinam a ocorrência dos primeiros eventos. O fim desse encadeamento define um conjunto de *resultados* que podem potenciar a ocorrência de outros eventos, eventos esses não necessariamente descritos através dos mesmos guiões. Neste encadeamento de acontecimentos, um evento está ligado a eventos anteriores que lhe deram origem e a eventos posteriores para cujo desencadeamento contribuiu.

Se um guião é apropriado para descrever uma determinada situação, então, será possível prever a ocorrência de eventos subsequentes, mesmo quando não hajam sido explicitamente mencionados.

² Embora o termo mais comum encontrado na bibliografia para a identificação destas estruturas seja “*scripts*”, aqui designar-se-ão por “*guiões*”.

Segundo esta perspectiva, um guião é uma estrutura de representação semelhante aos enquadramentos e que pode ser comparado a um guião cinematográfico, ou seja, à descrição de uma sequência de acontecimentos interpretados pelos agentes em cena, com papéis bem definidos.

5.2 Grafos Conceptuais

Os *grafos conceptuais* são uma aproximação às redes semânticas, introduzida por John Sowa [Sow84]. Embora dirigidos, em primeiro lugar, ao processamento de LN, o sistema está baseado em lógica de primeira ordem e foi concebido de modo a automatizar o cálculo lógico.

Um grafo conceptual representa uma frase através de uma rede semântica, sendo constituído, basicamente, por três entidades: conceitos, relações e ligações. Um *nodo conceptual* representa entidades, atributos, estados ou eventos, enquanto que um *nodo relacional* define o modo como os conceitos se relacionam. Cada relação liga-se, apenas, ao número de conceitos que define, e os conceitos estão unidos a uma ou mais relações.

Uma *ligação* não traduz qualquer significado em particular, apenas indica quais os conceitos com que cada relação lida directamente.

Desta forma, um grafo conceptual não é mais do que uma especialização de uma rede semântica dirigida, em particular, ao tratamento de LN.

5.3 Objecto-Atributo-Valor

A representação de conhecimento por triplos *Objecto-Atributo-Valor* (O-A-V), é uma forma comum de tratamento de informação factual [Hsl90].

Neste modelo, os *objectos* representam entidades abstractas, físicas ou conceptuais, os *atributos* denotam características genéricas ou propriedades

associadas aos objectos, e os *valores* especificam a natureza de um atributo numa situação particular.

Sistemas deste tipo podem ser vistos como especialização de redes semânticas, de onde se elimina uma grande variedade de ligações, substituindo-as por, apenas, duas: a ligação objecto-atributo e a ligação atributo-valor. Também podem ser vistos como uma aproximação aos sistemas de enquadramentos. A escolha entre a metáfora do enquadramento e a metáfora da rede semântica é, apenas, determinada pela escolha da representação das redes, isto é, optando-se por representar as redes como associações de alvéolos ou de grafos, respectivamente. Contudo, em termos de significado e implementação, poderá acontecer que nada haja a separar os dois sistemas.

Os sistemas de conhecimento baseados em triplos O-A-V, caracterizam-se pela forma como fazem a distinção entre a descrição abstracta e estática de objectos, e o modo como, dinamicamente, se materializam em termos de instâncias desses mesmos objectos, pela forma como tratam a organização dos objectos, ordenação e relacionamentos entre si e, por último, pela sua capacidade para representar informação incompleta, à semelhança do que acontece nos sistemas de enquadramentos.

6 Estruturas Hierárquicas

O sistema hierárquico de representação de conhecimento que será objecto de estudo deverá permitir a aplicação de mecanismos de herança, disponibilizar formas de raciocínio por defeito e possibilitar o tratamento de informação incompleta.

Será uma sistema hierárquico, com paralelo nas redes semânticas hierárquicas, organizando a informação através de diferentes níveis de abstracção (Figura 5).

Considerando o esquema da Figura 5 como uma extensão aos grafos, cada nodo deste grafo, ou, mais precisamente, cada agente cognitivo desta estrutura hierárquica, pode ser visto como sendo constituído por um conjunto de itens de informação, capazes de descreverem as propriedades que o individualizam (Figura 9).

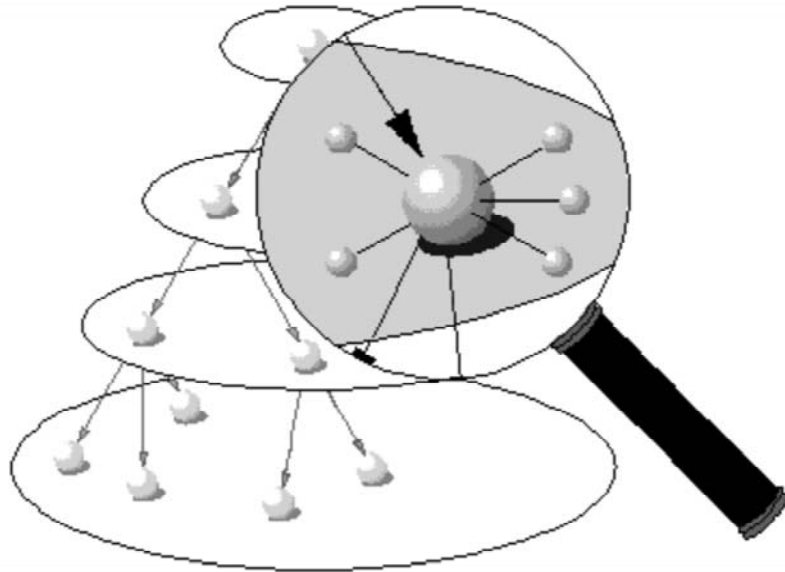


Figura 9: Organização dos agentes na estrutura hierárquica

6.1 Descrição da Estrutura do Sistema

A estrutura que comportará o conhecimento descrito neste sistema será baseada nas entidades *arco* e *nodo*, sobre grafos.

Os nodos representarão entidades específicas do domínio do problema, como, por exemplo, objectos, figuras, animais, pessoas, ou suas classes ou instâncias individuais, as quais recebem, por definição, a designação de “*agente*”, em detrimento da denominação de “*nodo*”.

Como se mostra na Figura 10, a cada agente pode ser associado um conjunto

de propriedades, que contribuem para a caracterização do conhecimento que o agente representa.

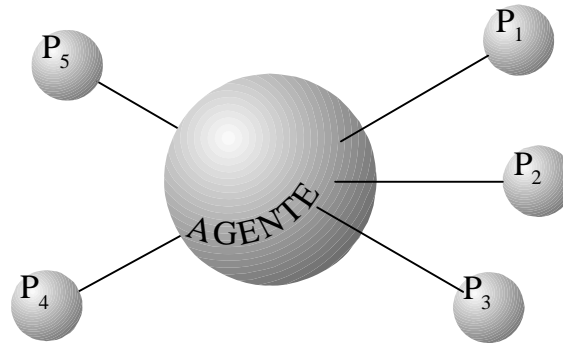


Figura 10: Constituição de um agente

Os arcos serão empregues como forma de relacionamento entre as entidades constituintes da estrutura (isto é, quer os agentes quer as suas propriedades), representando dois tipos de ligações distintas: um tipo estabelecerá a ligação entre dois agentes da estrutura, determinando a disposição hierárquica dos mesmos (Figura 5); outro tipo de ligação permitirá associar ao agente as suas propriedades (Figura 10).

O primeiro tipo de ligação mencionado corresponde à ligação ISA (do inglês “*is a*”), em que se estabelece uma relação de correspondência entre duas entidades, entre as quais se diz existir um relacionamento instância-classe ou subclasse-classe. São exemplos deste tipo de ligações a constatação de que o *Tweety* é uma ave e de que as avestruzes são aves. Materializando este relacionamento através da extensão da relação “*é um*”, esta informação traduz-se pelo grafismo dado pela Figura 11.

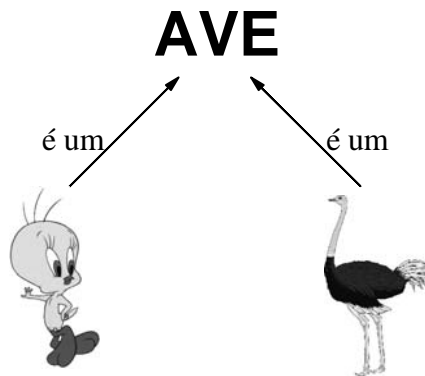


Figura 11: Exemplo da relação “é um”

Em termos da Programação em Lógica (PL), pode-se descrever este relacionamento pela introdução do predicado *é um*, composto por dois argumentos, em que o primeiro identifica a subclasse (ou instância) e o segundo identifica a classe a que aquela se liga:

$$\text{é um: SubClasse, Classe} \rightarrow \{ \mathbb{V}, \mathbb{F} \} \quad (1)$$

Para o exemplo da Figura 11, a extensão do predicado *é um* é dada pelo excerto do programa em lógica transcrito no Programa 1.

```
é um(tweety,ave)
é um(avestruz,ave)
```

Programa 1: Extensão do predicado *é um*

Na estrutura hierárquica que aqui se apresenta, não se distingue explicitamente o tipo de relacionamento instância-classe e o tipo subclasse-classe, sendo ambos os tipos expressos pela relação “*é um*”.

O segundo tipo de ligação mencionado, o que estabelece uma conexão entre o agente e as suas propriedades, será identificado pela simples associação, ao

agente, do valor específico da propriedade que este apresenta. Concretizando, podem-se enumerar algumas das propriedades que, geralmente, se encontram nas aves, como sejam a postura de ovos, a capacidade de voar, o facto de serem cobertas de penas, o facto de se alimentarem de minhocas e de raízes, etc. (Figura 12).

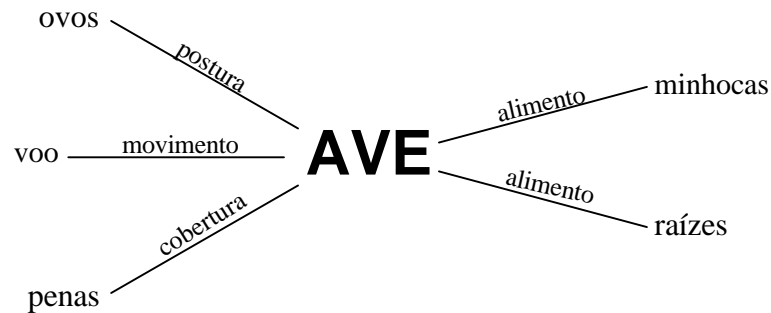


Figura 12: Propriedades das aves

Em termos de um programa em lógica, utilizar-se-á um predicado identificando a propriedade, p , com dois argumentos, sendo o primeiro o agente a que corresponde a propriedade e o segundo o valor particular da propriedade apresentada pelo agente:

$$p: \text{Agente}, \text{Valor} \rightarrow \{ \text{V}, \text{F} \}$$

A descrição das propriedades do agente *Ave* (Figura 12), será dada pelas produções:

```
postura(ave, ovos)
movimento(ave, voo)
cobertura(ave, penas)
alimento(ave, minhocas)
alimento(ave, raízes)
```

em que se supõe a existência de cada um dos predicados referenciados.

Deste modo, o agente terá as suas propriedades equacionadas na forma da extensão de um *conjunto de predicados*, ou seja, em termos de um programa em lógica, poder-se-á ter essa informação representada com recurso à notação de listas.

A consolidação desta descrição em termos de um programa em lógica, será feita através da extensão do predicado agente, com dois argumentos: a identificação do agente e a lista de propriedades que exhibe:

$$\text{agente: Agente, Propriedades} \rightarrow \{ \mathbb{V}, \mathbb{F} \}$$

A representação, em termos da PL, da informação descrita pela Figura 12, é dada no Programa 2.

```
agente (ave, [postura (ave, ovos) ,
               movimento (ave, voo) ,
               cobertura (ave, penas) ,
               alimento (ave, minhocas) ,
               alimento (ave, raízes) ] )
```

Programa 2: Representação das propriedades do agente ave

A apresentação feita sobre os constituintes elementares da estrutura hierárquica em desenvolvimento, baseada nos conceitos de *arco* e *nodo* sobre grafos, resultou numa organização da informação, através de ligações descritas pelo predicado *é um*, e na consolidação da informação existente ao nível dos agentes de conhecimento, pelo predicado agente, que inclui a informação disponível sobre as entidades que estão a ser representadas.

Daí resulta que a informação dada na Figura 11 e na Figura 12 pode ser descrita pelo Programa 3, onde se faz uso dos predicados agente e *é um*, como meio de representar a informação ao nível dos agentes e respectivas ligações.

```

agente (ave, [postura(ave, ovos) ,
               movimento(ave, voo) ,
               cobertura(ave, penas) ,
               alimento(ave, minhocas) ,
               alimento(ave, raízes) ] )
agente (tweety, [] )
agente (avestruz, [] )

é um(tweety, ave)
é um(avestruz, ave)

```

Programa 3: Informação contida na estrutura hierárquica

De notar que a não existência de propriedades definidas para um determinado agente, corresponde a um conjunto vazio de propriedades, ou seja, à existência de uma lista vazia associada à identificação do agente, como são os casos do *tweety* e da *avestruz*, no Programa 3.

6.2 Implementação do Raciocínio Não-Monótono

Em [Ana96] foi abordado o tema da Monotonia *versus* Não-Monotonia, sendo referida uma aproximação ao raciocínio por defeito em termos das estruturas hierárquicas, no âmbito da utilização de mecanismos de herança. Nesse contexto, o Raciocínio Não-Monótono (RNM) era implementado com recurso à identificação de situações anómalas (ou excepções), em relação às regras de cálculo.

No contexto actual, a informação associada ao agente é representada no nível em que este surge na hierarquia. Para aqueles que, na hierarquia, são subclasses ou instâncias de outras entidades, devido ao mecanismo de herança subjacente à estrutura, gozam de todas as propriedades definidas em níveis superiores da hierarquia. A menos, claro está, que algo em contrário seja definido.

O que falta, de momento, é dotar o sistema de conhecimento apresentado na subsecção anterior, de um mecanismo que permita resolver problemas em função destes pressupostos. Considere-se, para tal, a construção da extensão do predicado *prova*, que implementa, num esquema de representação de conhecimento como o descrito em epígrafe, os mecanismos de inferência adequados à construção de provas e obtenção de soluções, tendo em conta a problemática da implementação de mecanismos de herança. O predicado *prova* será dado com dois argumentos: a identificação do agente que se pretende inquirir e a questão que se pretende resolver:

$$\text{prova: Agente, Questão} \rightarrow \{ V, F \}$$

Para a definição deste predicado teve-se em atenção que:

- i) a questão pode ser provada ao nível do agente a que é dirigida;
- ii) poderá ser possível encontrar uma prova para a questão colocada no universo de conhecimento dos agentes em níveis superiores da hierarquia, devendo, então, a questão ser (re)dirigida a esse(s) agente(s).

Deste modo, é possível passar à construção da extensão do predicado *prova*:

$$\begin{aligned} \text{prova}(A, Q) \leftarrow \\ \text{agente}(A, P) \wedge \\ \text{processar}(Q, P) \end{aligned}$$

Esta cláusula implementa a situação enunciada em i): se existir, no conjunto (lista) de propriedades *P* do agente *A* a possibilidade de processar a questão *Q*, então a solução para *Q* será encontrada no corpo de conhecimento do próprio agente *A*.

O predicado *processar* que surge na cláusula em epígrafe implementa o

mecanismo de prova para a questão Q , a partir da lista de propriedades (axiomas) P do agente A .

Por outro lado, segundo o ponto *ii*), há que encontrar um agente a um nível hierárquico superior, por forma a ser-lhe dirigida a questão:

$$\begin{aligned} \text{prova}(A, Q) \leftarrow \\ \quad \text{é um}(A, C) \wedge \\ \quad \text{prova}(C, Q) \end{aligned}$$

Se existir uma classe C definida para o agente A , a solução será procurada recursivamente, reinvocando o predicado *prova*, mas, desta feita, dirigindo a questão Q à classe C .

Considere-se o Programa 4. A extensão do predicado *prova* descrito neste programa contém uma terceira cláusula, concretizando o PMF, significando que é falso que seja possível resolver a questão Q para o agente A , se não existir uma prova de que isso é possível.

$$\begin{aligned} \text{prova}(A, Q) \leftarrow \\ \quad \text{agente}(A, P) \wedge \\ \quad \text{processar}(Q, P) \\ \text{prova}(A, Q) \leftarrow \\ \quad \text{é um}(A, C) \wedge \\ \quad \text{prova}(C, Q) \\ \neg \text{prova}(A, Q) \leftarrow \\ \quad \text{não prova}(A, Q) \end{aligned}$$

*Programa 4: Extensão do predicado *prova**

O Programa 4 pode ser utilizado para, de uma forma bastante simples, questionar uma base de conhecimento construída segundo a estrutura definida para a representação de conhecimento, na forma aqui abordada. Utilizando este

programa, juntamente com a informação descrita pelo Programa 3, pode-se colocar, por exemplo, a questão:

`prova (ave, movimento (X, Y)) ?`

cujo significado é o de questionar a base de conhecimento sobre a forma de movimento do agente *ave*. Obviamente que esta questão tem solução, encontrada, em particular, ao nível da informação definida para o próprio agente questionado, quando a variável *X* identifica uma *ave* e a variável *Y* denota o *voou*.

No caso de a questão ser:

`prova (avestruz, movimento (X, Y)) ?`

continuará a ter solução, exactamente a mesma que para a questão anterior, uma vez que, embora o agente *avestruz* não possua qualquer tipo de informação definida (ver Programa 3), através do mecanismo de herança implementado pelo predicado `prova`, a *avestruz* tem o “*privilégio*” de gozar de todas as propriedades da classe a que está associada.

Se não fosse o facto de se estarem a obter conclusões que se sabe não serem reais³, tudo estaria correcto. Mas não é por qualquer erro do sistema que isto se passa.

O exemplo introduzido na subsecção anterior para apresentar a estrutura do sistema hierárquico que se pretende estudar é, também ele, suficientemente elementar para permitir uma primeira aproximação, tão simples e clara quanto possível, ao problema.

Esse exemplo não contempla, entre outras, situações como a das avestruzes não voarem ou a do *Tweety* não se alimentar de raízes. Considere-se a informação descrita na Figura 13.

³ As avestruzes não voam, a não ser que viajem de avião!

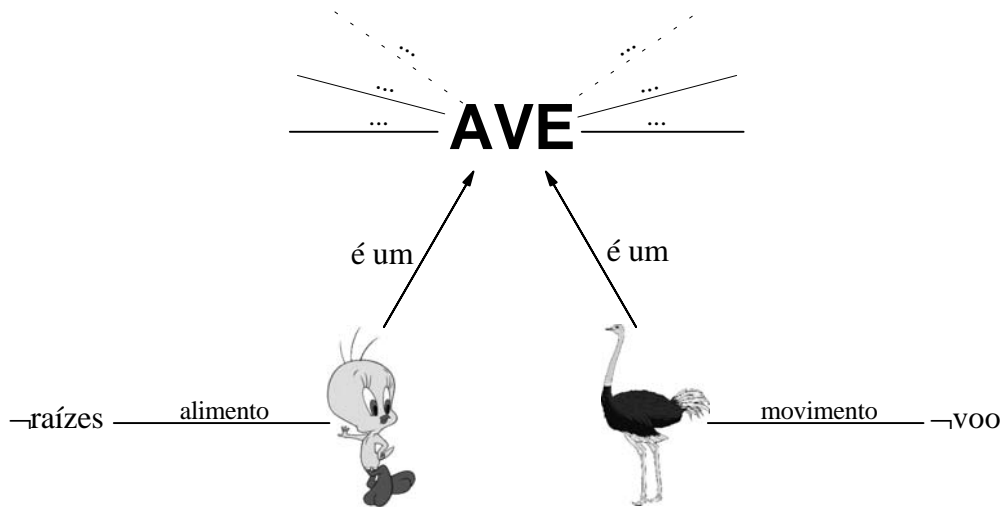


Figura 13: Propriedades do Tweety e das avestruzes

Esta figura descreve que o alimento do *Tweety* **não** são raízes e que a forma como as avestruzes se movimentam **não** é pelo voo. Estas circunstâncias indiciam a necessidade de se ter de considerar a inclusão de informação negativa, na definição do conjunto de propriedades dos agentes. Tal poderá ser concretizado alterando as definições dos agentes *tweety* e *avestruz*, do Programa 3. O agente *tweety* passará a ser definido por:

$$\text{agente}(\text{tweety}, [\neg \text{alimento}(\text{tweety}, \text{raízes})]) \quad (2)$$

e o agente *avestruz* definir-se-á como:

$$\text{agente}(\text{avestruz}, [\neg \text{movimento}(\text{avestruz}, \text{voo})]) \quad (3)$$

o que implica algumas alterações ao nível do predicado *prova* (Programa 4), em particular, considerando um terceiro argumento na definição do predicado *processar*, que permita conhecer o resultado (solução), do processamento da questão na lista de propriedades. Assim, o predicado *processar* será definido na forma:

$$\text{processar}: \text{Questão}, \text{Propriedades}, \text{Solução} \rightarrow \{ \mathbb{V}, \mathbb{F} \}$$

Nestes termos, a extensão do predicado `processar` será dada pelas cláusulas que se descrevem no Programa 5.

```
processar (Q, [Q|P], Q)
processar (Q, [¬Q|P], ¬Q)
processar (Q, [P1|P], S) ←
    processar (Q, P, S)
```

Programa 5: Extensão do predicado processar

A primeira cláusula do Programa 5 define que o resultado de `processar` a questão Q na lista de propriedades, terá como solução a própria questão Q , se for possível encontrar uma concretização dessa questão na lista de propriedades, $[Q|P]$ (em particular, à cabeça da lista). A segunda cláusula determina que o resultado de `processar` a questão Q terá como solução a sua negação, $\neg Q$, se for possível encontrar uma concretização para essa questão ao nível da informação negativa definida na lista de propriedades, $[\neg Q|P]$ (encontrada, mais uma vez, à cabeçada lista). A última cláusula implementa o processo recursivo de procura da solução S para a questão Q , na cauda da lista de propriedades, P , por não ter sido encontrada solução ao nível da propriedade P_1 , à cabeça da lista de propriedades.

As considerações feitas sobre o cálculo do resultado do processamento de questões são válidas, também, se enquadradas no âmbito do predicado `prova`. Será útil conhecer a solução específica que resultou da prova de uma questão do que saber, somente, qual o valor de verdade resultante da demonstração. Significa isto que se pode tirar proveito dessa situação, se for redefinido o predicado `prova`, nos mesmos termos em que se definiu o predicado `processar`.

Nesta linha de pensamento, será necessário reformular a extensão do predicado `prova`, apresentado no Programa 4. Essa reformulação é dada pelo Programa 6. A alteração passa pela introdução de um terceiro argumento no

predicado *prova*, com a finalidade de permitir conhecer a solução *S*, na demonstração da questão *Q*, utilizando o conhecimento do agente *A*.

```

prova(A, Q, S) ←
    agente(A, P) ∧
    processar(Q, P, S)
prova(A, Q, S) ←
    é um(A, C) ∧
    prova(C, Q, S)
¬prova(A, Q, S) ←
    não prova(A, Q, S)

```

Programa 6: Reformulação da extensão do predicado prova

Até ao momento, o cuidado tem sido colocado na demonstração de um teorema, ou, dito de outra forma, na procura de uma solução para uma dada questão.

O programa em lógica que descreve a informação da Figura 13 é o que se obtém a partir das alterações enunciadas pelas produções dadas em (2) e (3) no Programa 3. Nestas condições, a menos que se imponha que a verificação de uma determinada propriedade num certo nível da estrutura, impede a continuação da procura de soluções alternativas para a questão enunciada, pode acontecer que seja possível encontrar soluções contraditórias, em função do nível da estrutura em que se encontrar o agente cuja informação tenha produzido esses resultados.

Para o exemplo mencionado e que se reproduz no Programa 7, é possível determinar que a avestruz não voa, através da informação registada ao nível do agente avestruz, e é possível concluir que a avestruz voa, permitindo a herança dessa propriedade definida ao nível do agente ave. Como no Programa 6 não existe qualquer mecanismo que impeça a herança neste tipo de situação, é possível obter as duas soluções referidas, a partir do conhecimento descrito pelo Programa 7.


```

agente (ave, [postura(ave, ovos) ,
              movimento(ave, voo) ,
              cobertura(ave, penas) ,
              alimento(ave, minhocas) ,
              alimento(ave, raízes) ] )
agente (tweety, [¬alimento(ave, raízes) ] )
agente (avestruz, [¬movimento(ave, voo) ] )

é um(tweety, ave)
é um(avestruz, ave)

```

Programa 7: Informação complementar sobre os agentes

Este tipo de situações, em que a herança de propriedades, representando informação mais genérica (obtida a partir de níveis superiores na estrutura de conhecimento), pode vir a contradizer conclusões obtidas a partir de informação mais específica (descrita a níveis inferiores da hierarquia), pode ser abordada, numa primeira fase, incluindo uma forma de restrição à herança de propriedades, se essa informação já existir ao nível do agente inquirido (ou em qualquer nível no qual tenha sido obtida uma solução para a questão dada).

Na segunda cláusula do Programa 6, cláusula responsável pela implementação do mecanismo de herança na estrutura de conhecimento em utilização, será necessário garantir que a demonstração da questão Q só “avançará” para o nível da classe C do agente A , se ao nível desse agente não existir a informação necessária:

```

prova(A, Q, S) ←
    agente(A, P) ∧
    não existe(Q, P) ∧
    é um(A, C) ∧
    prova(C, Q, S)

```

O predicado *existe*, que surge na expressão em epígrafe, implementa um mecanismo de procura de uma possibilidade de prova da questão *Q* na lista de propriedades *P*, semelhante à descrita para o predicado *processar*, referido no Programa 5. No entanto, a sua funcionalidade não é de processamento da questão, mas, meramente, de verificação da existência de possibilidade de solucionar a questão *Q* ao nível das propriedades *P* do agente *A*.

Contudo, esta aproximação traz mais problemas do que aqueles que procura solucionar. Esta é uma abordagem que propõe uma solução para um problema específico, induzido pela Figura 13. O que acontecerá se a generalidade das situações não apresentarem essas características?

Considere-se o universo representado pela Figura 8, no sentido em que representa uma estrutura hierárquica de conhecimento. Nesse contexto, e uma vez que não existe informação em termos da entidade vampiro que caracterize a forma de movimento, não seria cancelada a herança dessa propriedade a partir de nenhuma das suas classes, mesmo sendo contraditórias.

Podem-se considerar, ainda, outros tipos de problemas, que não são resolvidos por esta abordagem e que são dados, mais uma vez, pela Figura 8, devido aos problemas que envolvem a herança múltipla. No caso do vampiro, seria conveniente permitir a herança, em particular, da propriedade que descreve a sua forma de movimento, através do ramo que estabelece a ligação às aves, mas não o permitir relativamente à ligação que estabelece a relação entre o vampiro e os animais mortos.

Este tipo de selecção do cancelamento da herança, não é o que está a ser

implementado pelo mecanismo descrito anteriormente. Contudo, este processo parece mais aceitável, no intuito de se conseguir generalizar o mecanismo de herança, não o limitando ao tratamento de situações particulares, como era o caso da primeira abordagem considerada.

Estes desenvolvimentos implicam modificações ao nível da representação de conhecimento na estrutura hierárquica. Como ficou implícito, o método a seguir implica o estabelecimento de restrições à herança que dependem, em particular, da ligação entre os agentes, ou seja, as restrições deverão ser contempladas e descritas ao nível do relacionamento implementado pelo predicado `é um`.

Voltando ao exemplo da Figura 13, pretende-se descrever, nessa estrutura de conhecimento, que a propriedade que concretiza a forma de movimento das aves-truzes não se possa herdar a partir da ligação que as relaciona com as aves (ver Figura 14). Qualquer outra ligação não será afectada por este condicionalismo.

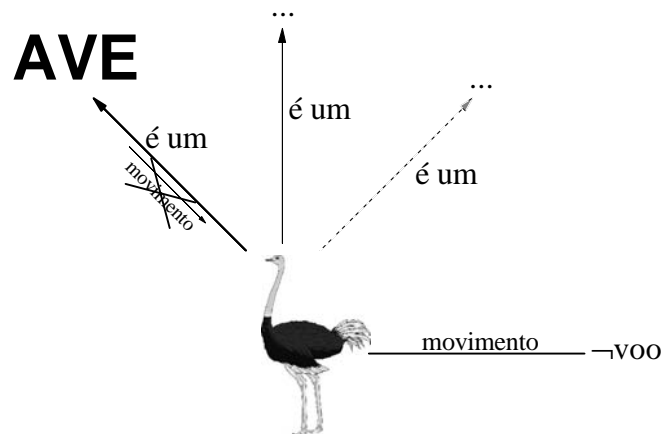


Figura 14: Cancelamento selectivo da herança

Dado que através de uma mesma ligação poderá ser cancelada a herança, por parte de uma subclasse, de propriedades da(s) classe(s) a que se reporta, o predicado `é um` deverá ser redefinido, na forma:

é um: SubClasse, Classe, Cancelar $\rightarrow \{ \mathbb{V}, \mathbb{F} \}$

através da inclusão de um terceiro argumento na estrutura definida em (1), representando a lista de propriedades a Cancelar no processo de herança. Para o exemplo dado pela Figura 14, tem-se que:

é um (avestruz, ave, [movimento(A,M)])

em que as variáveis A e M representam entidades abstractas no âmbito da cláusula em que surgem, por se pretender não permitir herdar qualquer forma de movimento que esteja definida em termos das aves. O caso do *Tweety* apresentado na Figura 13 é análogo ao das avestruzes, embora não totalmente idêntico.

O que se pretende apresentar através da Figura 13 é o facto de o *Tweety* não se alimentar, necessariamente, como as aves (Figura 12), no caso, não se alimentar de raízes. Significa isto que o cancelamento da herança não deve acontecer para qualquer valor da propriedade alimento, mas apenas no caso concreto de se tratar do alimento raízes:

é um (tweety, ave, [alimento(A,raízes)])

Obviamente que estas alterações produzidas ao nível da representação de conhecimento terão de ser levadas em conta em termos da redefinição da extensão do predicado prova, descrito pelo Programa 6. Tendo em conta que a alteração proposta na anterior reformulação da segunda cláusula do predicado prova não será aceite devido aos problemas já enumerados, e, considerando a modificação que foi produzida ao nível da representação de conhecimento, a segunda cláusula do Programa 6, tomará a forma:

```

prova(A, Q, S) ←
    é um(A, C, PC) ∧
    não existe(Q, PC) ∧
    prova(C, Q, S)

```

O predicado *existe*, já referenciado, limita-se a verificar a possibilidade de resolver a questão *Q* na lista de propriedades *PC* definidas para cancelar a herança.

Basicamente, a diferença entre esta e a anterior definição da segunda cláusula do predicado *prova* está em que, nesta situação, o cancelamento da herança é feito pela inclusão explícita de uma lista de propriedades com essa finalidade, na definição da relação entre dois agentes, enquanto que na aproximação inicial, tornava-se implícita a enumeração das propriedades a cancelar, sugerindo que fossem canceladas, apenas (ou todas), aquelas que se encontrassem definidas ao nível do agente. Resumem-se, estas modificações, à redefinição da extensão do predicado *prova* nos termos descritos pelo Programa 8.

```

prova(A, Q, S) ←
    agente(A, P) ∧
    processar(Q, P, S)
prova(A, Q, S) ←
    é um(A, C, PC) ∧
    não existe(Q, PC) ∧
    prova(C, Q, S)
¬prova(A, Q, S) ←
    não prova(A, Q, S)

```

*Programa 8: Extensão do predicado *prova* com cancelamento de herança*

6.3 Representação de Informação Incompleta

No que respeita à representação de informação incompleta, o tema será abordado, em particular, no âmbito das estruturas de conhecimento em desenvolvimento, nos termos apresentados para a representação de valores nulos, tal como foram discutidos no capítulo 0.

Considere-se, novamente, a Figura 12, em que se caracterizam propriedades das aves. O Programa 2 representa essa informação em termos de um programa em lógica. No actual contexto e em termos do processo de representação de conhecimento, há que explicitar quando é que é falso a postura das aves e quais as excepções que se podem verificar, quando é falso o movimento das aves e quais as excepções que existem, e assim por diante. Estas considerações implicam que, a informação da Figura 12, seja dada na forma que se apresenta no Programa 9.

```

agente (ave, [postura(ave, ovos) ,
               ¬postura(ave, P) ←
                 não postura(ave, P) ∧
                 não excepçãopostura(ave, P) ,
               movimento(ave, voo) ,
               ¬movimento(ave, M) ←
                 ... ,
               ...] )

```

Programa 9: Propriedades do agente ave

É óbvio que existe a necessidade de considerar a descrição da informação que é, efectivamente, falsa, ao nível de todos os agentes da estrutura e que as excepções se podem encontrar em qualquer nível ou em qualquer agente. Deste modo, todos os agentes descritos nesta estrutura deverão comportar uma representação desta forma, explicitando as situações em que se deverá concluir sobre a falsidade das questões e eventuais excepções.

Estas situações vêm tornar mais natural a descrição da informação ao nível de cada agente. Cada propriedade exibida por cada agente, pode comportar várias extensões que a concretizem, como, por exemplo, o alimento das aves (ver Figura 12). Pode admitir, também, explicitamente, a inclusão de informação negativa, como no caso da avestruz cuja forma de movimento não é o voo (ver Figura 13). Nos termos já focados nesta secção, cada agente pode, ainda, incluir regras complementares de cálculo de propriedades.

Por estes motivos, torna-se necessário “rearranjar” a forma de descrever o conhecimento existente ao nível de cada agente. Assim, em vez de a caracterização de um agente ser feita através da enumeração directa de todas as concretizações de propriedades que possa incluir, essas concretizações serão agrupadas sob um identificador comum, o identificador da própria propriedade, associado, então, ao conjunto (ou lista) de situações concretas que o agente verifique. Essa associação será identificada pelo símbolo “::”:

$$\text{agente} \left(a, \left[p_1(a, X) :: \left[p_1(a, x_1), \dots, p_1(a, x_n), \right. \right. \right. \\ \left. \left. \left. \neg p_1(a, x_{n+1}), \dots, \neg p_1(a, x_{n+m}), \right. \right. \right. \\ \left. \left. \left. \neg p_1(a, X) \leftarrow \dots, \right. \right. \right. \\ \left. \left. \left. \dots \right] , \right. \right. \\ \left. \left. p_2(a, Y) :: [\dots] , \right. \right. \\ \left. \left. \dots \right] \right)$$

em que o a representa o agente concreto, os p_i denotam as propriedades que o agente exhibe, e os X e os Y identificam a concretização das propriedades do agente.

Nestes termos, a descrição do agente *ave* efectuada no Programa 9 será reformulada, passando a representar-se como se mostra no Programa 10.

$$\text{agente} \left(\text{ave}, \left[\text{postura}(\text{ave}, P) :: \right. \right. \\ \left. \left. [\text{postura}(\text{ave}, \text{ovos}), \right. \right. \\ \left. \left. \dots \right] \right)$$

```

¬postura(ave,P) ←
    não postura(ave,P) ∧
    não excepçãopostura(ave,P) ],
movimento(ave,M)::
    [movimento(ave,voo) ,
    ¬movimento(ave,M) ←
        ...] ,
    ...] )

```

Programa 10: Rescrita das propriedades do agente ave

Naturalmente que todas estas alterações à forma de representação da informação ao nível dos agentes de conhecimento, implicam modificações na abordagem das questões que se possam colocar. Contudo, uma vez que estas alterações apenas se reflectem na organização do conhecimento do agente, as modificações necessárias apenas se repercutirão ao nível do processamento das questões, ou seja, no mecanismo implementado pelo predicado `processar`.

6.3.1 Nulos do Tipo Desconhecido

Por uma questão de continuidade, tentar-se-á manter os exemplos que foram equacionados em [AnNe96]. O exemplo considerado para abordar valores nulos do tipo desconhecido, foi o da existência de uma ave rara cujo canto seria o fado. Este exemplo pressupõe a existência de um agente que designará a entidade ave rara:

```
agente(ave rara, [])
```

Esta ave rara representa uma ave desconhecida que, à partida, não possui qualquer propriedade definida, ou seja, representa um agente do tipo desconhecido. Existe, contudo, a necessidade de declarar que as aves apresentam uma situação de excepção no que respeita à forma de canto, quando se trata de identificar a ave que canta o fado. Deste modo, e tendo em conta a forma como foi


```

agente (ave, [...,
               canto(ave, P)::
               [canto(ave rara, fado),
                ¬canto(ave, Som) ←
                  não canto(ave, Som) ∧
                  não excepçãocanto(ave, Som),
                excepçãocanto(ave, Som) ←
                  canto(ave rara, Som)] ,
               ...] )

```

Programa 11: Valor nulo do tipo desconhecido

abordado este tipo de valores nulos em [AnNe96], é necessário incluir a concretização desta situação, possibilitando o aparecimento de excepções para as aves:

```

¬canto(Ave, Som) ←
  não canto(Ave, Som) ∧
  não excepçãocanto(Ave, Som)

```

e especificando a excepção a ser considerada:

```

excepçãocanto(Ave, Som) ←
  canto(ave rara, Som)

```

sempre que for possível encontrar uma ave rara que cante o fado:

```

canto(ave rara, fado)

```

Como já se viu, esta informação dever-se-á encontrar representada ao nível da definição do agente ave. Esta situação encontra-se descrita no Programa 11.

Conforme se pode constatar pelo Programa 11, na definição de cada agente, existe um conjunto de informação (lista de propriedades) que caracteriza o seu domínio de conhecimento, formado por uma sequência de identificadores aos quais se associam os valores específicos das propriedades que representam.

Considere-se uma base de conhecimento composta pela definição do agente ave, apresentada no Programa 11, juntamente com a restante informação definida para os outros agentes presentes na estrutura, conforme se mostra na compilação da informação que se faz no Programa 12.

Perante a informação descrita pelo Programa 12, pode-se questionar o sistema por forma a averiguar quais as soluções propostas para o problema. Assim, quando colocada a questão:

$\text{prova}(\text{tweety}, \text{canto}(\text{tweety}, \text{fado}), \text{Solução}) ?^4$

a resposta será desconhecida e a Solução não será concretizada em virtude dessa resposta. Isso acontece porque, em primeiro lugar, não existe informação ao nível do *tweety* respeitante à sua forma de canto, e, em segundo lugar, porque se encontra informação no nível hierárquico superior, correspondente às aves, que define uma excepção em relação à ave que canta o fado, que não é conhecida.

Para a questão:

$\text{prova}(\text{Agente}, \text{canto}(\text{Agente}, \text{fado}), \text{Solução}) ?$

a única interpretação que produz uma resposta verdadeira é a que associa ao Agente o valor nulo ave rara, caso em que a Solução identifica que o canto da ave rara é o fado. Neste caso, é possível provar que existe solução para a questão $\text{canto}(\text{Agente}, \text{fado})$. O que fica por conhecer é a ave concreta que valida essa solução, uma vez que o termo ave rara corresponde a um valor nulo e, por conseguinte, identifica uma ave desconhecida.

⁴ Notar que as questões deverão estar a ser colocadas em termos da programação em lógica estendida, permitindo respostas em termos da sua veracidade, falsidade ou desconhecimento de solução. Significa isto que se está a supor a aplicação implícita dos pressupostos implementados no predicado apresentado em [AnNe96]. No entanto, por motivo de manter uma certa simplicidade na notação utilizada para a colocação das questões, manter-se-ão esses pressupostos implícitos, não se lhes fazendo referência directa na colocação das questões, mas aplicando o raciocínio subjacente a esse tipo de sistema na obtenção das respostas.

```

agente (ave, [...,
    canto(ave,P)::
        [canto(ave rara, fado),
         ¬canto(ave, Som) ←
             não canto(ave, Som) ∧
             não excepçãocanto(ave, Som),
         excepçãocanto(ave, Som) ←
             canto(ave rara, Som)] ,
    ...] )
agente (tweety, [alimento(tweety,A)::
    [¬alimento(tweety, raízes)] ] )
agente (avestruz, [movimento(avestruz,M)::
    [¬movimento(avestruz, voo)] ] )
agente (ave rara, []) )

é um (tweety, ave, [alimento(A, raízes)] )
é um (avestruz, ave, [movimento(A,M)] )
é um (ave rara, ave, []) )

```

Programa 12: Base de conhecimento, instante I

A discussão mantida até ao momento tem-se desenrolado em função da caracterização das propriedades dos agentes. Contudo, no presente exemplo, o valor nulo ave rara representa, em si, um agente desconhecido. Sendo assim, também a extensão do predicado agente, que tem sido utilizado para descrever os agentes presentes na estrutura de conhecimento, deverá ser considerada em termos da Programação em Lógica Estendida (PLE), isto é, há que complementar a extensão do predicado agente com:

$$\neg \text{agente}(\text{Agente}, \text{Propriedades}) \leftarrow \\ \text{não agente}(\text{Agente}, \text{Propriedades}) \wedge \\ \text{não exceção}_{\text{agente}}(\text{Agente}, \text{Propriedades})$$

ou seja, é falso que se considere a existência de um Agente se não existir uma prova para tal facto e se não existir uma excepção definida para o Agente.

Neste sentido, o valor nulo ave rara, concretiza uma excepção à definição de um agente, uma vez que representará um valor desconhecido. Desta forma, deverá ser incluída a excepção:

$$\text{excepção}_{\text{agente}}(\text{Agente}, \text{Propriedades}) \leftarrow \\ \text{agente}(\text{ave} \text{ rara}, \text{Propriedades})$$

juntamente com a informação já existente na base de conhecimento. A consequência imediata deste facto é que a extensão do predicado prova (Programa 8), deverá ser rescrita por forma a considerar este tipo de situação. Em termos da primeira cláusula do predicado prova não é suficiente considerar que a prova da questão está condicionada à existência de informação definida para o predicado agente. O Programa 8 tem de ser re-equacionado em termos da PLE. Deste modo, qualquer resposta é dada em termos do Programa 9 de [AnNe96].

Tendo em conta o exposto, a colocação da questão:

$$\text{prova}(\text{coruja}, \text{canto}(\text{coruja}, \text{fado}), \text{Solução}) ?$$

teria uma resposta do tipo desconhecido, uma vez que, apesar de não existir nenhum agente que concretize a existência da coruja, essa poderá materializar uma possível solução, em virtude de poder ser considerada como uma das excepções às aves que cantam o fado, isto é, desconhece-se se é a coruja que concretiza a ave rara que canta o fado.

```

agente (ave, [...,
    canto(ave, P)::
        [canto(ave rara, fado),
         ¬canto(ave, Som) ←
             não canto(ave, Som) ∧
             não exceçãocanto(ave, Som),
         exceçãocanto(ave, Som) ←
             canto(ave rara, Som)] ,
    ...] )
agente (tweety, [alimento(tweety, A)::
    [¬alimento(tweety, raízes)] ] )
agente (avestruz, [movimento(avestruz, M)::
    [¬movimento(avestruz, voo)] ] )
agente (ave rara, []) )

¬agente (Agente, Propriedades) ←
    não agente(Agente, Propriedades) ∧
    não exceçãoagente(Agente, Propriedades)

exceçãoagente(Agente, Propriedades) ←
    agente(ave rara, Propriedades)

é um (tweety, ave, [alimento(A, raízes)] )
é um (avestruz, ave, [movimento(A, M)] )
é um (ave rara, ave, []) )

¬é um (Agente, SuperClasse, Cancelar) ←
    não é um(Agente, SuperClasse, Cancelar) ∧
    não exceçãoé um(Agente, SuperClasse, Cancelar)

```

Programa 13: Base de conhecimento, instante II

Da mesma forma que se abordou a extensão do predicado agente, pode-se discutir a extensão do predicado é um, ou seja, também para este predicado deverá ser considerada a possibilidade de incluir excepções, descrevendo-o em termos da PLE, complementando-o com:

$$\begin{aligned} \neg \text{é um}(\text{Agente}, \text{SuperClasse}, \text{Cancelar}) \leftarrow \\ \text{não é um}(\text{Agente}, \text{SuperClasse}, \text{Cancelar}) \wedge \\ \text{não excepção}_{\text{é um}}(\text{Agente}, \text{SuperClasse}, \text{Cancelar}) \end{aligned}$$

A base de conhecimento será composta, então, do modo como se descreve, de forma resumida, no Programa 13.

6.3.2 Nulos do Tipo Desconhecido, de um Conjunto de Valores

No que diz respeito à representação de valores nulos do tipo desconhecido, mas pertencentes a um conjunto determinado de valores, foi utilizado o exemplo do pinguim que era uma ave que cantava uma de duas coisas: ópera ou música clássica. Segundo a abordagem efectuada, esta situação corresponde a uma disjunção de observações, que se contemplam através da inclusão de excepções explícitas à forma de canto para cada uma das situações referidas:

$$\begin{aligned} \text{excepção}_{\text{canto}}(\text{pinguim}, \underline{\text{ópera}}) \\ \text{excepção}_{\text{canto}}(\text{pinguim}, \underline{\text{clássica}}) \end{aligned}$$

Da mesma forma que para o caso anterior, de valores nulos do tipo desconhecido, esta é informação que diz respeito à caracterização das aves, pelo que deverá ser incluída ao nível do agente ave:

$$\begin{aligned} \text{agente}(\text{ave}, [\dots, \\ \text{canto}(\text{ave}, P) :: \\ [\dots, \\ \text{excepção}_{\text{canto}}(\text{pinguim}, \underline{\text{ópera}}), \\ \text{excepção}_{\text{canto}}(\text{pinguim}, \underline{\text{clássica}})], \\ \dots]) \end{aligned}$$

passando a complementar a definição do agente ave apresentada no Programa 13.

De igual modo, o agente pinguim deverá passar a ter existência:

```
agente (pinguim, [])
```

sabendo tratar-se de uma ave:

```
é um (pinguim, ave, [])
```

Tal como seria de esperar, tanto a questão:

```
prova (pinguim, canto (pinguim, ópera) , Solução) ?
```

como a questão:

```
prova (pinguim, canto (pinguim, clássica) , Solução) ?
```

não terão resposta concreta, uma vez que não se conhece, especificamente, qual a forma de canto do pinguim: será a ópera ou a música clássica, embora não se saiba, concretamente, qual.

Já quanto à questão:

```
prova (avestruz, canto (avestruz, ópera) , Solução) ?
```

a resposta será falsa, porque as exceções definidas não dizem respeito ao agente avestruz, pelo que a Solução não se encontrará concretizada.

Também para este tipo de valores nulos se podem considerar exceções em relação à definição da estrutura do conhecimento.

Considere-se a estrutura hierárquica representada na rede semântica da Figura 8. Esta figura ilustra um problema comum em estruturas hierárquicas, que surge quando se permite a herança múltipla, isto é, quando uma entidade da estrutura pode herdar propriedades de mais do que uma superclasse, sendo, eventualmente, contraditórias. Sujeite-se a Figura 8 a uma ligeira alteração, como se ilustra na Figura 15.

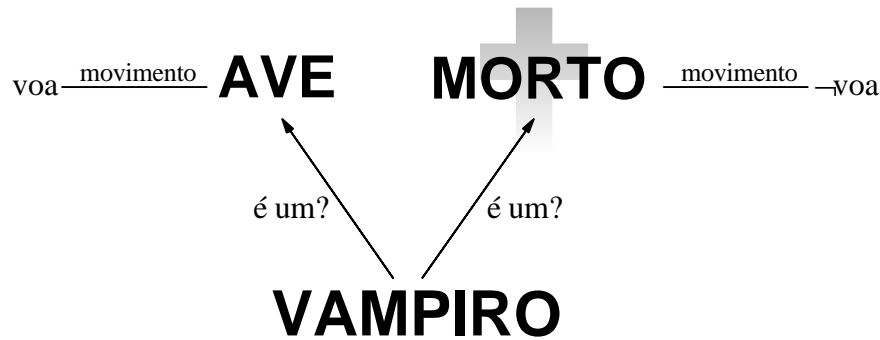


Figura 15: Herança múltipla através de uma relação desconhecida

Com a Figura 15 pretende-se ilustrar que o vampiro pode ser uma ave ou um animal morto, desconhecendo-se, no concreto, de que situação se trata. O objectivo desta figura é o de ilustrar um valor nulo do tipo desconhecido, mas de um conjunto determinado de valores, que ocorre ao nível da definição da estrutura de conhecimento.

Esta situação pode ser descrita, através da forma de representação de valores nulos do tipo em discussão, como excepções à extensão da relação implementada pelo predicado `é um`:

```
excepçãoé um (vampiro,ave, [])
excepçãoé um (vampiro,morto, [])
```

assumindo-se que, o predicado `é um`, é dado pelo Programa 13. Obviamente que, a par desta definição, seria necessário incluir dois novos agentes: o agente `vampiro` e o agente `morto`. A base de conhecimento resultante é a que se mostra no Programa 14.

Através desta abordagem ao problema da herança múltipla, a situação foi solucionada recorrendo à utilização de valores nulos para representar a estrutura, correspondendo à forma como foi apresentado o exemplo. Qualquer solução que seja obtida através dos ramos definidos pela Figura 15 será do tipo desconhecido.


```

agente (ave, [...,
    canto (ave, P) ::
        [canto (ave rara, fado),
         ¬canto (ave, Som) ←
             não canto (ave, Som) ∧
             não excepçãocanto (ave, Som),
         excepçãocanto (ave, Som) ←
             canto (ave rara, Som),
         excepçãocanto (pinguim, ópera),
         excepçãocanto (pinguim, clássica)] ,
    ...] )
agente (tweety, [...] )
agente (avestruz, [...] )
agente (ave rara, [...] )
agente (pinguim, [...] )
agente (vampiro, [...] )
agente (morto, [...] )
¬agente (Agente, Propriedades) ←
    não agente (Agente, Propriedades) ∧
    não excepçãoagente (Agente, Propriedades)
excepçãoagente (Agente, Propriedades) ←
    agente (ave rara, Propriedades)

é um (tweety, ave, [...] )
é um (avestruz, ave, [...] )
é um (ave rara, ave, [...] )
¬é um (Agente, SuperClasse, Cancelar) ←
    não é um (Agente, SuperClasse, Cancelar) ∧
    não excepçãoé um (Agente, SuperClasse, Cancelar)
excepçãoé um (vampiro, ave, [])
excepçãoé um (vampiro, morto, [])

```

Programa 14: Base de conhecimento, instante III

Da mesma forma que aconteceu para o exemplo dos agentes desconhecidos, abordados na subsecção anterior, onde se utilizou o valor nulo do tipo desconhecido, através da ave rara, também para este exemplo, o predicado prova (Programa 8), deverá considerar os pressupostos descritos pelo no Programa 9 de [AnNe96], para a obtenção das soluções em termos da PLE.

6.3.3 Nulos do Tipo Não Permitido

Relativamente aos valores nulos do tipo não permitido, conforme já foi referido em [AnNe96], o seu significado estático corresponde aos valores nulos do tipo desconhecido, embora tal situação não consiga capturar o significado dinâmico que se pretende atribuir a este tipo de valores nulos.

O exemplo considerado foi o da existência de uma ave ω cujo canto era o hino, mas que se desconhecia de que ave concreta se tratava. Simultaneamente, qualquer tentativa de inclusão de informação que concretizasse essa ave deveria ser rejeitada, uma vez que também não se pretende admitir que seja conhecida.

A representação deste tipo de valor nulo é feita como se se tratasse de um valor nulo do tipo desconhecido, incluindo essa informação, mais uma vez, ao nível do conhecimento definido para o agente ave:

```

agente (ave, [... ,
               canto(ave, P) ::
                 [... ,
                  canto( $\omega$ , hino) ,
                  ... ,
                  exceçãocanto(ave, Som) ←
                    canto( $\omega$ , Som) ] ,
               ...] )

```

e, ainda, a descrição da existência do agente desconhecido, ω :

agente (ω , [])

sabendo-se que esse agente desconhecido é uma ave:

é um (ω , ave, [])

No entanto, a inclusão da restrição a verificar na assimilação de novo conhecimento, deverá estar fora do âmbito do domínio de conhecimento do agente ave, uma vez que não se pretende permitir a inclusão de informação que a viole em qualquer nível da estrutura, e não apenas quando isso disser respeito, directamente, às aves.

Assim, a restrição:

$\leftarrow \text{canto}(\text{Ave}, \text{hino}) \wedge \text{não nulo}(\text{Ave})$

e a identificação do valor nulo específico:

nulo (ω)

deverão estar definidos fora do âmbito do agente ave.

A interpretação de situações como a descrita é realizada com base nos mesmos pressupostos que se aplicaram para os valores nulos do tipo desconhecido, uma vez que a diferença se encontra no significado do valor nulo não permitido, significado esse que deve ser levado em conta em situações de assimilação de conhecimento.

A base de conhecimento onde se descreve esta informação é dada pelo Programa 15.

```

agente (ave, [... ,
    canto (ave, P) ::
        [canto (ave rara, fado) ,
         canto (ω, hino) ,
         ¬canto (ave, Som) ←
             não canto (ave, Som) ∧
             não excepçãocanto (ave, Som) ,
         excepçãocanto (ave, Som) ←
             canto (ave rara, Som) ,
         excepçãocanto (pinguim, ópera) ,
         excepçãocanto (pinguim, clássica) ,
         excepçãocanto (ave, Som) ←
             canto (ω, Som) ] ,
    ...] )
...
agente (ω, [])
¬agente (Agente, Propriedades) ←
    não agente (Agente, Propriedades) ∧
    não excepçãoagente (Agente, Propriedades)
excepçãoagente (Agente, Propriedades) ←
    agente (ave rara, Propriedades)
...
é um (ω, ave, [])
¬é um (Agente, SuperClasse, Cancelar) ←
    não é um (Agente, SuperClasse, Cancelar) ∧
    não excepçãoé um (Agente, SuperClasse, Cancelar)
excepçãoé um (vampiro, ave, [])
excepçãoé um (vampiro, morto, [])

← canto (Ave, hino) ∧ não nulo (Ave)

nulo (ω)

```

Programa 15: Base de conhecimento, instante IV

7 Resumo

Neste trabalho é feita uma abordagem à representação de conhecimento em estruturas hierárquicas, nas quais é possível tirar proveito de mecanismos de herança, no intuito de implementar formas de RNM, em particular, em situações caracterizadas por falta de informação.

Apresentam-se diversas estruturas de representação de conhecimento do tipo das que se pretendem adoptar, como forma de fundamentar e justificar a sua utilização no âmbito do trabalho.

Começa-se por abordar as características genéricas que se podem encontrar num grafo e a sua topologia básica, alicerçada em dois elementos: os arcos e os nodos.

Os enquadramentos são apresentados como uma forma de representação hierárquica de informação, habilitada para a manipulação de informação incompleta, mas cujo campo de aplicação se enquadra numa área algo diferente da que se pretende caracterizar.

No sentido de se chegar a uma estrutura hierárquica suficientemente genérica, abordam-se as redes semânticas, colocando-se particular ênfase na hierarquização de tais estruturas, realçando-se as suas capacidades para a representação de conhecimento e RNM, fazendo-se notar algumas das suas insuficiências e apontando-se certas desvantagens, que se encontram neste tipo de estruturas, especialmente ao nível da concretização de excepções e do tratamento de herança múltipla.

De seguida são abordadas, ainda, outras estruturas típicas de representação de conhecimento em termos hierárquicos, como sejam os guiões, os grafos conceptuais e as estruturas do tipo Objecto-Atributo-Valor.

No que respeita à estrutura hierárquica de representação de conhecimento

que se pretende desenvolver, começa-se por descrever a estrutura básica que se adopta para o desenvolvimento do sistema, abordando-se de seguida as questões levantadas pela implementação do RNM, construindo um sistema de inferência específico para a “*animação*” deste sistema. Na sequência desta aproximação, é abordada a temática da informação incompleta, no que respeita à representação de valores nulos, no âmbito da estrutura de conhecimento hierárquico desenvolvida.

A abordagem à representação de valores nulos é feita segundo dois vectores: em primeiro lugar representando valores nulos ao nível estrutura de conhecimento de cada agente e, em segundo lugar, introduzindo os valores nulos na estrutura de conhecimento do modelo hierárquico.

8 Referências

- [Ana96] Analide, C., 1996
“Representação de Conhecimento e Raciocínio em Estruturas Hierárquicas”,
Tese de Dissertação de Mestrado, Departamento de Informática, Universidade do Minho, Braga, Portugal
- [AnNe96] Analide, C., Neves, J., 1996
“Representação de Informação Incompleta”,
Relatório interno, Departamento de Informática, Universidade do Minho, Braga, Portugal
- [CoMe76] Costa, J. A., Melo, A. S., 1976
“Dicionário da Língua Portuguesa”,
5ª Edição, Porto Editora Lda., Porto, Portugal
- [Fil88] Filgueiras, M., 1988
“Linguagem Natural e Representação de Conhecimento”,
Texto preparado para a Escola Avançada de Inteligência Artificial, Associação Portuguesa Para a Inteligência Artificial
- [Hsl90] Hsleh, D., 1990
“A Logic to Unify Semantic Network Knowledge Systems with Object-Oriented Database Models”,
SRI International, CSL Technical Report

- [Min81] Minsky, M., 1981
“A Framework for Representing Knowledge”,
in *Mind Design*, Haugeland (ed.), pp. 95-128, Cambridge,
Massachusetts, The MIT Press
- [RiDu88] Ringland, G. A., Duce, D. A., 1988
“Approaches to Knowledge Representation: An Introduction”,
Research Studies Press Ltd, Taunton, Somerset, Inglaterra
- [RiKn91] Rich, E., Knight, K., 1991
“Artificial Intelligence – Second Edition”,
McGraw Hill, Inc., Nova Iorque, E.U.A.
- [Sow84] Sowa, J. F., 1984
“Conceptual Structures: Information Processing in Mind and
Machine”,
Addison-Wesley
- [Way91] Way, E.C., 1991
“Knowledge Representation and Metaphor”,
Kluwer Academic Publishers, Dordrecht, Holanda