UNIVERSIDADE TÉCNICA DE LISBOA

INSTITUTO SUPERIOR TÉCNICO

LEP: (meta-)Lógica de Especificação de Preferências*

João Manuel Pinheiro Cachopo
Aluno N$^{\underline{o}}$ 34085

Ana Margarida de Jesus Cardoso
Aluna N$^{\underline{o}}$ 34042

Setembro de 1994

Trabalho final de curso

da

Licenciatura em Engenharia Informática e de Computadores

Ramo de Inteligência Artificial

*Este trabalho foi parcialmente apoiado pelo projecto PBIC/TIT/1243/92 da Junta Nacional de Investigação Científica e Tecnológica (JNICT).

Trabalho realizado sob a orientação de

Maria dos Remédios Cravo

Professora Auxiliar do Departamento de Engenharia Mecânica

Instituto Superior Técnico

Agradecimentos

Gostaríamos de agradecer à Professora Maria dos Remédios Cravo, primeiro que tudo, por nos ter criado o interesse nas áreas de Raciocínio e de Revisão de Crenças, que nos levou à execução deste trabalho, bem como pelos inúmeros conhecimentos que nos transmitiu nestas áreas.

O nosso obrigado também pela paciência com que nos orientou, através de muitas leituras, revisões e comentários às várias versões pelas quais passou o trabalho, e pela compreensão para com as mudanças de ritmo de escrita do mesmo.

Gostaríamos também de agradecer ao Professor João Pavão Martins pelos seus comentários e sugestões para melhorar o conteúdo do trabalho.

Finalmente, a todos os membros do Grupo de Inteligência Artificial do I.S.T., o nosso agradecimento por nos proporcionarem um ambiente onde é agradável trabalhar.

Índice

1 Introdução

A capacidade de raciocínio é uma das características do ser humano que o distinguem como ser inteligente. Uma vez que a preocupação fundamental da Inteligência Artificial (IA) é criar programas que exibam comportamento inteligente, é natural que se tente modelar o raciocínio humano em computador.

Neste trabalho preocupamo-nos com um tipo de raciocínio vulgarmente designado por *raciocínio de senso comum*. Este tipo de raciocínio caracteriza-se por ser baseado em conhecimento do "mundo real", sempre em mudança, normalmente incompleto e contraditório. Por este motivo, ao tentarmos modelar em computador o raciocínio de senso comum, deparamo-nos com algumas dificuldades.

Normalmente, a informação de que dispomos sobre um determinado domínio é incompleta, porque raramente temos a possibilidade de ter toda a informação necessária para podermos tirar conclusões de uma forma absolutamente segura. Essa informação incompleta traduz-se em regras gerais, ou regras com excepções, do tipo "Normalmente, *alguma-coisa* verifica-se" — por exemplo, "Normalmente, a informação de que dispomos sobre um determinado domínio é incompleta". No entanto, continuamos a poder raciocinar sobre esse domínio *assumindo* que, se não temos informação em contrário, podemos concluir que "*alguma-coisa*" se verifica numa determinada situação. Se mais tarde nos chegar nova informação que contrarie "*alguma-coisa*", temos que rever essa conclusão, podendo eventualmente abandoná-la.

As *lógicas não-monótonas* surgiram para permitir raciocinar com base em informação incompleta. Um agente que use uma lógica não-monótona para guiar o seu raciocínio é capaz de saltar para conclusões que não são certas (do ponto de vista lógico), mas que são tão plausíveis que, na ausência de informação em contrário, se justifica acreditar nelas. Esta possibilidade de saltar para conclusões que não são certas leva ao aparecimento de *múltiplas extensões*, correspondentes a conjuntos aceitáveis de crenças dada a informação disponível. As múltiplas extensões aparecem quando existem regras gerais que se podem aplicar dada a informação disponível, e que sugerem conclusões contraditórias entre si.

Neste caso, o agente pode ter que escolher uma de entre as várias extensões existentes. Por exemplo, suponhamos que o agente tem a seguinte informação: normalmente os Quakers são pacifistas, normalmente os Republicanos não são pacifistas e o Nixon é Quaker e é Republicano. Com base nesta informação, o agente tem motivos para acreditar que o Nixon é pacifista e também para acreditar que não é pacifista, dando origem a duas extensões. Cada extensão contém apenas uma destas crenças.

Estando a raciocinar num mundo em mudança, as crenças de um agente devem sofrer alterações de forma a reflectir as mudanças que ocorrem no mundo. As *teorias de revisão de crenças* definem as alterações a fazer de forma a incluir ou excluir determinada crença. No entanto, a inclusão ou exclusão de uma crença num conjunto de crenças pode ser efectuada de diversas formas, algumas das quais são mais racionais do que outras. As várias teorias de revisão de crenças seguem alguns critérios gerais de racionalidade na definição das operações de mudança sobre as crenças. Um desses critérios é o *critério da mudança mínima*, segundo o qual as alterações a efectuar nas crenças de um agente devem ser mínimas. Apesar destes critérios gerais de racionalidade, continuam a existir, normalmente, várias alternativas possíveis para efectuar uma alteração. Assim, nestas teorias é comum existir uma *ordem de preferência* entre as crenças, relacionando a sua importância e que, quando há necessidade de abandonar crenças, permite abandonar as que são menos importantes. As preferências correspondem a conhecimento adicional que um agente tem acerca das suas crenças. A motivação para este trabalho é a de que esse conhecimento pode e deve ser representado, tal como o restante conhecimento que um agente tem acerca do mundo. No caso dos formalismos não-monótonos, em que existem várias extensões devido a um conflito entre regras gerais, as preferências podem também servir para escolher as extensões preferidas por um agente.

A realização destas alterações sobre as crenças de um agente, num sistema computacional tem que levar em linha de conta alguns aspectos:

- É necessário propagar eficientemente as alterações numa crença a todas as crenças que dependem dela — as crenças de um agente resultam não só da informação vinda do exterior, mas também das deduções feitas pelo agente a partir de outras crenças. Assim, quando uma crença sofre alterações, todas as crenças que foram deduzidas a partir dela podem ter que sofrer alterações também. Computacionalmente, não é viável eliminar todas as deduções efectuadas anteriormente, sendo necessário utilizar outros mecanismos para que a propagação das alterações às restantes crenças seja feita de forma eficiente.

- É preciso detectar conjuntos de crenças inconsistentes — os agentes que utilizam lógicas clássicas, quando as suas crenças são inconsistentes, permitem deduzir qualquer crença, deixando assim de ter utilidade. Por este motivo, é essencial que as situações em que as crenças do agente são inconsistentes sejam detectadas e identificadas as crenças na origem dessa contradição, para que se possa resolver a contradição e continuar com um conjunto de crenças consistente.

Os *sistemas de revisão de crenças* (SRC) foram criados para resolver estes problemas computacionais. Existem dois tipos de sistemas de revisão de crenças:

- Os sistemas do tipo ATMS,[1] baseados em suposições, que associam a cada fórmula as suposições (crenças básicas ou não derivadas) que estão na base da sua derivação. Em [Martins 1983] e [de Kleer 1986] são apresentados os primeiros sistemas do tipo ATMS.
- Os sistemas do tipo JTMS,[2] baseados em justificações, que associam a cada fórmula as fórmulas directamente utilizadas para a derivar. O TMS de Doyle, [Doyle 1979], deu início à área de revisão de crenças em IA e é um sistema do tipo JTMS.

Repare-se que não é tarefa destes sistemas determinar qual deve ser a alteração a fazer nas crenças do agente quando existe uma contradição, mas apenas detectar a contradição e quais as crenças na sua origem. A alteração a fazer nas crenças é determinada pelas teorias de revisão de crenças.

Este trabalho assenta numa formalização de raciocínio de senso comum que é composta por uma lógica não-monótona — a lógica SWMC de [Cravo 1993*b*] — e por uma teoria de revisão de crenças descrita em [Cravo 1993*a*]. Com base nestes dois trabalhos existe um sistema computacional capaz de efectuar raciocínio de senso comum, o SNePSwD [Cravo & Martins 1993].

O nosso objectivo é definir uma linguagem que permita especificar as ordens de preferência entre as várias crenças de uma forma que seja simples e intuitiva, mas ao mesmo tempo suficientemente poderosa para permitir exprimir vários tipos de preferências.

Neste trabalho tentamos modelar um agente cognitivo capaz de efectuar raciocínio de senso comum, representando explicitamente as preferências entre as suas crenças e raciocinando sobre elas. A nossa proposta consiste numa arquitectura de dois níveis, separando assim o conhecimento acerca do "mundo real" do conhecimento (preferências) acerca desse conhecimento do agente.

Para reduzir ao mínimo a necessidade de consultar outra bibliografia, vamos descrever sucintamente a lógica, a teoria de revisão de crenças e o sistema de revisão de crenças nas secções seguintes. Depois, apresentaremos a linguagem de especificação de preferências, as suas características e alguns dos aspectos mais importantes da sua implementação. Finalmente, apresentaremos alguns exemplos de utilizações possíveis desta linguagem.

[1] Do inglês *Assumption-based Truth Maintenance System.*
[2] Do inglês *Justification-based Truth Maintenance System.*

2 A lógica SWMC

A SWMC é uma lógica não-monótona que foi desenvolvida para apoiar um sistema de revisão de crenças do tipo ATMS.

Ao longo desta secção vamos fazer um breve resumo dos conceitos mais importantes desta lógica. Uma descrição completa pode ser encontrada em [Cravo 1993*b*].

2.1 A linguagem — fórmulas bem formadas

Nesta secção vamos descrever a linguagem da lógica SWMC. Ao conjunto de todas as *fórmulas bem formadas* (fbfs) da SWMC, damos o nome de $\mathcal{L}$. Este conjunto de fbfs está particionado em vários sub-conjuntos disjuntos:

- O *conjunto das fórmulas da lógica de primeira ordem* (LPO), representado por $\mathcal{L}_{FOL}$.[3] Assumimos as regras de formação da LPO para as conectivas $\neg$, $\wedge$, $\vee$ e $\rightarrow$, e para os quantificadores $\forall$ e $\exists$. Estas fbfs vão representar as crenças do agente, como por exemplo, a fbf $Pinguim(Tweety)$, para representar o facto de que o Tweety é um pinguim.

- O *conjunto das regras de omissão*, representado por $\mathcal{L}_D$.[4] Para estas regras, é preciso um novo quantificador, o quantificador de omissão, representado pelo símbolo "$\bigtriangledown$", e a regra de formação é a seguinte: se $A(x)$[5] $\in \mathcal{L}_{FOL}$, então $\bigtriangledown(x)A(x)$ é uma fbf. Fbfs deste tipo são chamadas *regras de omissão*, e servem para representar regras gerais ou com excepções. As regras de omissão são sugestões de como o agente pode estender as suas crenças, mas não são crenças do agente, logo não se pode dizer que uma regra de omissão é acreditada ou não. Por exemplo, $\bigtriangledown(x)Ave(x) \rightarrow Voa(x)$ é uma regra de omissão que representa o conhecimento de que normalmente as aves voam.

- O *conjunto das suposições*,[6] representado por $\mathcal{L}_A$.[7] Se $D \in \mathcal{L}_D$, e c é um símbolo individual, então $Applicable(D, c)$ é uma fbf que representa a suposição de que a regra de omissão D é aplicável ao símbolo individual c. Por exemplo, se tivermos a regra $D = \bigtriangledown(x)Ave(x) \rightarrow Voa(x)$ e $Tweety$ for um símbolo da linguagem, então

[3] Do inglês *First Order Logic*.
[4] Do inglês *Default*.
[5] $A(x)$ significa qualquer fbf da LPO cuja variável livre seja x.
[6] O termo "suposições" tem um significado diferente daquele utilizado na nomenclatura de sistemas de revisão de crenças. A SWMC tem mecanismos de suporte a um ATMS, mas o que corresponde às suposições do ATMS são as fórmulas não derivadas ou hipóteses da SWMC.
[7] Do inglês *Assumption*.

$Applicable(D, Tweety)$ é uma suposição. As suposições são necessárias para se saber de que fórmulas depende uma conclusão obtida através da aplicação de uma regra de omissão: depende não só da regra de omissão utilizada, mas também da suposição de que a regra é aplicável àquela instância em particular.

- O *conjunto das excepções*, representado por $\mathcal{L}_E$.[8] Se $D \in \mathcal{L}_D$, $E(x) \in \mathcal{L}_{FOL}$, então $\forall(x)E(x) \rightarrow \neg Applicable(D, x)$ é uma excepção. Estas regras são usadas para exprimir excepções às regras de omissão, bloqueando a sua aplicação. Seguindo o exemplo anterior, temos que $\forall(x)Pinguim(x) \rightarrow \neg Applicable(\bigtriangledown(x)Ave(x) \rightarrow Voa(x), x)$ é uma excepção, que significa que os pinguins são uma excepção à regra de que "normalmente as aves voam".

As regras de formação das fbfs dos conjuntos $\mathcal{L}_D$, $\mathcal{L}_A$ e $\mathcal{L}_E$ podem ser estendidas para uma sequência de variáveis $\overline{x}$. Apresentamos aqui a versão mais simples, sem perda de generalidade, para facilitar a compreensão dos aspectos fundamentais da lógica SWMC. A versão estendida pode ser encontrada em [Cravo 1993*b*].

2.2 As fbfs suportadas — registo de dependências

A SWMC mantém um registo de dependências entre as fórmulas, o que a torna adequada para apoiar um sistema de revisão de crenças do tipo ATMS. Para associar a cada fórmula as fórmulas de que ela depende, a SWMC utiliza *fbfs suportadas*. Uma fbf suportada é da forma $<A, \tau, \alpha>$, onde $A \in \mathcal{L}$ é uma fbf, $\tau \in \{hyp, asp, der\}$ é um rótulo de origem, e $\alpha \subset \mathcal{L}$ é um conjunto de origem. Consideramos definidas as funções wff, ot e os, cujo domínio é o conjunto das fbfs suportadas:[9] $wff(<A, \tau, \alpha>) = A$; $ot(<A, \tau, \alpha>) = \tau$ e $os(<A, \tau, \alpha>) = \alpha$. A fbf suportada $<A, \tau, \alpha>$ corresponde a uma derivação particular da fbf A. Nesta fbf suportada:

- τ, o rótulo de origem, indica como é que a fbf foi gerada: hyp identifica hipóteses, asp identifica suposições, e der identifica fbfs derivadas.[10]

- α, o conjunto de origem, contém as hipóteses e/ou suposições que foram usadas numa derivação particular de A.

[8] Do inglês *Exception*.
[9] Do inglês *well-formed formula* (wff); *origin tag* (ot); *origin set* (os).
[10] Do inglês *hypothesis* (hyp); *assumption* (asp); *derived* (der).

O registo de dependências pode servir para distinguir entre os vários tipos de crenças que o agente pode ter: por um lado, pode ter crenças certas, porque foram derivadas sem usar regras de omissão; por outro lado, pode ter crenças plausíveis, em que acredita "condicionalmente", isto é, apenas se não tiver informação em contrário. Estas crenças plausíveis são representadas por fórmulas em cuja derivação foram usadas uma ou mais regras de omissão, o que pode ser observado a partir do conjunto de origem das fbfs suportadas que lhes correspondem. Vamos voltar a este assunto na secção 2.4, onde explicaremos em que condições é que se podem classificar as fbfs que representam as crenças como sólidas ou não.

2.3 As regras de inferência

As regras de inferência da SWMC definem como se podem derivar fbfs suportadas a partir de outras fbfs suportadas: caracterizam a fbf derivada, o seu rótulo de origem e o seu conjunto de origem a partir dos elementos correspondentes das fbfs suportadas que são as premissas da regra.[11] Há dois tipos de regras de inferência em SWMC:

- As *regras de inferência clássicas*, que são semelhantes às de um sistema de dedução natural para a Lógica de Primeira Ordem (LPO), tendo sido modificadas apenas para poderem lidar com fbfs suportadas.

- A *regra de inferência estendida*, que serve para permitir o raciocínio por omissão. O raciocínio por omissão caracteriza-se por utilizar regras (as regras de omissão) que não são universalmente verdadeiras porque têm excepções. A regra de inferência estendida é:

  Suposição e eliminação do quantificador de omissão (Sup-E$\bigtriangledown$)
  A partir de $<\bigtriangledown(x)A(x)$, hyp, $\{\bigtriangledown(x)A(x)\}>$, podemos inferir, para qualquer símbolo individual c, $<Applicable(\bigtriangledown(x)A(x), c)$, asp, $\alpha>$ e $<A(c)$, der, $\alpha>$, em que $\alpha = \{\bigtriangledown(x)A(x), Applicable(\bigtriangledown(x)A(x), c)\}$.

  Esta regra de inferência é semelhante à regra de eliminação do quantificador universal mas, uma vez que uma regra de omissão pode ter excepções, precisamos de assumir que o símbolo c utilizado não é uma excepção à regra, isto é, que a regra é aplicável a c.

As regras de inferência da lógica SWMC estão apresentadas no apêndice A.

[11] Designamos por *premissas* as fbfs suportadas utilizadas na aplicação de uma regra de inferência para deduzir outra fbf suportada.

2.4 A noção de consequência — contextos e espaços de crenças

Uma noção fundamental em todas as lógicas é a noção de consequência, definida entre um conjunto de fbfs e uma fbf. Esta noção, dada a informação que um agente dispõe, determina em que é que ele deve acreditar.

A informação de que um agente que utilize a SWMC dispõe é representada por um conjunto de hipóteses[12] designado por *contexto*. Assim, a noção de consequência em SWMC é definida entre um contexto e uma fbf.

O objectivo é, a partir de um contexto, chegar a todos os conjuntos de crenças em que é razoável um agente acreditar, designados por *espaços de crenças*. O conceito de espaço de crenças é fundamental para a definição de consequência. Por isso, primeiro vamos descrever como é que a partir de um contexto se chega aos espaços de crenças, e só depois damos a noção de consequência.

Um determinado contexto pode dar origem a mais do que um espaço de crenças (conjunto de fbfs da LPO), por regras de omissão diferentes "sugerirem" maneiras diferentes e contraditórias entre si de o agente estender as suas crenças; este facto corresponde à existência de *múltiplas extensões*[13] nas lógicas não-monótonas em geral. Em SWMC, para encontrar os espaços de crenças definidos por um contexto, existe um processo construtivo em quatro passos que parte do contexto e vai construindo estruturas intermédias até chegar aos espaços de crenças. No caso de o contexto ser inconsistente, não existe nenhum espaço de crenças definido por ele.

Uma noção auxiliar utilizada na construção dos espaços de crenças é a noção de derivabilidade:

> **Definição 1** Derivabilidade em SWMC
> Dado um conjunto de fbfs α e uma fbf A, diz-se que A é derivável a partir de α ($\alpha \vdash_{SWMC} A$), sse existe uma fbf suportada $<A, \tau, \gamma>$, com $\gamma \subseteq \alpha$.

Esta noção corresponde à noção de consequência da lógica clássica, mas, sendo monótona, não serve para capturar a noção de consequência em SWMC.

De seguida, vamos fazer a descrição intuitiva do processo de construção dos espaços de crenças a partir dos contextos; para uma descrição formal ver [Cravo 1993*b*].

O processo é constituido por quatro passos:

[12] Hipóteses são todas as fbfs introduzidas pela regra de Introdução da Hipótese: fbfs da LPO, regras de omissão ou excepções, ou seja, todas as fbfs da SWMC menos as suposições.

[13] Em SWMC, as extensões designam-se por espaços de crenças.

1. Estender o contexto com todas as suposições que individualmente são consistentes com o contexto inicial, obtendo-se assim o *contexto estendido*.

2. Dividir o contexto estendido em conjuntos máximos que sejam consistentes, os *núcleos primitivos*, que podem servir de base para a construção de um estado de crenças aceitável a partir das hipóteses do contexto. Este passo é necessário porque pode haver suposições ou pressupostos para a aplicação de regras de omissão que sejam contraditórios entre si.

3. Cada um dos núcleos primitivos é testado para ver se satisfaz determinados critérios de racionalidade, e os que não os satisfazem são eliminados. Os restantes são designados por *núcleos*. Os critérios de racionalidade utilizados podem ser encontrados em [Cravo 1993*b*].

4. Cada um dos núcleos define um *espaço de crenças*, que contém todas as fbfs da lógica de primeira ordem deriváveis em SWMC a partir do núcleo, e que corresponde a um estado de crença aceitável do agente.

Depois de apresentada a noção de espaço de crenças, podemos partir para o que nos propusémos no início desta secção: apresentar a noção de consequência em SWMC.

Em SWMC, dado um contexto β, existem três tipos distintos de consequências de β:

- *Consequência sólida* — dizemos que uma fbf A é uma consequência sólida de β, $\beta \vdash A$, se na derivação de A apenas foram usadas fbfs da LPO que estão no contexto β. Neste caso, a fbf A pertence a todos os espaços de crenças definidos por β. Esta noção de consequência é monótona, isto é, A é acreditada dado qualquer contexto consistente que contenha β.

- *Consequência plausível* — dizemos que uma fbf A é uma consequência plausível de β, $\beta \vdash_P A$, se na derivação de A foram eventualmente usadas uma ou mais suposições, e A pertence a todos os espaços de crenças definidos pelo contexto β. Intuitivamente esta situação corresponde a existirem razões para acreditar em A a partir de β e não existirem razões em contrário. Esta noção de consequência, ao contrário da anterior, é não-monótona, uma vez que, ao adicionarmos outras hipóteses ao contexto, A pode deixar de ser acreditada em alguns ou mesmo em todos os espaços de crenças.

- *Consequência concebível* — dizemos que A é uma consequência concebível de β, $\beta \vdash_C A$, se na derivação de A foram eventualmente usadas uma ou mais suposições e A pertence a pelo menos um espaço de crenças definido pelo contexto β. Intuitivamente, esta situação corresponde a existirem razões para acreditar em A mas também existirem razões para não acreditar.

A partir destas definições, é fácil ver que uma consequência sólida também é plausível e que por sua vez uma consequência plausível também é concebível.

A distinção entre os vários tipos de consequência é importante, pois embora o agente continue a conseguir estender as suas crenças com consequências que não são sólidas, por vezes é importante saber se está a raciocinar com base em informação que é sólida ou que é apenas plausível ou concebível.

Em [Cravo 1993*b*] é apresentada uma semântica baseada em conjuntos de modelos para a SWMC, e são provados os resultados de correcção e completude para a lógica com essa semântica.

3 A teoria de revisão de crenças

Na secção anterior apresentámos uma lógica que permite efectuar raciocínio não-monótono, isto é, saltar para conclusões baseadas em conhecimento que não é certo, e que mais tarde podem ter que ser abandonadas. Nesta secção vamos fazer um breve resumo da Teoria de Revisão de Crenças (TRC), apresentada em [Cravo 1993*a*], baseada na lógica SWMC.

Qualquer teoria de revisão de crenças tem por objectivo caracterizar as mudanças a fazer nas crenças de um agente, de modo a incluir ou excluir determinada crença. As mudanças a efectuar nas crenças são guiadas por alguns critérios de racionalidade, um dos quais é o de que as mudanças a efectuar devem ser mínimas.

Em [Gärdenfors 1988] são apresentadas três operações sobre as crenças de um agente (vistas como conjuntos de crenças fechados em relação à noção de consequência lógica): expansão, contracção e revisão. A TRC de [Cravo 1993*a*] define também três operações — adição, remoção e revisão — correspondendo às operações de Gärdenfors, mas que apresentam algumas diferenças pelo facto de considerar bases de crenças finitas (no sentido de [Nebel 1989]) e não conjuntos de crenças fechados em relação à noção de consequência lógica. Para além disso, a lógica subjacente não é a Lógica Proposicional, mas sim uma lógica não-monótona, o que traz algumas alterações na definição das operações que serão descritas nas secções seguintes. Para uma descrição mais completa e formal, bem como para a demonstração de algumas propriedades destas operações, ver [Cravo 1993*a*].

As entidades sujeitas às operações de mudança são conjuntos finitos de crenças, que constituem a base para todas as crenças do agente. Na terminologia da SWMC estes conjuntos correspondem a contextos. Para que possamos caracterizar as operações é necessário definir primeiro quais são as consequências de um contexto. Consideramos como consequências de um contexto (para a definição das operações) a reunião de todos os espaços de crenças definidos por esse contexto em SWMC, ou seja todas as consequências concebíveis do contexto.

3.1 Adição de uma fbf a um contexto

A adição de uma fbf A a um contexto β, representada por $(\beta + A)$, corresponde a acrescentar A a β, e é trivialmente definida da seguinte forma: $(\beta + A) = \beta \cup \{A\}$.

Quando se acrescenta uma fbf a um contexto, é possível que passem a existir novas consequências desse contexto, mas por esta teoria ter subjacente uma lógica não-monótona, também é possível que algumas das consequências que existiam anteriormente deixem de ser acreditadas. No entanto, e como se pode ver pela sua definição, esta operação é sempre definida de forma única.

3.2 Remoção de uma fbf de um contexto

A remoção de uma fbf A de um contexto β corresponde a alterar β de modo a que A deixe de ser uma consequência desse contexto. Para isso, têm que ser invalidadas todas as derivações de A a partir de β.

As derivações de A a partir de β são definidas como sendo os conjuntos mínimos de fbfs α tais que $\alpha \vdash_{SWMC} A$, considerando que α está contido em algum núcleo definido por β.

A operação de remoção de uma fbf A de um contexto β é representada por $(\beta - A)$ e pode ser feita de várias formas, quer retirando quer acrescentando fbfs a β, dependendo das derivações existentes para A a partir de β. Por este motivo, o resultado da operação de remoção de uma fbf de um contexto é um conjunto de contextos.

A forma de invalidar uma derivação α de uma fbf A depende do conteúdo de α:

- Se $\alpha = \{\}$, então A é um teorema, isto é, A é consequência de qualquer contexto consistente, e não existe nenhuma maneira de fazer com que A deixe de ser consequência de β. Neste caso, por definição, $(\beta - A) = \{\beta\}$.

- Se $\alpha \cap \mathcal{L}_A = \{\}$, isto é, se em α só foram utilizadas hipóteses, então a única forma de invalidar α é retirar de β uma das hipóteses que tenham sido usadas nesta derivação. Se quisermos que a mudança a fazer em β seja mínima, então devemos retirar exactamente uma dessas hipóteses.

- Se $\alpha \cap \mathcal{L}_A \neq \{\}$, o que significa que foram usadas uma ou mais regras de omissão na derivação α, então α pode ser invalidada, quer retirando uma das hipóteses em α, quer acrescentando a β informação que bloqueie a aplicação de alguma regra de omissão que tenha sido usada em α. Para bloquear a aplicação de uma regra de omissão, a informação a acrescentar consiste em assumir que estamos na presença do que corresponde ou a uma das excepções da regra ou então à negação da conclusão da regra.

No caso de A não ser um teorema, podem existir várias maneiras possíveis de invalidar todas as derivações de A em β. De todas essas maneiras, estamos interessados em escolher uma que seja mínima, isto é, que faça o mínimo de alterações possível em β. Para isso, de entre as várias alternativas para invalidar uma derivação de A que retiram fbfs de β, escolhemos a que retirar menos; se ainda existirem várias alternativas "mínimas" que retiram as mesmas fbfs de β, de entre elas escolhemos a que acrescentar menos.

Mas este critério não é suficiente para escolher a mudança de forma única, como vimos pelas definições dadas acima. Em [Gärdenfors 1988] é introduzido o conceito de *valor epistémico* de uma crença, de forma a que se possam ordenar as crenças do agente de acordo com esse valor, que intuitivamente representa a importância dessa crença. Assim, quando existem várias alternativas para a remoção, pode-se escolher as que retiram as crenças de menor valor. Na abordagem de Gärdenfors a ordem entre as crenças é total. No entanto, como [Doyle 1991] argumenta, a informação sobre a importância relativa das várias crenças de um agente raramente é tão completa. Normalmente apenas existem várias ordens parciais que podem, inclusivé, ser contraditórias entre si. Por este motivo, a TRC considera a possibilidade de existirem várias ordens parciais de preferência entre as crenças do agente, e ainda uma ordem parcial entre essas ordens, para resolver os casos em que elas se contradizem. Na TRC é criada uma ordem parcial entre as crenças que resulta da combinação destas várias ordens entre crenças e da ordem entre as ordens. Esta ordem é usada para guiar as alterações a efectuar nas crenças do agente.

Apesar da utilização destas ordens de preferência para guiar as alterações a efectuar nas crenças do agente, pode não ser possível determinar essas alterações de forma única. Neste caso dizemos que existem várias alternativas igualmente aceitáveis para remover A de β.

3.3 Revisão de um contexto com uma fbf

A operação de revisão de um contexto β com uma fbf A, representada por $(\beta * A)$, corresponde a alterar β de modo a que contenha A e seja consistente, e é definida à custa da remoção e da adição, usando a identidade de Levi [Levi 1977].

Tal como na remoção, o resultado desta operação é um conjunto de contextos. Para definir a operação de revisão, consideramos $\beta_{FOL} = \beta \cap \mathcal{L}_{FOL}$, $\beta_D = \beta \cap \mathcal{L}_D$, e $\beta_E = \beta \cap \mathcal{L}_E$:

$$(\beta * A) = \begin{cases} \{\beta' \cup \beta_D \cup \beta_E \cup \{A\} : \beta' \in (\beta_{FOL} - \neg A)\} & se\ A \in \mathcal{L}_{FOL} \\ \\ \{\beta \cup \{A\}\} & caso\ contrário \end{cases}$$

Repare-se que:

- Se A for consistente com β, basta adicioná-la ao contexto e não fazer mais alterações.
- Se A for incompatível com uma consequência sólida de β, então $\beta \cup \{A\}$ é inconsistente, e é necessário remover pelo menos uma hipótese que não seja A para o tornar consistente.
- Se A for incompatível com uma consequência plausível ou concebível (mas não sólida) de β, não é necessário fazer nenhuma alteração a β, porque a lógica subjacente (a SWMC) garante que a consequência que era incompatível com A já não é consequência de $\beta \cup \{A\}$.

4 O sistema de revisão de crenças

Em [Cravo 1992] é apresentado um Sistema de Revisão de Crenças (daqui em diante designado por SRC) que se baseia na lógica SWMC e na Teoria de Revisão de Crenças descritas nas secções 2 e 3, respectivamente.[14]

Os sistemas de revisão de crenças são, fundamentalmente, sistemas computacionais. Como tal, apesar de o SRC ser apresentado de uma forma abstracta, não se compremetendo com uma implementação particular, a sua definição tem em consideração as limitações inerentes aos sistemas computacionais.

[14] A lógica SWMC e a TRC descritas neste trabalho correspondem a versões mais recentes das que são apresentadas em [Cravo 1992]. As alterações registadas, no entanto, não influenciam o SRC, pelo que a referência apresentada é suficiente para uma boa compreensão do mesmo.

Uma das limitações evidentes destes sistemas é a sua incapacidade de lidar com espaços de crenças, tal como eles são definidos na lógica, uma vez que estes, sendo fechados em relação à operação de consequência lógica, são infinitos.

Por outro lado, a implementação de um sistema de dedução em Lógica de Primeira Ordem enfrenta sempre um problema: a semi-decidibilidade do processo de dedução. As lógicas não-monótonas, geralmente, permitem saltar para conclusões se isso não tornar as crenças do agente inconsistentes, o que significa que necessitam de efectuar testes de consistência numa lógica de primeira ordem, um problema semi-decidível. Este é um dos motivos que torna particularmente difícil a utilização computacional de formalismos não-monótonos.

Estes são alguns dos problemas que o SRC tem de considerar. Para além disso, tem que considerar as tarefas básicas de um sistema de revisão de crenças: manter dependências entre as crenças para que as alterações possam ser propagadas sem muito esforço computacional; e resolver (ou ajudar o utilizador a resolver) as contradições que surjam nas crenças do agente.

Tendo em conta o registo de dependências, o SRC é um sistema do tipo ATMS, associando a cada crença as crenças que estão na base da sua derivação. Repare-se que a SWMC mantém já esse registo nas fbfs suportadas, no conjunto de origem. No SRC utilizam-se *fbfs-justificadas* para fazer esse registo de dependências, para manter informação que permita a propagação de alterações nas crenças quando tal é necessário, e para permitir que o sistema explique as razões pelas quais acredita numa determinada fbf. Uma fbf-justificada é representada por $<A, \tau, \alpha, \gamma>$, correspondendo à fbf suportada da SWMC $<A, \tau, \alpha>$ mas em que se regista, em γ, quais foram as crenças imediatamente utilizadas na derivação de A. Este registo serve também para que o sistema consiga actualizar as crenças com todas as suas possíveis derivações. Para mais detalhes consultar [Cravo 1992].

4.1 Estruturas do SRC

A informação que representa as crenças do agente é mantida em quatro estruturas:

- A *base de conhecimento* onde estão todas as fbfs justificadas conhecidas do sistema (quer por introdução do exterior, quer por inferência). Repare-se que este conjunto é necessariamente finito.

- O *conjunto de contextos* onde estão todos os contextos considerados até ao momento pelo sistema. Associadas a cada contexto são armazenadas as estruturas que ele define: contexto estendido, núcleos primitivos e núcleos.

- O *conjunto de conjuntos inconsistentes* onde estão os conjuntos de fórmulas já detectados pelo agente como sendo inconsistentes. De notar que podem existir outros conjuntos de fórmulas inconsistentes, sem o sistema ter "conhecimento" disso.

- O *conjunto de suposições* onde estão todas as suposições levantadas pelo sistema até ao momento.

Estas estruturas são actualizadas pelos dois componentes fundamentais do SRC: o *motor de inferência* e o *revisor de crenças*. É através da consulta e actualização destas estruturas que o motor de inferência e o revisor de crenças comunicam e cooperam na modelação das crenças de um agente.

4.2 O motor de inferência

O motor de inferência serve para adicionar nova informação à base de conhecimento, quer vinda do exterior, quer resultante da aplicação de regras de inferência para gerar novas fbfs.

As regras de inferência utilizadas pelo SRC são as da lógica SWMC, modificadas para trabalhar com as fbfs-justificadas, e podem ser utilizadas pelo SRC quer para efectuar inferência para a frente, quer para efectuar inferência para trás.

São também tarefas do motor de inferência:

- A determinação das consequências de um contexto — calcular os contextos estendidos, os núcleos primitivos e os núcleos definidos por um contexto. Para isso baseia-se nas definições destes conceitos na SWMC.

- A explicação — dizer por que razões acredita numa determinada fbf. Utiliza as justificações das fbfs justificadas para saber quais os passos seguidos na sua derivação e dar uma explicação que depende das regras de inferência utilizadas.

- A propagação de dependências — para além da propagação de dependências que é feita automaticamente através das regras de inferência, e que corresponde a dizer de que fbfs depende uma fbf, o motor de inferência também se serve das justificações para actualizar os conjuntos de origem de todas as fbfs que dependem de uma fbf para a qual foi encontrada uma nova derivação.

- A actualização das estruturas do SRC — é através da actualização destas estruturas que se representam as alterações feitas na informação existente no sistema. A sua actualização é importante, não só para posteriores derivações do motor de inferência, mas também para o revisor de crenças. Por exemplo, o motor de inferência actualiza o conjunto de conjuntos inconsistentes sempre que é detectada uma contradição.

4.3 O revisor de crenças

O revisor de crenças é o componente responsável pela tarefa de revisão de crenças: sempre que existe uma contradição deve detectá-la e determinar quais os possíveis "culpados"; é também o revisor de crenças o responsável pela remoção de fbfs de contextos. Estas tarefas são efectuadas com base na TRC descrita na secção 3. Para além disso, o revisor também é o responsável pela escolha dos espaços de crenças preferidos pelo sistema.

Uma vez que a SWMC, na qual o SRC se baseia, associa a cada fbf as fórmulas utilizadas numa derivação (no conjunto de origem das fbfs suportadas), a determinação dos "culpados" de uma contradição pode ser feita de uma forma relativamente simples: quando é detectada uma contradição, os "culpados" encontram-se entre as fbfs do conjunto de origem da fbf suportada que corresponde à contradição. Em SWMC existem dois tipos de contradição: *contradições reais* e *contradições aparentes*. As primeiras ocorrem entre conclusões sólidas, derivadas apenas a partir de hipóteses, e são tratadas pelo revisor de crenças. As contradições aparentes são aquelas em cuja derivação foram usadas suposições, e não constituem na realidade contradições, não dando lugar a uma revisão de crenças. Em qualquer dos casos, o conjunto de conjuntos inconsistentes é actualizado pelo motor de inferência.

Tal como foi dito, o revisor de crenças apenas trata das contradições reais e, ao contrário da generalidade dos sistemas de revisão de crenças, que apenas identificam os "culpados" de uma contradição, mas não dizem nada acerca da forma de a resolver, utiliza a TRC para resolver a contradição, ou pelo menos auxiliar o utilizador do sistema nessa tarefa. Para isso, o SRC utiliza as ordens de preferência que o utilizador tenha eventualmente fornecido. Para além disso, o sistema determina uma dessas ordens de forma automática — a *ordem de especificidade* — segundo a qual são preferidas as regras de omissão mais específicas às regras de omissão menos específicas. A forma de determinar se uma regra de omissão é mais específica do que outra é a seguinte: dizemos que a regra de omissão $\nabla(x)A_1(x) \rightarrow B_1(x)$ é mais específica que a regra de omissão $\nabla(x)A_2(x) \rightarrow B_2(x)$, ambas pertencentes a um contexto β, se e só se

$$\beta \vdash \forall(x)A_1(x) \rightarrow A_2(x) \;\; e \;\; \beta \not\vdash \forall(x)A_2(x) \rightarrow A_1(x).$$

Esta ordem parcial, apesar de definida automaticamente pelo sistema, comporta-se como uma qualquer ordem introduzida pelo utilizador, na medida em que lhe pode ser dada uma importância relativamente às restantes, através da ordem entre as ordens introduzida pelo utilizador. A ordem entre as ordens é única, e serve para desambiguar os casos em que existe conflito entre as várias ordens de preferência parciais. A combinação de todas as ordens parciais (incluindo a ordem de especificidade), usando a ordem entre as ordens sempre que necessário, dá origem à *ordem de preferência combinada*.

Esta ordem é utilizada pelo revisor de crenças no processo de resolução de uma contradição real, bem como para determinar quais os espaços de crenças preferidos definidos por um determinado contexto.

Uma vez que cada espaço de crenças é determinado univocamente por um dos núcleos definidos pelo contexto, preferir um espaço de crenças vai corresponder, na realidade, a preferir um dos núcleos. A escolha do núcleo preferido vai ser feita apenas com base nas preferências existentes entre as suposições, pois todas as hipóteses do contexto estão em todos os núcleos definidos por ele, e por isso as preferências entre as hipóteses do contexto não vão servir para se saber qual é preferido. No entanto, como o sistema apenas admite a especificação de preferências entre hipóteses, para determinar as preferências entre as suposições vão ser usadas as preferências entre as regras de omissão que lhes correspondem.[15] Para além disso, nem todas as suposições dos núcleos vão servir para escolher o preferido. As únicas suposições importantes para escolher, de entre dois núcleos, qual o núcleo preferido são as suposições que estão em conflito nesses dois núcleos, isto é, as suposições que estão num dos núcleos e não estão no outro porque se estivessem o tornariam inconsistente. Assim, na escolha de qual é o núcleo preferido de entre dois núcleos, apenas são consideradas as preferências entre as suposições que estão em conflito dos dois núcleos.

Em [Cravo 1992] (e resumidamente em [Cravo & Martins 1993]) é apresentada a descrição de uma implementação deste SRC — o SNePSwD — uma extensão ao sistema de representação do conhecimento e raciocínio baseado em redes semânticas, o SNePS ([Shapiro 1979, Shapiro & Martins 1990, Shapiro & Rapaport 1993]).

5 Preferências entre crenças como meta conhecimento

Na modelação de um agente cognitivo, um dos problemas fundamentais que precisa de ser resolvido é a escolha da linguagem utilizada para representar o conhecimento desse agente.

[15] Como seria de esperar, uma suposição é preferida a outra se a regra de omissão que lhe corresponde for preferida em relação à da outra.

Naturalmente, associados à linguagem escolhida, estão os mecanismos para inferir conhecimento implícito a partir do conhecimento explícito. A adequação de uma linguagem depende de uma série de factores: o poder expressivo, a complexidade, a naturalidade de representação, etc. Este é um assunto muito controverso e sobre o qual não existe consenso generalizado, mas as abordagens baseadas em lógica são frequentemente utilizadas, dado o seu poder expressivo e a existência de sistemas de inferência sólidos.

Este trabalho enquadra-se no objectivo geral de modelar um agente que faça raciocínio de senso-comum, utilizando para isso uma lógica não-monótona, a SWMC, descrita na secção 2.

No entanto, a lógica é apenas um dos componentes de um sistema que modele um agente cognitivo. Como vimos na descrição do Sistema de Revisão de Crenças (secção 4), este é composto por dois componentes: o motor de inferência, que se baseia na lógica; e o revisor de crenças, que revê as crenças do agente quando é detectada uma contradição, ou que escolhe um espaço de crenças entre os vários aceitáveis para o agente. Para efectuar estas tarefas, o revisor de crenças utiliza informação acerca das crenças do agente — que crenças são preferidas a quais, ou seja, quais são as crenças mais importantes para o agente. Essa informação é representada por várias ordens parciais entre as crenças do agente.

Consideramos que esta informação constitui conhecimento importante que um agente deve ter acerca das suas crenças, isto é, *meta conhecimento*. Como a lógica utilizada para representar o conhecimento do agente é uma lógica de primeira ordem, não permite representar este tipo de informação, que passa a constituir um nível adicional de conhecimento, que deve ser representado utilizando uma linguagem adequada. Para distinguir entre os dois níveis de conhecimento, vamos designar por:

- *Nível-base* — o nível correspondente à lógica SWMC, onde está representado o conhecimento que o agente tem sobre um determinado domínio e sobre o qual pretende raciocinar.

- *Nível-meta* — o nível onde está representado o conhecimento sobre as preferências entre as crenças do nível-base do agente.

O conhecimento do nível-meta é constituido por relações de ordem parcial entre a "importância" das crenças do nível-base. Uma forma simples e natural de representar este conhecimento é através da enumeração exaustiva das várias relações: se o agente tem n crenças, representadas no nível-base por n fbfs de SWMC, $fbf_1 \ldots fbf_n$, então uma forma de indicar que a fbf_i é preferida à fbf_j (com $i, j \leq n$), segundo uma determinada ordem parcial é dizer que o par $<fbf_i, fbf_j>$ pertence a essa relação.

No entanto, esta forma de especificar as preferências não é muito adequada, visto a explicitação das ordens parciais poder ser muito extensa. A motivação para este trabalho é a convicção de que é possível definir uma linguagem que permita estabelecer as preferências entre as crenças de um agente de uma forma mais sintéctica e intuitiva do que a enumeração explícita e exaustiva de todas as ordens de preferência. Pensamos que as preferências de um agente sobre as suas crenças podem muitas vezes ser expressas através de regras que indicam que as crenças de um determinado tipo (ou forma) são preferidas às crenças de um outro tipo. Essas regras permitem que, implicitamente, com apenas uma frase da nossa linguagem, se indique um conjunto de elementos de relações de preferência, em vez de apenas um.

Julgamos que a maior parte do conhecimento de preferências entre crenças é dependente do domínio representado no nível-base, e constitui conhecimento existente sobre esse domínio, essencial para o raciocínio de senso comum, mas que, normalmente, por ser mais difícil de formalizar em lógica, não é representado. A existência de dois níveis conceptualmente separados permite a modularização desse conhecimento, possibilitando a sua representação de uma forma mais natural. No entanto, com uma linguagem suficientemente expressiva, no nível-meta pode estar representado conhecimento geral, indicando princípios de preferência gerais e independentes do domínio do nível-base, como, por exemplo, a ordem parcial de especificidade existente no SRC. Assim, pretendemos uma linguagem onde possam ser representados estes dois tipos de conhecimento: conhecimento de preferências dependente e independente do domínio.

Vamos agora apresentar um exemplo em que a utilização de uma linguagem mais expressiva para descrever as preferências entre as crenças do agente é útil.

Exemplo 1 (Exemplo das reuniões)

O General Guerra prefere ter reuniões de manhã

O Capitão Capucho prefere ter reuniões de tarde

Normalmente, uma pessoa tem reuniões na altura que prefere

Uma reunião só pode ser de manhã ou de tarde

O General Guerra e o Capitão Capucho têm uma reunião

A modelação deste conhecimento em SWMC dá origem a dois espaços de crenças: um em que a reunião ocorre de manhã e outro em que a reunião ocorre de tarde. Intuitivamente, devemos preferir aquele em que a reunião ocorre de manhã, uma vez que isso corresponde aos gostos do general Guerra, e os gostos de militares hierarquicamente superiores sobrepõem-se aos dos seus subordinados. É este tipo de conhecimento de senso comum, difícil de representar no nível-base, que deve estar no nível-meta.

Neste exemplo, se déssemos preferência aos gostos do General sobre os do Capitão, poderíamos escolher o espaço de crenças pretendido.[16] Mas aquilo que nós sabemos é mais geral: "os gostos dos militares hierarquicamente superiores sobrepõem-se aos dos seus subordinados".[17] Para podermos representar este conhecimento, não basta a enumeração dos pares da relação de preferência entre as crenças, precisamos de introduzir uma regra.

A indicação explícita das relações de preferência existentes entre as crenças do agente tem, para além da questão anterior, outras desvantagens:

- Se tivermos vários generais e vários capitães, para especificar as preferências precisamos de indicar $n^{\underline{o}}$ *de generais* $\times$ $n^{\underline{o}}$ *de capitães* pares da ordem parcial.
- Para que possamos indicar as preferências entre duas crenças elas têm, naturalmente, que existir nessa altura.

Consideremos agora o seguinte exemplo em que eram adicionadas várias hipóteses às crenças do agente.

Exemplo 2 (Exemplo das reuniões (cont.))

O General Guerra gosta de almoçar às 13 horas

O Capitão Capucho gosta de almoçar às 14 horas

Normalmente, uma pessoa almoça à hora que mais gosta

Uma pessoa só pode almoçar a uma determinada hora

O General Guerra e o Capitão Capucho vão almoçar juntos

Com a existência de uma linguagem mais expressiva em que se possa representar uma regra que reflicta as preferências da hierarquia militar, não é necessário adicionar nenhum

[16] Na realidade, admitindo que a representação do exemplo em SWMC apenas contém uma regra de omissão, correspondendo à crença de que "*Normalmente, uma pessoa tem reuniões na altura que prefere*", não seria possível, com o sistema de revisão de crenças descrito na secção 4, preferir um dos espaços de crenças. No entanto, neste trabalho estendemos as preferências de forma a poder tratar estes casos (secção 8).

[17] Pelo menos no que diz respeito a reuniões e assuntos militares.

conhecimento ao nível-meta para tratar também deste caso. Basta "raciocinar" ao nível-meta, sobre as crenças existentes no nível-base nesta altura, para que se infira que se prefere o espaço de crenças em que o almoço é às 13 horas.

Assim, a linguagem utilizada para representar o conhecimento do nível-meta deve ter mecanismos de inferência associados.

É geralmente aceite que, de entre as várias linguagens existentes para representar conhecimento, a lógica é a que tem um maior poder expressivo, com mecanismos de inferência sólidos, e cuja semântica é bem definida. Para além disso, se considerarmos que o raciocínio no nível-meta é efectuado pelo mesmo agente que raciocina no nível-base, os processos de raciocínio devem ser essencialmente os mesmos nos dois níveis. Como a linguagem utilizada para representar e raciocinar sobre o conhecimento do nível-base é a lógica SWMC, é natural que essa seja a linguagem utilizada também no nível-meta. Na realidade, a lógica utilizada para representar o conhecimento no nível-meta vai ser uma especialização da lógica SWMC, no sentido de representar conhecimento específico de preferências sobre as crenças no nível-base. Os dois níveis diferem, não pelo tipo de raciocínio efectuado, mas sim pelo tipo de conhecimento existente.

Poderíamos considerar que não é necessário utilizar uma linguagem tão expressiva para representar o conhecimento sobre preferências, sendo possível a utilização de uma linguagem especializada. Mas, se pretendemos representar conhecimento do tipo do indicado acima (preferências das hierarquias militares), em que uma regra se aplica a todas as crenças do nível-base que dizem respeito a generais ou capitães, então precisamos de uma linguagem de primeira ordem, que possa quantificar sobre todas essas crenças.[18] E se, de facto, o conhecimento a representar no nível-meta não é tão geral como o do nível-base, constitui ainda, como já referimos anteriormente, conhecimento de senso comum, provavelmente incompleto, e necessitando assim de uma linguagem que permita lidar com este tipo de conhecimento. A lógica SWMC foi concebida precisamente para esse propósito, motivo pelo qual é utilizada para representar o conhecimento não só do nível-base, mas também do nível-meta.

6 Referência no nível-meta ao nível-base

Um dos problemas que necessita de ser resolvido, na especificação de uma linguagem para descrever o conhecimento do nível-meta, é a forma como se faz a referência, nesse nível,

[18] Para isso é preciso ainda que nos possamos referir às crenças do nível-base no nível-meta. Esse assunto é abordado na secção 6.

às crenças existentes no nível-base. De facto, se o conhecimento do nível-meta é sobre as crenças do nível-base, estas são alguns dos objectos do universo de discurso do nível-meta. Naturalmente, a referência às crenças do nível-base vai ser feita através das fbfs da SWMC utilizadas para as representar.

O que se pretende é definir um "nome" no nível-meta para uma fbf do nível-base. Essa definição pode ser feita de várias formas, e é vulgar designar-se essa operação por "*lifting*". Em [van Harmelen, Simpson, Giunchiglia, Serafini & Smaill 1994] são discutidas algumas das propriedades da operação de "lifting", e defende-se a utilidade de essa operação definir nomes no nível-meta que sejam "mais informativos" do que a mera descrição sintáctica das fbfs utilizadas no nível-base. Apesar de concordarmos com essa ideia, de uma forma geral, a definição da operação necessária para efectuar esse tipo de "lifting" está fora do âmbito deste trabalho.

6.1 O poder expressivo necessário

A definição de uma operação de "lifting" deve ser adequada ao tipo de utilização que se pretende dar à meta-teoria, de acordo com o nível de detalhe com que se pretende aceder ao nível-base. A forma mais simples de nos referirmos no nível-meta a uma fórmula do nível-base é considerar que, para cada fórmula do nível-base, é criada uma constante no nível-meta que lhe corresponde. Por exemplo, a fórmula do nível-base $General(Guerra)$ era associada à constante "$General(Guerra)$" no nível-meta. Esta correspondência é simples de definir, mas limita muito o poder expressivo do nível-meta. Como pretendemos, com uma fórmula do nível-meta, descrever as relações de preferência entre várias crenças do nível-base, é essencial que tenhamos a possibilidade de colocar variáveis no lugar de alguns dos símbolos de uma fórmula do nível-base. Para além da quantificação sobre os símbolos do nível-base, queremos quantificar sobre as próprias fórmulas ou suas sub-fórmulas. Alguns exemplos do tipo de termos que desejamos utilizar no nível-meta são:

- Fórmulas de SWMC:

$$\begin{gathered}\bigtriangledown(x)[Ave(x) \to Voa(x)] \\ General(Guerra)\end{gathered} \tag{1}$$

- Fórmulas de SWMC em que se substituem alguns dos termos da fórmula por variáveis:

$$\begin{gathered}\bigtriangledown(x)[Ave(x) \to Voa(\omega)] \\ General(\omega)\end{gathered} \tag{2}$$

em que ω é uma variável da nossa lógica. Repare-se que estes termos, dependendo da instanciação da variável ω, descrevem outras fórmulas do nível-base para além das correspondentes aos termos em (1). Por exemplo, o termo $\bigtriangledown(x)[Ave(x) \rightarrow Voa(\omega)]$, quando ω é instanciado com x corresponde à fbf do nível-base $\bigtriangledown(x)[Ave(x) \rightarrow Voa(x)]$; quando ω é instanciado com $m\tilde{a}e(x)$, à fbf $\bigtriangledown(x)[Ave(x) \rightarrow Voa(m\tilde{a}e(x))]$.

- Fórmulas de SWMC em que se substituem alguns dos símbolos de predicado por variáveis:

$$\begin{gathered} \bigtriangledown(x)[Ave(x) \rightarrow A(x)] \\ A(Guerra) \end{gathered} \tag{3}$$

  em que A é uma variável da nossa lógica. Estes termos, com as possíveis instanciações para a variável A, correspondem a todas as fórmulas que dizem que "Tipicamente, as aves são A", ou que "Normalmente, as aves têm a propriedade A", e todas as fórmulas atómicas que dizem algo acerca do "$Guerra$". Em particular, descrevem as fórmulas correspondentes aos termos em (1), mas não as de (2), a menos que se utilize a variável ω, tal como nesse exemplo.

- Fórmulas de SWMC em que se substituem algumas das suas sub-fórmulas por variáveis

$$\begin{gathered} \bigtriangledown(x)[Ave(x) \rightarrow \Psi] \\ \Psi \end{gathered} \tag{4}$$

  em que Ψ é uma variável da nossa lógica. Estes termos, com as possíveis instanciações para a variável Ψ correspondem a todas as fórmulas que dizem que "Tipicamente, se x é uma ave, então Ψ", no caso do primeiro; e todas as fórmulas (atómicas e não atómicas) no caso do segundo. Estes termos são ainda mais gerais que os termos em (3).

A operação de "lifting" definida para a LEP tem que levar em linha de conta o poder expressivo que desejamos para nos referirmos aos objectos do nível-base — as crenças básicas do agente.

6.2 Uma solução da literatura

Em [Genesereth & Nilsson 1987, Genesereth 1988] é apresentada uma variação à lógica clássica, de forma a ter capacidades introspectivas, isto é, na lógica podemos não só falar dos objectos do "mundo exterior", mas também dos próprios símbolos e estruturas da linguagem da lógica.[19]

Neste caso, há também a necessidade de descrever como é que se faz a referência às estruturas da própria linguagem e, em particular, de distinguir entre a referência a um objecto do "mundo exterior" e a referência a um objecto da própria linguagem. Por exemplo, distinguir entre o pássaro "$Tweety$" e o símbolo "$Tweety$". A solução apresentada permite descrever as fórmulas da linguagem de forma a que se possa falar sobre os seus vários constituintes até ao nível dos símbolos utilizados:

- A forma de distinguir entre o símbolo e o objecto do "mundo exterior" a que ele se refere é adoptar uma convenção de notação — quando nos queremos referir ao pássaro "$Tweety$" utilizamos o símbolo $Tweety$; quando nos queremos referir ao símbolo escrevemos o símbolo entre aspas: "$Tweety$".

- Qualquer expressão da linguagem é representada por uma sequência da representação das suas sub-expressões constituintes. Assim, para falarmos sobre a fórmula atómica $Ave(Tweety)$, utilizamos o termo $[\text{``}Ave\text{''}, \text{``}Tweety\text{''}]$ e para a fórmula $Ave(Tweety) \rightarrow Voa(Tweety)$ utilizamos o termo $[[\text{``}Ave\text{''}, \text{``}Tweety\text{''}], \rightarrow, [\text{``}Voa\text{''}, \text{``}Tweety\text{''}]]$.

Com esta descrição das fórmulas, é possível quantificar não só sobre os termos das fórmulas, como também sobre as suas sub-fórmulas, sobre os símbolos de predicado, e até sobre as conectivas lógicas. Para além desta forma de descrever as fórmulas através de termos, considera-se ainda a existência na linguagem de quatro predicados que permitem indicar qual é o tipo dos símbolos: se são constantes, variáveis, símbolos de função ou símbolos de predicado.

As extensões destes predicados são definidas através de axiomas na lógica, indicando o tipo dos vários símbolos: por exemplo, temos como axiomas $Pred(Voa)$ e $Const(Tweety)$.[20]

O sistema dedutivo para a nossa lógica é um sistema de dedução natural, logo, sem axiomas. Assim, a nossa formulação para a operação de "lifting" das fórmulas do nível-base é ligeiramente diferente desta, evitando a existência destes predicados. Fazendo na operação de "lifting" uma representação com mais informação, é possível guardar a informação relativa ao tipo dos símbolos nos termos que correspondem às fórmulas do nível-base.

[19] A lógica apresentada por Genesereth permite na própria linguagem fazer referência às fórmulas da linguagem. No nosso caso, temos duas linguagens, em dois níveis distintos, em que uma fala sobre a outra mas nunca há referência à mesma linguagem.

[20] O nome dos predicados apresentados no trabalho de Genesereth é diferente. Utilizamos estes nomes por o seu significado ser evidente.

6.3 A operação de "lifting" para o nível-meta

Para guardarmos informação acerca do tipo dos símbolos do nível-base, criamos representações estruturadas (como as sequências de Genesereth, apresentadas na secção anterior), em que temos várias funções para cada um dos tipos possíveis de expressões existentes no nível-base: constantes, variáveis, termos funcionais, fórmulas atómicas, disjunções, ...

No nosso caso, a operação de "lifting" é descrita através da função de tradução $\mathcal{T}$:

$$
\begin{array}{lcl}
\mathcal{T}(a) & = & ct(a) \\
\mathcal{T}(x) & = & var(x) \\
\mathcal{T}(f(t_1,\ldots,t_n)) & = & function(f,[\mathcal{T}(t_1),\ldots,\mathcal{T}(t_n)]) \\
\mathcal{T}(P(t_1,\ldots,t_n)) & = & pred(P,[\mathcal{T}(t_1),\ldots,\mathcal{T}(t_n)]) \\
\mathcal{T}(\neg A) & = & not(\mathcal{T}(A)) \\
\mathcal{T}(A \wedge B) & = & and(\mathcal{T}(A),\mathcal{T}(B)) \\
\mathcal{T}(A \vee B) & = & or(\mathcal{T}(A),\mathcal{T}(B)) \\
\mathcal{T}(A \rightarrow B) & = & imply(\mathcal{T}(A),\mathcal{T}(B)) \\
\mathcal{T}(\forall(x)A) & = & forall(\mathcal{T}(x),\mathcal{T}(A)) \\
\mathcal{T}(\exists(x)A) & = & exists(\mathcal{T}(x),\mathcal{T}(A)) \\
\mathcal{T}(\bigtriangledown(x)A) & = & default(\mathcal{T}(x),\mathcal{T}(A)) \\
\mathcal{T}(Applicable(D,c)) & = & applicable(\mathcal{T}(D),\mathcal{T}(c))
\end{array}
$$

em que:

$$
\left.
\begin{array}{ll}
a & \text{é uma constante} \\
x & \text{é uma variável} \\
t_1,\ldots,t_n & \text{são termos} \\
f & \text{é um símbolo de função} \\
P & \text{é um símbolo de predicado} \\
A \text{ e } B & \text{são fórmulas} \\
D & \text{é uma regra de omissão}
\end{array}
\right\} \text{do nível-base.}
$$

A aridade de cada um dos símbolos de função introduzidos no nível-meta é evidente a partir da definição da função de tradução. Repare-se que os símbolos de função *function* e *pred* têm ambos aridade 2, em que: o primeiro argumento é um símbolo de função ou de predicado, respectivamente, do nível-base; o segundo argumento é uma sequência com a representação dos argumentos desses símbolos.[21]

A tradução dos exemplos em (1) utilizando esta função de tradução é feita da seguinte forma:

$$
\begin{aligned}
\mathcal{T}(\bigtriangledown(x)[Ave(x) \rightarrow Voa(x)]) &= default(var(x), imply(pred(Ave, [var(x)]), \\
&\qquad\qquad pred(Voa, [var(x)]))) \\
\mathcal{T}(General(Guerra)) &= pred(General, [ct(Guerra)])
\end{aligned}
$$

Agora, a forma de obter os termos descritos nos exemplos (2) a (4) é muito simples, basta substituir alguns dos sub-termos por variáveis da LEP:

$$
\begin{aligned}
\bigtriangledown(x)[Ave(x) \rightarrow Voa(\omega)] &\rightsquigarrow default(var(x), imply(pred(Ave, [var(x)]), \\
&\qquad\qquad pred(Voa, [\omega]))) \\
General(\omega) &\rightsquigarrow pred(General, [\omega]) \\
\bigtriangledown(x)[Ave(x) \rightarrow A(x)] &\rightsquigarrow default(var(x), imply(pred(Ave, [var(x)]), \\
&\qquad\qquad pred(A, [var(x)]))) \\
A(Guerra) &\rightsquigarrow pred(A, [ct(Guerra)]) \\
\bigtriangledown(x)[Ave(x) \rightarrow \Psi] &\rightsquigarrow default(var(x), imply(pred(Ave, [var(x)]), \Psi)) \\
\Psi &\rightsquigarrow \Psi
\end{aligned}
$$

Com esta formulação, no entanto, não permitimos a quantificação sobre as conectivas das fórmulas. A alteração da função $\mathcal{T}$ para permitir isso é trivial (basta, por exemplo, fazer uma tradução do tipo: $A\Theta B$ traduzido por $formula(\Theta, A, B)$, sendo Θ uma conectiva binária) mas julgamos que essa possibilidade não tem utilidade para o tipo de conhecimento que pretendemos exprimir ao nível-meta.

Como existe uma correspondência biunívoca entre as fórmulas do nível-base e os termos utilizados ao nível-meta para as descrever, para facilitar a leitura das fórmulas do nível-meta que utilizam estes termos, vamos escrever os termos do nível-meta como as fbfs do nível-base, isto é, como nos exemplos acima (1–4).

Por fim, falta ainda considerar como é que se pode distinguir entre as referências aos objectos do nível-base e as referências aos objectos do "mundo exterior". Vamos adoptar a solução, descrita acima, de colocar os símbolos entre aspas. Só que, como normalmente estamos a falar sobre os objectos do nível-base, utilizamos a notação ao contrário, isto é, o

[21]Representamos uma sequência por "$[e_1, \ldots, e_n]$", sendo $e_1, \ldots, e_n$ os elementos da sequência. Utilizamos esta notação vulgarmente utilizada para evitar complicar a apresentação da função de tradução.

símbolo $Tweety$ no nível-meta refere-se ao símbolo $Tweety$ do nível-base; o símbolo "$Tweety$", também do nível-meta, refere-se a um objecto do "mundo exterior".

6.4 A possibilidade de falar sobre esquemas de fórmulas

Na meta-linguagem utilizada para descrever propriedades acerca de certas fórmulas de uma lógica, são utilizados frequentemente esquemas de fórmulas, isto é, fórmulas contendo meta-variáveis que podem ser subtituidas por qualquer fórmula.

No nível-meta, de acordo com o que foi apresentado na secção anterior, podemos utilizar esse tipo de esquemas, utilizando variáveis do nível-meta que podem ser substituidas por termos do nível-meta que representam fórmulas do nível-base. Por exemplo, a forma de representar o esquema $A \rightarrow B$ no nível-meta, é simplesmente $A \rightarrow B$,[22] em que A e B são variáveis do nível-meta.

Um outro tipo de esquemas muito utilizado pode ser encontrado, por exemplo, na definição de "regra de omissão mais específica segundo o SRC", na página 15. Nessa definição, utilizamos o esquema

$$\forall(x)A_1(x) \rightarrow B_1(x) \tag{5}$$

para representar qualquer fórmula do tipo do esquema (5), em que $A_1(x)$ e $B_1(x)$ são duas fórmulas quaisquer, cuja variável livre é x.

Este tipo de esquemas não tem uma forma tão simples de ser representado no nível-meta como o do caso anterior, sendo, no entanto, possível fazê-lo sem muita dificuldade.

Se tentássemos representar o esquema (5) no nível-meta pelo termo $\forall(x)A_1(x) \rightarrow B_1(x)$, sendo x, A_1 e B_1 variáveis do nível-meta, não obteríamos o mesmo esquema, uma vez que A_1 e B_1 apenas podem ser substituidas por símbolos para que o termo resultante corresponda a uma fórmula do nível-base. Este termo permite apenas como antecedente e consequente da implicação fórmulas atómicas (e unárias) e não qualquer fórmula, isto é, não permite a representação da fórmula $\forall(x)(A(x) \wedge B(x)) \rightarrow C(x)$, que é coberta pelo esquema anterior.

Para permitir a representação destes esquemas, consideramos a existência de uma relação $varsLivres$, que indica quais são as variáveis livres de um termo que representa uma fórmula do nível-base: temos $varsLivres(\Psi, vars)$ quando o conjunto das variáveis livres do termo Ψ é $vars$. As fbfs a adicionar ao contexto, correspondendo aos axiomas próprios necessários para definir esta relação, estão no apêndice C.

[22] Utilizando a notação simplificada, para facilitar a leitura do termo $imply(A, B)$.

Assim, a forma correcta de representar o esquema (5) numa fórmula do nível-meta é:

$$varsLivres(A_1, \{x\}) \rightarrow (varsLivres(B_1, \{x\}) \rightarrow P(\forall(x)A_1 \rightarrow B_1))$$

em que $P(t)$ é uma fórmula qualquer que utiliza o termo t.

Para facilitar a escrita das fórmulas no nível-meta, vamos adoptar a seguinte notação: utilizamos $\Psi\{x_1, \ldots, x_n\}$ para uma variável Ψ que esteja no lugar de uma fórmula num termo, significando que essa variável apenas pode ser instanciada com fórmulas cujas variáveis livres sejam $\{x_1, \ldots, x_n\}$. Escrevemos uma fórmula no nível-meta com uma variável $\Psi\{x_1, \ldots, x_n\}$, $A(\Psi\{x_1, \ldots, x_n\})$ em vez de $(varsLivres(\Psi, \{x_1, \ldots, x_n\}) \rightarrow A(\Psi))$. Assim, a ocorrência do esquema (5) numa fórmula do nível-meta será representada por:

$$\forall(x)A_1\{x\} \rightarrow B_1\{x\}$$

7 A lógica de especificação de preferências LEP

A Lógica de Especificação de Preferências (LEP) vai ter como base a lógica SWMC, especializando a sua linguagem e adicionando algumas regras de inferência que permitem estabelecer a ligação entre o nível-base e o nível-meta.

Nesta secção vamos descrever formalmente a LEP, fazendo uma apresentação da sua linguagem e das suas regras de inferência, apresentando informalmente a semântica à medida que as várias extensões são feitas.

7.1 A linguagem

A linguagem da LEP, que designaremos por $\mathcal{L}_{LEP}$, corresponde ao conjunto das *fórmulas bem-formadas* cujas regras de formação são idênticas às da lógica SWMC. Assim, a linguagem da LEP é idêntica à da SWMC, particularizada em alguns aspectos:

- O conjunto de símbolos de predicado contém o seguinte conjunto de símbolos especiais $\{InBaseContext$, $CurrentBaseContext$, $Sound$, $Plausible$, $Derivable$, $Conceivable$, $ProofTree$, $Preferred$, $MostPreferred\}$, que são utilizados para descrever algumas características das crenças que estão no nível-base. Estes predicados serão introduzidos através de regras de inferência especiais.

- Os termos descritos na secção 6 vão ser interpretados (na semântica) como as fbfs da SWMC que lhes correspondem.

Assumimos a utilização de vários termos na LEP, correspondendo às várias estruturas de dados úteis para descrever alguns objectos, como por exemplo sequências e pares. A sua utilização em lógica de primeira ordem não apresenta nenhuma dificuldade, nem a escrita dos axiomas que definem essas estruturas, por isso não os apresentaremos aqui.

7.2 O registo de dependências no nível-meta

Na LEP, tal como na SWMC, vamos utilizar o conceito de *fbf suportada* para manter o registo das dependências entre as várias fórmulas.

O registo de dependências serve para fazer as alterações no estado de crença de cada fórmula quando se verificarem alterações no "mundo" que está a ser conceptualizado. No caso da LEP, o "mundo" que está a ser conceptualizado são as crenças de um agente representadas em SWMC. Assim, é natural que as fórmulas de uma teoria da LEP dependam do estado do nível-base, que corresponde ao "mundo" que está a ser conceptualizado. Por este motivo, para determinar quais as crenças do agente no nível-meta, consideramos a existência de um contexto fixo do nível-base que designamos por *contexto corrente do nível-base*.

Para que na LEP possamos registar as dependências de uma fórmula em relação ao estado do nível-base, necessitamos de introduzir hipóteses especiais que reflictam esse estado. Assim, consideramos a existência de dois predicados unários especiais: $InBaseContext$ e $CurrentBaseContext$. Informalmente, $InBaseContext(A)$ representa o facto de a fórmula A do nível-base ser uma hipótese no contexto corrente do nível-base; $CurrentBaseContext(\beta)$ representa o facto de o contexto corrente do nível-base ser o contexto β.

O estado do nível-base pode ser caracterizado pelo seu contexto, uma vez que um contexto define os possíveis estados de crença que um agente racional pode ter. Quando o contexto do nível-base sofre alterações, também o estado desse nível se altera — existem fórmulas que deixam de ser acreditadas e/ou outras que passam a sê-lo. Uma vez que as crenças no nível-meta dependem das crenças no nível-base, sempre que existe uma alteração do contexto do nível-base também se deve proceder às alterações necessárias no nível-meta. A forma de conseguir isso é através da inclusão no contexto do nível-meta das hipóteses especiais $InBaseContext(A)$ e $CurrentBaseContext(\beta)$ de acordo com o contexto corrente do nível-base, como veremos na secção 7.5.

7.3 As hipóteses especiais InBaseContext e CurrentBaseContext

A utilização de hipóteses especiais para representar informação sobre o estado do nível-base já foi motivada anteriormente, e a sua utilidade tornar-se-á evidente na definição de algumas das regras de inferência.

O predicado especial $InBaseContext$ permite representar a informação de que uma determinada hipótese do nível-base é acreditada. Esta informação sobre o estado do nível-base, no entanto, não é suficiente para o caracterizar completamente, uma vez que não sabemos nada acerca das hipóteses que não são acreditadas.

O predicado especial $CurrentBaseContext$ vai permitir-nos representar de uma forma completa o estado do nível-base. A fórmula $CurrentBaseContext(\beta)$ representa um estado possível do nível-base em que as únicas hipóteses acreditadas são as existentes em β. Todas as hipóteses que não pertençam a β não são acreditadas.

Uma vez que a hipótese $CurrentBaseContext$ contém toda a informação sobre quais são as hipóteses acreditadas num determinado estado do nível-base, poderíamos considerar a fórmula $InBaseContext(H)$, não como uma hipótese, mas sim como derivada a partir do $CurrentBaseContext(\beta)$ para cada β que contenha H.

Esta alteração, no entanto, teria como consequência que as fórmulas $InBaseContext(H)$ não poderiam aparecer no conjunto de origem de outras fórmulas. Em seu lugar iriam aparecer as fórmulas do tipo $CurrentBaseContext(\beta)$.

Quando temos uma fórmula A que depende de $InBaseContext(H)$, isto significa que devemos acreditar em A quando H é uma hipótese acreditada no nível-base. A substituição de $InBaseContext$ por $CurrentBaseContext$ não nos parece muito adequada sob o ponto de vista epistemológico, uma vez que isso corresponderia a dizer que a crença em A deve ser mantida quando o estado do nível-base é tal que a hipótese H é acreditada, ou então as hipóteses H e H' são acreditadas, ou então as hipóteses H, H' e H'' são acreditadas, $\ldots$, para cada um dos possíveis contextos do nível-base que contêm a hipótese H.

O perigo que a existência simultânea de duas hipóteses que traduzem informação relacionada traz, é a possibilidade de termos informação incoerente: acreditarmos simultaneamente em $CurrentBaseContext(\{H1, H2\})$ e em $InBaseContext(H3)$, em que $H3 \notin \{H1, H2\}$. No nosso caso, como a inclusão destas hipóteses especiais num contexto LEP vai ser feita de uma forma controlada (definição 4 na página 40), este problema não se coloca. Assim, preferimos manter as duas fórmulas como hipóteses, visto considerarmos ser esta a solução epistemologicamente mais adequada.

7.4 As regras de inferência

Para além das regras de inferência da SWMC, a LEP vai ainda ter um conjunto de regras de inferência especiais, que vão permitir inferir fórmulas no nível-meta a partir de informação existente no nível-base. A estas regras chamamos *regras de inferência de ligação*.

As regras de inferência de SWMC indicam como é que uma fbf suportada pode ser obtida a partir de outras fbfs suportadas (à excepção da regra de Introdução da Hipótese), pertencentes também à linguagem. Na LEP, as regras de inferência de ligação geram novas fbfs suportadas a partir de informação do nível-base. Essa informação pode estar representada no nível-base (fórmulas de SWMC, por exemplo), ou fazer parte da meta-linguagem utilizada para falar sobre o nível-base.

De seguida vamos apresentar as regras de inferência da LEP. Para facilitar a leitura do texto, em que ocorrem simultaneamente fbfs suportadas da LEP e da SWMC, vamos adoptar uma notação diferente para a escrita das fbfs suportadas da LEP: em vez de escrevermos $<A$, τ, $\alpha>$, no caso das fbfs suportadas da LEP escrevemos $\ll A$, τ, $\alpha \gg$.

7.4.1 Regras sobre a natureza das crenças no nível-base

A lógica utilizada para modelar o conhecimento do agente ao nível-base — a SWMC — é uma lógica não-monótona. Uma das características destas lógicas é permitirem saltar para conclusões com base em suposições, conhecimento que não é certo, mas que se assume como sendo o que "tipicamente" se verifica, até que surja informação em contrário. Estas conclusões, no entanto, são de natureza diferente das que se obtêm através do raciocínio com base em conhecimento certo. Por este motivo, existe a distinção entre os vários tipos de conclusões: conclusões sólidas, conclusões plausíveis e conclusões concebíveis. Esta distinção é importante para o agente, porque apesar de se pretender que ele utilize as conclusões baseadas em suposições, em certos casos pode ser necessário que se tenha a certeza acerca de uma determinada conclusão.

Para que o agente possa raciocinar sobre os vários tipos de crenças existentes em SWMC, vamos introduzir três predicados:

- $Sound(A)$, que representa o facto de a fbf A ser uma conclusão sólida do contexto corrente do nível-base.[23]

[23] Na LEP, A é um termo que representa a crença do nível-base.

- $Plausible(A)$, que representa o facto de a fbf A ser uma conclusão plausível do contexto corrente do nível-base.
- $Conceivable(A)$, que representa o facto de a fbf A ser uma conclusão concebível do contexto corrente do nível-base.

Embora nos predicados acima apenas seja considerada a fórmula A, a noção de consequência é definida entre um contexto e uma fórmula. O motivo pelo qual não incluimos o contexto nos predicados é porque vamos considerar um contexto corrente do nível-base quando queremos determinar quais são as fórmulas acreditadas no nível-meta. A dependência destes predicados em relação ao estado do nível-base vai-se manifestar no conjunto de origem associado a eles na fbf suportada gerada pelas regras de inferência de introdução respectivas.

Estes predicados vão poder ser introduzidos através das seguintes regras de inferência:

Introdução de Sound

Dada uma fbf suportada do nível-base $<A, \tau, \alpha>$, em que α não contém suposições,[24] podemos inferir $\ll Sound(A), der, \{InBaseContext(H) : H \in \alpha\}\gg$.

Naturalmente, só devemos acreditar que A é uma consequência sólida no nível-base se isso de facto se verifica dado o contexto corrente do nível-base. Por este motivo, introduzimos no conjunto de origem associado à fórmula $Sound(A)$ as hipóteses especiais $InBaseContext(H)$ para cada fórmula H do conjunto de origem α. Assim só acreditamos nesta fórmula se acreditarmos que todas as hipóteses existentes em α (que foram utilizadas na derivação de A) pertencem ao contexto corrente do nível-base.

Introdução de Plausible

Dado um contexto β do nível-base, e uma fbf $A \in \mathcal{L}$, tal que $\beta \vdash_P A$, podemos inferir $\ll Plausible(A), der, \{CurrentBaseContext(\beta)\}\gg$.

Para saber se uma fórmula é uma consequência plausível de um contexto, não basta verificar se existe uma fbf suportada cujo conjunto de origem esteja contido no contexto. Uma fórmula é uma consequência plausível de um contexto, se pertencer a todos os espaços de crenças definidos por esse contexto. Por este motivo, precisamos de utilizar a noção de consequência plausível ($\vdash_P$), definida na lógica SWMC, como premissa para a aplicação da regra de inferência, em vez de utilizarmos características sintácticas de algumas fórmulas. Este

[24] Não foram utilizadas regras de omissão nesta derivação de A.

é um dos casos em que as premissas da regra de inferência não são outras fbfs suportadas, mas sim condições que apenas eram descritas ao nível da meta-linguagem.

Tal como no caso anterior, só devemos acreditar em $Plausible(A)$ se de facto A for uma consequência plausível do contexto corrente do nível-base. No entanto, para que isso se verifique não é suficiente considerarmos apenas o conjunto de origem de uma fbf suportada para A: se por um lado esse conjunto pode não conter apenas hipóteses (e por isso pode não estar contido no contexto), por outro, a noção de consequência plausível em SWMC é uma noção não-monótona, logo, se acrescentarmos ou retirarmos alguma hipótese do contexto, não temos maneira de saber se a fbf deixou ou não de ser acreditada. Por isso, o facto de acreditarmos em todas as hipóteses que pertençam ao seu conjunto de origem no nível-base não nos garante que A seja uma conclusão plausível ou sequer concebível de um contexto que as contenha.

Assim, a dependência desta fórmula em relação ao estado do nível-base é mais forte: apenas acreditamos na fórmula se o contexto corrente, ou seja, o estado do nível-base, for o mesmo que o utilizado na regra de inferência. No caso anterior, com consequências sólidas, acreditamos na fórmula $Sound(A)$ em qualquer contexto que contenha o conjunto α.

Introdução de Conceivable

> Dado um contexto β do nível-base, e uma fbf $A \in \mathcal{L}$, tal que $\beta \vdash_C A$, podemos inferir $\ll Conceivable(A),\ der,\ \{CurrentBaseContext(\beta)\} \gg$.

Esta regra é em tudo idêntica à anterior, utilizando agora o conceito de consequência concebível.

7.4.2 Regras sobre a prova de uma fbf

As fbfs suportadas da lógica SWMC permitem efectuar um registo de dependências entre as fórmulas, suportando assim um sistema de revisão de crenças (SRC). De acordo com a nomenclatura usual em SRC's, a SWMC suporta um sistema de revisão de crenças do tipo ATMS.

A regra de inferência de introdução do *Derivable* vai permitir obter informação, no nível-meta, acerca da derivabilidade ($\vdash_{SWMC}$) de uma fbf do nível-base, a partir de um conjunto de fbfs do nível-base. Esse conjunto de fbfs corresponde na realidade a um superconjunto do conjunto de origem de uma fbf suportada para essa fbf.

Introdução de Derivable

Dada uma fbf suportada do nível-base $<A,\ \tau,\ \alpha>$, podemos inferir $\ll Derivable(A, \gamma)$, der, $\{\}\gg$, para qualquer $\gamma \subset \mathcal{L}$ tal que $\alpha \subseteq \gamma$.

O facto de o conjunto de origem da fbf suportada inferida por esta regra ser vazio significa que a fbf $Derivable(A, \gamma)$ é sempre acreditada, qualquer que seja o contexto. Isto não significa, no entanto, que a fbf A é sempre acreditada, uma vez que, de acordo com a semântica pretendida para este predicado, o que se está a afirmar é que a fbf A é derivável do conjunto de fbfs γ, mas nada se diz quanto a ela ser acreditada ou não. Se existe uma fbf suportada do nível-base $<A,\ \tau,\ \alpha>$, é porque essa fbf suportada foi derivada no nível-base a partir de um conjunto de outras fbfs suportadas cujas fbfs são as existentes em α. Neste caso, a fbf A é derivável em SWMC a partir de qualquer superconjunto γ de α, uma vez que a noção de derivabilidade em SWMC é monótona.[25]

Uma outra abordagem aos SRC's consiste nos sistemas baseados em justificações (JTMS), nos quais em cada crença é mantido o registo de quais foram as crenças imediatamente utilizadas na sua derivação e não quais as que estão na sua base.

Com a utilização de um JTMS, temos a possibilidade de ir seguindo para trás as dependências de uma crença, através das suas justificações, e determinar a sequência de crenças utilizadas para chegar a ela. O conhecimento dessa sequência pode constituir mais informação acerca do motivo porque se mantém uma determinada crença. Uma vez que, para especificar preferências ao nível-meta, precisamos de informação acerca das crenças do nível-base, este tipo de informação sobre a forma como cada crença foi obtida pode ser importante para a construção de uma teoria de preferências.

Como na lógica SWMC não existem mecanismos para fazer o registo da sequência de fbfs suportadas utilizadas numa derivação, temos que introduzir algumas definições que nos permitam lidar com esta informação:

Definição 2 Passo de prova em SWMC
Um passo de prova em SWMC, para a fbf suportada $\mathcal{A} = <A,\ \tau,\ \alpha>$, é um par $<[A_1, \ldots, A_n],\ A>$ em que $[A_1, \ldots, A_n]$ é uma sequência de fbfs, tais que existe uma sequência $[\mathcal{A}_1, \ldots, \mathcal{A}_n]$ de fbfs suportadas onde $wff(\mathcal{A}_i) = A_i, i = 1, \ldots, n$ e estas fbfs suportadas são as premissas para a aplicação de uma regra de inferência

[25] Repare-se que o carácter não-monótono da lógica SWMC se manifesta na noção de consequência.

de SWMC cuja conclusão é $\mathcal{A}$. No caso de $\tau = hyp$, essa sequência é a sequência vazia [ ].

Esta definição corresponde à aplicação de uma regra de inferência, indicando as premissas e a conclusão da regra aplicada. Através desta definição, temos acesso à justificação da fbf suportada: quais são as fórmulas que permitem a sua derivação. Esta definição vai agora poder ser usada para a definição de árvore de prova, que vai permitir caracterizar a derivação de uma fbf suportada.

Definição 3 Árvore de prova em SWMC

Uma árvore de prova em SWMC para a fbf suportada $\mathcal{A} = <A, \tau, \alpha>$, é:

- o passo de prova em SWMC para $\mathcal{A}$: $<[\,], A>$, se $\tau = hyp$
- um par $<[Arv_1, \ldots, Arv_n], A>$ em que $[Arv_1, \ldots, Arv_n]$ é uma sequência de árvores de prova com raízes[26] $[A_1, \ldots, A_n]$, e tal que $<[A_1, \ldots, A_n], A>$ é um passo de prova em SWMC para $\mathcal{A}$, se $\tau \neq hyp$.

O tipo de informação fornecido por uma árvore de prova pode ser ilustrado melhor com a utilização de um exemplo. O conceito de árvore de prova é muito utilizado em sistemas de dedução natural, e o exemplo que vamos apresentar é uma derivação em SWMC utilizando uma das notações mais usuais nestes sistemas. A partir das hipóteses de que $Tweety$ é uma ave e de que "Tipicamente as aves voam" vamos poder inferir que $Tweety$ voa (onde $D_{A \to V} = \triangledown(x) Ave(x) \to Voa(x)$):

$$\dfrac{\dfrac{\triangledown(x)Ave(x) \to Voa(x) \qquad \dfrac{\triangledown(x)Ave(x) \to Voa(x)}{Applicable(D_{A\to V}, Tweety)}}{Ave(Tweety) \to Voa(Tweety)} \qquad Ave(Tweety)}{Voa(Tweety)}$$

Na realidade, esta prova deveria ser feita com fbfs suportadas e não com fbfs, uma vez que as regras de inferência da SWMC manipulam fbfs suportadas. Não utilizámos fbfs suportadas na prova acima por uma questão de facilidade de leitura, omitindo assim o rótulo e o conjunto de origem. Para além disso, a nossa definição de prova apenas considera as fbfs e não as fbfs suportadas.

[26] A raíz de uma árvore de prova é o segundo elemento do par.

Um passo de prova para a fbf suportada correspondente à fbf $Voa(Tweety)$, no exemplo acima, é $<[Ave(Tweety) \rightarrow Voa(Tweety), Ave(Tweety)], Voa(Tweety)>$. A árvore de prova para a mesma fbf suportada é

$$
\begin{array}{l}
<[<[<[\,], \bigtriangledown(x)Ave(x) \rightarrow Voa(x)>, \\
\quad\quad <[<[\,], \bigtriangledown(x)Ave(x) \rightarrow Voa(x)>], \\
\quad\quad\quad Applicable(\bigtriangledown(x)Ave(x) \rightarrow Voa(x), Tweety)>], \\
\quad\quad Ave(Tweety) \rightarrow Voa(Tweety)>, \\
\quad <[\,], Ave(Tweety)>], \\
\quad Voa(Tweety)>
\end{array}
$$

A partir destas definições de prova, vamos poder definir uma nova regra de inferência que nos permite obter informação no nível-meta acerca da forma como as derivações no nível-base são efectuadas.

Introdução de ProofTree

Dada uma árvore de prova em SWMC, $\mathcal{P}$, para uma fbf suportada $\mathcal{A} = <A, \tau, \alpha>$ podemos inferir $\ll ProofTree(A, \mathcal{P}), der, \{\}\gg$

Tal como no caso anterior, da regra de introdução do $Derivable$, a fbf suportada derivada tem como conjunto de origem o conjunto vazio, ou seja, é acreditada em qualquer contexto.

A fbf $ProofTree(A, \mathcal{P})$ significa que $\mathcal{P}$ é uma árvore de prova em SWMC para uma fbf suportada $\mathcal{A}$, tal que $wff(\mathcal{A}) = A$, o que é sempre verdade ou sempre falso, uma vez que as regras de inferência não se alteram. Assim, se através desta regra de inferência pudermos introduzir esta fbf, ela será sempre acreditada.

7.4.3 Regras sobre as ordens de preferência

A semântica pretendida para o predicado especial $Preferred$ é a de que a fbf $Preferred(A, B)$ significa que o valor da crença A é pelo menos tão grande como o valor da crença B. Através deste predicado, estabelecemos uma relação de ordem entre os valores das várias crenças que o agente mantém no nível-base. Uma vez que esta relação é uma relação de ordem parcial, verifica as três propriedades características de ordens parciais: transitividade, reflexividade e anti-simetria.

Geralmente, fala-se da relação de ordem entre crenças, em vez de entre os seus valores. Assim, se aplicássemos a propriedade de anti-simetria a duas crenças que são preferidas uma em relação à outra, diríamos que elas eram iguais, mas, na realidade, a igualdade verifica-se entre os seus valores. Tendo isto em consideração, continuaremos a falar de ordem entre crenças como sinónimo de ordem entre os valores das crenças.

Naturalmente, as três propriedades das ordens parciais têm que se verificar numa teoria da LEP que utilize o predicado $Preferred$. Para expressar estas propriedades bastaria introduzir nas teorias LEP três axiomas simples que as descrevessem. No entanto, como toda a formulação da SWMC e da LEP se baseia num sistema de dedução natural, sem axiomas e só com regras de inferência, vamos também introduzir estas propriedades do predicado $Preferred$ através de regras de inferência.

Transitividade do Preferred

De $\ll Preferred(A, B),\ \tau_1,\ \alpha_1 \gg$ e de $\ll Preferred(B, C),\ \tau_2,\ \alpha_2 \gg$ podemos inferir $\ll Preferred(A, C),\ der,\ \alpha_1 \cup \alpha_2 \gg$.

Esta regra de inferência traduz a propriedade da transitividade da ordem parcial: se uma fbf A é pelo menos tão importante como uma outra fbf B, e essa por sua vez é pelo menos tão importante como uma fbf C, então a fbf original A é também pelo menos tão importante como a fbf C.

Reflexividade do Preferred

Dada uma fbf do nível-base $A \in \mathcal{L}$, podemos inferir $\ll Preferred(A, A),\ der,\ \{\} \gg$.

Neste caso, limitamo-nos a dizer que uma crença é pelo menos tão importante como ela própria. Repare-se que a fbf suportada derivada tem como conjunto de origem o conjunto vazio, ou seja, $Preferred(A, A)$ é sempre acreditada, desde que A seja uma fbf do nível-base.

Anti-Simetria do Preferred

No caso da propriedade de Anti-Simetria, não existe nenhum constrangimento adicional a impôr ao predicado especial $Preferred$. O facto de termos uma fbf A que é pelo menos tão importante como a fbf B e esta por sua vez tem a mesma relação com A, significa que elas são equivalentes, isto é, têm o mesmo valor. Isto é representado pelo facto de termos simultaneamente $Preferred(A, B)$ e $Preferred(B, A)$, não sendo necessário introduzir outra fbf para representar este facto.

Para além destas regras sobre o predicado $Preferred$, precisamos ainda de estabelecer regras sobre o predicado $MostPreferred$. Intuitivamente, este predicado unário permite indicar algumas crenças que são consideradas como máximos da ordem de preferência, isto é, são crenças nas quais temos a confiança máxima, nunca podem deixar de ser acreditadas. Esta situação pode ocorrer em domínios onde algumas das crenças do agente sejam "leis" ou definições, isto é, não podem ser violadas, e portanto, em caso de revisão das crenças, estas nunca deverão ser abandonadas. Para garantirmos isso, tornamo-las máximos da ordem de preferência. As proposições que correspondem a definições, por exemplo "Um solteiro é uma pessoa que nunca casou", são boas candidatas a serem colocadas como máximos da ordem de preferência.

Introdução do MostPreferred

De $\ll MostPreferred(A),\ \tau_1,\ \alpha_1 \gg$ e $\ll Preferred(B, A),\ \tau_2,\ \alpha_2 \gg$, podemos inferir $\ll MostPreferred(B),\ der,\ \alpha_1 \cup \alpha_2 \gg$.

Neste caso, quando temos uma fbf A que é das mais preferidas e também uma outra fbf B que é tão ou mais importante do que ela, como B não pode ser mais importante do que A (por esta ser um máximo da ordem parcial), B é também $MostPreferred$.

7.4.4 Regras sobre a negação dos predicados especiais

As regras de inferência introduzidas na LEP são de natureza diferente das regras de inferência existentes em SWMC. Enquanto que na SWMC as regras de inferência descrevem manipulações sintácticas sobre a estrutura das fórmulas compostas (com conectivas lógicas), permitindo introduzir ou eliminar símbolos lógicos, na LEP as regras de inferência introduzem fórmulas atómicas através de símbolos de predicado especiais, com uma semântica pré-definida.

As regras de inferência de ligação apresentadas para a LEP permitem apenas introduzir fórmulas atómicas. No entanto, a negação dessas fórmulas pode também ser útil na construção de uma teoria de preferências. Em SWMC, a negação de uma fórmula é obtida através da regra de *Introdução da negação* que permite, a partir de uma contradição derivada de um conjunto de fbfs, introduzir a negação de uma qualquer hipótese utilizada na derivação.

Assim, para podermos obter a negação de uma fórmula atómica constituida por um dos predicados especiais da LEP, por exemplo $\neg Sound(A)$, teríamos que derivar uma contradição a partir da hipótese $Sound(A)$. Uma vez que não temos nenhuns axiomas acerca dos predicados especiais, a única forma de obter uma contradição a partir de $Sound(A)$ consiste em considerar na derivação da contradição um conjunto de hipóteses inconsistentes com

$Sound(A)$. No entanto, e uma vez que os predicados especiais já têm uma semântica pré-definida, não é natural que um contexto, que represente o conhecimento sobre preferências do agente, contenha hipóteses que sejam inconsistentes com $Sound(A)$.

Na realidade, os predicados especiais servem para caracterizar aspectos das crenças que o agente mantém no nível-base e, uma vez que as fórmulas não negadas são obtidas tendo em conta o estado desse nível, também a sua negação o deve ser.

Assim vamos ter regras de inferência também para a introdução dos predicados negados:

Introdução de ¬InBaseContext

Dado um contexto β e uma hipótese H do nível-base, tal que $H \notin \beta$, podemos inferir $\ll \neg InBaseContext(H), der, \{CurrentBaseContext(\beta)\} \gg$.

A regra de "Introdução de ¬InBaseContext" permite inferir que uma determinada hipótese H do nível-base não pertence ao contexto corrente desse nível. Obviamente, isso acontece em qualquer contexto que não contenha H. Por este motivo, o conjunto de origem da fbf suportada inferida contém a fbf $CurrentBaseContext(\beta)$ para um contexto β qualquer que não contenha H. Repare-se que vamos poder inferir uma fbf suportada para $\neg InBaseContext(H)$, para cada contexto β que não contenha H.

Introdução de ¬CurrentBaseContext

Dado um contexto β do nível-base, podemos inferir, para qualquer contexto $\beta' \neq \beta$, $\ll \neg CurrentBaseContext(\beta'), der, \{CurrentBaseContext(\beta)\} \gg$.

Sempre que um contexto β' seja diferente do contexto corrente do nível-base, podemos inferir $\neg CurrentBaseContext(\beta')$, uma vez que na determinação da satisfação das fórmulas do nível-meta fixamos um contexto do nível-base, e apenas podemos ter um contexto corrente de nível-base em cada instante. Assim, o conjunto de origem da fbf suportada inferida é idêntico ao da regra anterior.

Introdução de ¬Sound

Dado um contexto β do nível-base, e uma fbf $A \in \mathcal{L}$, tal que $\beta \not\vdash A$, podemos inferir $\ll \neg Sound(A), der, \{CurrentBaseContext(\beta)\} \gg$.

Introdução de ¬Plausible

Dado um contexto β do nível-base, e uma fbf $A \in \mathcal{L}$, tal que $\beta \not\vdash_P A$, podemos inferir $\ll \neg Plausible(A), der, \{CurrentBaseContext(\beta)\} \gg$.

Introdução de ¬Conceivable

Dado um contexto β do nível-base, e uma fbf $A \in \mathcal{L}$, tal que $\beta \not\vdash_C A$, podemos inferir $\ll\neg Conceivable(A)$, der, $\{CurrentBaseContext(\beta)\}\gg$.

A definição destas regras não apresenta nenhuma dificuldade, uma vez que inferimos a negação de cada uma das fórmulas atómicas $Sound(A)$, $Plausible(A)$ e $Conceivable(A)$, com conjunto de origem $\{CurrentBaseContext(\beta)\}$ sempre que A não seja uma conclusão do tipo correspondente do contexto β.

Esta definição é em tudo idêntica à do caso das fórmulas não negadas, com excepção da regra de "Introdução do Sound". No caso dessa regra, o conjunto de origem da fbf suportada inferida não era $\{CurrentBaseContext(\beta)\}$, mas sim composto por hipóteses do tipo *In-BaseContext*(H) para cada hipótese H do nível-base usada na derivação sólida de A. A ideia subjacente a essa definição era a de que se deveria acreditar em $Sound(A)$ sempre que o contexto corrente do nível-base contivesse as hipóteses utilizadas na derivação de A.

No caso da negação, no entanto, isso não se verifica, uma vez que A pode não ser consequência sólida de um determinado contexto β, mas ser de um contexto que o contém. Aquilo que podemos dizer, neste caso, é que A não é consequência sólida de nenhum contexto contido em β. De acordo com a noção de derivabilidade utilizada para a LEP (idêntica à da SWMC), não temos possibilidade de criar um conjunto de origem que permita que a fbf suportada correspondente seja derivável de qualquer conjunto de fórmulas contido noutro conjunto (apenas o contrário). Por este motivo, a dependência de $\neg Sound(A)$ em relação ao estado do nível-base, imposta pela regra de "Introdução de ¬Sound", é mais forte: apenas acreditamos nessa fbf se o contexto corrente do nível-base for aquele para o qual a regra foi aplicada e do qual sabemos que A não é consequência sólida.

Introdução de ¬Derivable

Dada uma fbf $A \in \mathcal{L}$ e $\alpha \subset \mathcal{L}$, tal que não existe nenhuma fbf suportada $<A, \tau, \alpha'>$, em que $\alpha' \subseteq \alpha$, podemos inferir $\ll\neg Derivable(A, \alpha)$, der, $\{\}\gg$.

Introdução de ¬ProofTree

Dada uma fbf $A \in \mathcal{L}$, podemos inferir, para qualquer árvore $\mathcal{P}$ que não constitua uma árvore de prova em SWMC para uma fbf suportada $\mathcal{A}$ qualquer com $wff(\mathcal{A}) = A$, $\ll\neg ProofTree(A, \mathcal{P})$, der, $\{\}\gg$.

Estas duas regras são em tudo idênticas às das correspondentes não negadas, pelo que não necessitam de mais explicações.

Não definimos regras para a introdução da negação das fórmulas atómicas construidas pelos predicados especiais $Preferred$ e $MostPreferred$, porque estes não correspondem a predicados que forneçam informação sobre o estado do nível-base. São especiais na medida em que a sua interpretação é fixada na lógica, mas a sua negação fica ao cuidado de quem estiver a modelar o conhecimento de preferências do agente.

7.5 Derivabilidade, contextos e espaços de crenças ao nível-meta

A noção de derivabilidade em LEP é idêntica à de SWMC: dizemos que uma fbf A é derivável de um conjunto de fbfs α, o que escrevemos como $\alpha \vdash_{LEP} A$, sse existe uma fbf suportada da LEP, $\mathcal{A}$, tal que $wff(\mathcal{A}) = A$ e $os(\mathcal{A}) \subseteq \alpha$.

A noção de derivabilidade não precisa de sofrer alterações porque, como vamos ver de seguida, a dependência das crenças do nível-meta em relação ao estado do nível-base vai-se manifestar através da alteração do contexto de nível-meta.

Em SWMC, para podermos decidir se uma fórmula deve ser acreditada ou não, dado um conjunto de fbfs, não nos basta utilizar a noção de derivabilidade — é necessário utilizar a noção de consequência. Os vários tipos de consequência existentes em SWMC são definidos a partir da construção dos espaços de crenças, um processo efectuado em três passos a partir de um conjunto de hipóteses, o contexto.

Na LEP, para podermos decidir se uma fórmula deve ser acreditada ou não, para além do que acontece na SWMC, temos necessidade de conhecer o estado do nível-base. Através de uma alteração criteriosa do contexto, podemos manter sem alterações o processo de construção dos espaços de crenças, bem como os vários tipos de consequências.

De modo a reflectir o estado do nível-base no nível-meta, vamos utilizar as hipóteses especiais já referidas $InBaseContext$ e $CurrentBaseContext$.

A construção dos espaços de crenças a partir de um conjunto de hipóteses na LEP pressupõe que é dado um contexto de nível-base fixo que caracteriza o estado desse nível. A definição de contexto para a LEP vai ter que considerar esse contexto de nível-base, ou seja, a informação do contexto do nível-base tem que ser expressa no contexto de nível-meta. A forma de exprimir essa informação no nível-meta é através dos predicados especiais $InBaseContext$ e $CurrentBaseContext$:

Definição 4 Contexto do nível-meta
Consideremos um contexto do nível-base β_B, e um conjunto de hipóteses do nível-meta α que não contenha nenhuma hipótese especial $InBaseContext(H)$

ou $CurrentBaseContext(\beta)$, para algum $H \in \mathcal{L}$, ou $\beta \subset \mathcal{L}$. O contexto do nível-meta β_M é o conjunto que corresponde à reunião das hipóteses de nível-meta α com as hipóteses especiais $InBaseContext(H)$, para cada $H \in \beta_B$, e $Current\text{-}BaseContext(\beta_B)$:

$$\beta_M = \alpha \cup \{InBaseContext(H) : H \in \beta_B\} \cup \{CurrentBaseContext(\beta_B)\}$$

Com esta construção do contexto de nível-meta, o processo de obter os espaços de crenças definido na SWMC não precisa de sofrer alterações. Sempre que ocorrer alguma alteração no estado do nível-base, isso deve-se necessariamente a uma alteração do contexto, logo, para repercutir essas alterações no nível-meta basta alterar o contexto de nível-meta de acordo com a definição anterior.

8 Especificação de preferências entre suposições

Pensamos que é importante que o utilizador do sistema de revisão de crenças possa estabelecer preferências, não só entre as hipóteses ($\mathcal{L}_{FOL} \cup \mathcal{L}_D \cup \mathcal{L}_E$) da base de crenças do agente, mas também entre as suposições ($\mathcal{L}_A$) que possam ser geradas a partir das regras de omissão.

Para ilustrar o nosso ponto de vista, vamos apresentar um exemplo de um problema e a sua resolução usando o SRC descrito na secção 4, seguida da solução que achamos mais correcta, mas agora usando a LEP e permitindo a especificação de preferências entre suposições.

8.1 Apresentação do problema

Suponhamos que vamos usar o SRC, e que codificámos em SWMC o conhecimento acerca do exemplo das reuniões (exemplo 1 da página 18) da seguinte forma:

Exemplo 3 (Exemplo das reuniões em SWMC)

$$fbf1 : \bigtriangledown(p,t)[(vaiReuniao(p) \wedge prefereReunioes(p,t)) \rightarrow ocorreReuniao(t)]$$
$$fbf2 : ocorreReuniao(Manha) \leftrightarrow \neg ocorreReuniao(Tarde)$$
$$fbf3 : general(Guerra)$$
$$fbf4 : capitao(Capucho)$$
$$fbf5 : prefereReunioes(Guerra, Manha)$$
$$fbf6 : prefereReunioes(Capucho, Tarde)$$
$$fbf7 : vaiReuniao(Guerra)$$
$$fbf8 : vaiReuniao(Capucho)$$

Consideremos que estas são as fbfs que estão no nosso contexto. A partir dele, a SWMC gera dois núcleos, a que vão corresponder dois espaços de crenças: um em que a reunião é de manhã e outro em que é de tarde. Nestas condições, o sistema de revisão de crenças não é capaz de preferir um espaço a outro, pois não tem a informação acerca de preferências para o guiar na sua escolha.

No entanto, intuitivamente nós preferimos o espaço de crenças em que a reunião é de manhã, que é o que está de acordo com aquilo que o general prefere, porque sabemos que um general tem mais poder hierárquico do que um capitão.

Suponhamos agora que vamos tentar fazer com que o sistema escolha um dos espaços de crenças através da especificação de uma ordem de preferências que nos diz que as vontades dos generais são preferidas em relação às dos capitães. Mas entre que regras é que vamos especificar essa preferência? Dizer que preferimos a $fbf3$ em relação à $fbf4$, ou a $fbf7$ em relação à $fbf8$, não traduz de forma alguma a noção de preferência que pretendemos representar (não preferimos saber que "o Guerra é um general" a saber que "o Capucho é um capitão"). Uma outra alternativa seria dizer que a $fbf5$ é preferida em relação à $fbf6$ (que é mais importante o facto de o Guerra preferir as reuniões de manhã do que o Capucho preferir as reuniões de tarde). Mas qualquer um destes pares de fbfs, por todas elas serem hipóteses do contexto, está em todos os espaços de crenças gerados a partir dele, e como a escolha do espaço de crenças preferido é feita apenas com base nas suposições que estão num e não estão no outro, a preferência entre estas fbfs não vai servir para nada.

Para podermos determinar como é que a escolha de um espaço de crenças pode ser feita neste caso, vamos olhar para as diferenças existentes entre os dois núcleos gerados por este contexto. Os núcleos são:

$$\Sigma_1 = \beta \cup \{Applicable(fbf1, Guerra, Manha)\}$$
$$\Sigma_2 = \beta \cup \{Applicable(fbf1, Capucho, Tarde)\}$$

em que β é o contexto contendo as hipóteses $fbf1, \ldots, fbf8$. Como se pode observar, a única diferença existente entre os dois núcleos é que num deles a regra de omissão $fbf1$ é aplicada a uns símbolos e no outro é aplicada a outros. Para preferirmos o núcleo Σ_1 ao núcleo Σ_2, é necessário que exista no SRC uma ordem de preferência em β, $\leq_\beta$, tal que $Applicable(fbf1, Capucho, Tarde) <_\beta Applicable(fbf1, Guerra, Manha)$.

Uma vez que as ordens de preferência apenas podem ser estabelecidas entre hipóteses, o SRC induz uma preferência entre suposições segundo a seguinte definição [Cravo 1994]:

Definição 5 Preferência induzida entre suposições
Dado um contexto β, uma ordem $\leq_\beta$ entre as hipóteses de β, e duas suposições $A_1 = Applicable(D_1, c_1)$ e $A_2 = Applicable(D_2, c_2)$ pertencentes a algum dos núcleos de β, se $D_2 \leq_\beta D_1$, então dizemos que A_1 é preferida a A_2 ($A_2 \leq_\beta A_1$).

No entanto, neste caso as duas suposições dizem respeito à mesma regra de omissão, não sendo possível, segundo esta definição, preferir um dos núcleos.

8.2 Resolução do problema em SWMC

Para sermos capazes de resolver este problema em SWMC, temos que alterar a maneira como representámos o conhecimento acerca do domínio, de forma a podermos agora ter duas regras de omissão em que uma é preferida à outra. Para isso, podemos substituir a $fbf1$ por duas regras de omissão:

$$fbf1a : \bigtriangledown(p,t)[general(p) \rightarrow ((vaiReuniao(p) \wedge prefereReunioes(p,t)) \rightarrow ocorreReuniao(t))]$$
$$fbf1b : \bigtriangledown(p,t)[capitao(p) \rightarrow ((vaiReuniao(p) \wedge prefereReunioes(p,t)) \rightarrow ocorreReuniao(t))]$$

Assim, os núcleos gerados seriam:

$$\Sigma_1' = \beta \cup \{Applicable(fbf1a, Guerra, Manha)\}$$
$$\Sigma_2' = \beta \cup \{Applicable(fbf1b, Capucho, Tarde)\}$$

pelo que bastaria a existência de uma ordem parcial em que $fbf1b \leq fbf1a$ para que o núcleo Σ_1' fosse preferido a Σ_2'.

8.3 Resolução do problema usando preferências entre suposições

A abordagem anterior, no entanto, não corresponde a uma forma natural de representar o nosso conhecimento, uma vez que repetimos a mesma regra de omissão apenas para permitir a representação das preferências. Julgamos que a representação desse conhecimento deve estar ao nível-meta, sendo estabelecida aí uma preferência entre as duas suposições dos núcleos Σ_1 e Σ_2. Intuitivamente, pensamos que faz sentido preferir a aplicação de uma regra de omissão a uma constante c em relação a outra c', significando com isso que é mais provável que a conclusão da regra se verifique usando a constante c do que a c'.

Com a especificação de preferências entre suposições, para além de hipóteses, podemos estabelecer uma ordem de preferência $\leq$ em que

$$Applicable(fbf1, Capucho, Tarde) \leq Applicable(fbf1, Guerra, Manha)$$

permitindo, desta forma, a escolha do núcleo Σ_1 tal como o desejado.

Por permitirmos a especificação de preferências entre suposições, é necessário substituir a definição 5 por:

Definição 6 Ordem de preferências estendida
Dado um contexto β, e uma ordem $\leq_\beta$ entre as hipóteses de β e as suposições de algum núcleo de β, a ordem de preferências estendida, $\leq'_\beta$, é a extensão de $\leq_\beta$ com as preferências $A_1 \leq'_\beta A_2$, em que $A_1 = Applicable(D_1, c_1)$ e $A_2 = Applicable(D_2, c_2)$ pertencem a algum dos núcleos de β, e são tais que $A_1 \not\leq_\beta A_2$ e $A_2 \not\leq_\beta A_1$ e $D_1 \leq_\beta D_2$.

Na definição 5 as preferências entre as suposições eram induzidas a partir das regras de omissão que lhes correspondiam. Agora, as preferências entre suposições apenas são induzidas se não existir uma ordem que as relacione directamente.

A ideia subjacente a esta definição é a de que o conhecimento (de nível-meta) expresso através de uma relação de preferência explícita entre duas suposições, é mais específico do que o conhecimento expresso através da relação de preferência entre as regras de omissão correspondentes a essas suposições. Por exemplo, suponhamos que o contexto continha as hipóteses $fbf1a$ e $fbf1b$, em vez da hipótese $fbf1$. Neste caso os núcleos definidos pelo contexto são Σ_1'

e Σ_2'. Suponhamos ainda que temos uma ordem de preferência, $\leq'$, em que $fbf1b \leq' fbf1a$, e também $Applicable(fbf1a, Guerra, Manha) \leq' Applicable(fbf1b, Capucho, Tarde)$. Esta segunda relação resulta do facto de o capitão *Capucho* ter uma consulta médica inadiável de manhã. A especificação da ordem de preferência explícita entre as duas suposições pretende, precisamente, representar esta informação de forma a que o núcleo Σ_2' seja o preferido. No entanto, segundo a definição 5, podemos induzir $Applicable(fbf1b, Capucho, Tarde) \leq' Applicable(fbf1a, Guerra, Manha)$, fazendo com que o valor das duas suposições seja o mesmo segundo a ordem $\leq'$, e não permitindo a escolha de nenhum núcleo.

Por este motivo, na definição 6, apenas induzimos uma relação de preferência entre duas suposições a partir das suas regras de omissão, se entre essas duas suposições não existir nenhuma preferência explícita, e logo mais específica. De acordo com esta definição, para preferirmos os núcleos consideramos, em vez de $\leq_\beta$, $\leq'_\beta$.

9 Os espaços de crenças da LEP e a TRC

A SWMC permite que um agente represente e raciocine sobre conhecimento de *senso comum*, em que existe informação contraditória e sempre em mudança. A teoria de revisão de crenças (TRC) apresentada em [Cravo 1993*a*] descreve as alterações que as crenças do agente devem sofrer quando ele é confrontado com esse tipo de informação.

Apesar de as alterações a efectuar nas crenças serem condicionadas por alguns princípios de racionalidade gerais, essas alterações são efectuadas tendo em linha de conta conhecimento adicional existente sobre as crenças. Esse conhecimento é expresso através de ordens parciais, que estabelecem uma relação entre a importância das várias crenças do agente. A TRC permite a existência de várias ordens parciais sobre as mesmas crenças (que podem ser contraditórias entre si), e ainda a existência de uma ordem parcial que relaciona as várias ordens parciais, indicando qual a sua importância relativa.

A definição da LEP foi feita considerando que a informação expressa nas ordens parciais da TRC constitui conhecimento que um agente deve ter sobre as suas crenças, e sobre o qual pode e deve raciocinar. A criação de um conjunto de fórmulas da LEP serve para representar esse conhecimento.

Os dois tipos de ordens parciais existentes na TRC — ordens sobre as crenças e ordem sobre as ordens — são de natureza muito diferente. Enquanto as ordens de preferência sobre as crenças representam informação que julgamos ser dependente do domínio, que pode ser capturada através de regras gerais e normalmente associada a uma fonte de informação, a ordem sobre as ordens representa a preferência entre essas fontes de informação, não sendo muito natural que se possa descrever essa preferência de uma forma mais sucinta do que a sua simples enumeração. O objectivo da LEP é especificar as ordens de preferência entre crenças, e é apenas delas que iremos falar em seguida.

9.1 Tradução das fbfs da LEP para ordens da TRC

Pretendemos determinar como é que as ordens parciais sobre as crenças são obtidas a partir de um conjunto de fórmulas na LEP. Considerando a interpretação dada ao predicado especial $Preferred$, um conjunto de fórmulas do tipo $Preferred(A, B)$ pode definir uma ordem parcial da seguinte forma: para cada instância de $Preferred(A, B)$ existente nesse conjunto, a ordem parcial correspondente contém o par (B, A), o que podemos escrever como $B \leq A$. Se considerarmos que o conjunto de fórmulas é fechado para as regras de inferência da LEP, isto especifica uma ordem parcial, uma vez que as regras de inferência de "Transitividade do $Preferred$" e "Reflexividade do $Preferred$" garantem que o predicado especial $Preferred$ representa uma relação que corresponde a uma ordem parcial. Com esta abordagem podemos especificar apenas uma ordem parcial. Para que tenhamos as várias ordens parciais, precisamos de ter vários conjuntos de fórmulas LEP, cada um definindo a sua ordem.

Uma possibilidade de obter os vários conjuntos de fórmulas LEP, que corresponderiam às várias ordens parciais, é considerar os espaços de crenças definidos por um contexto-meta da LEP. Na realidade, uma vez que a LEP é uma lógica não-monótona,[27] a utilização de regras de omissão num contexto pode dar origem a vários espaços de crenças. Em cada um desses espaços de crenças vão existir várias fórmulas do tipo $Preferred(A, B)$ e, uma vez que cada espaço de crenças é fechado para as regras de inferência do $Preferred$, pode-se extrair de cada um uma ordem parcial.

Apesar de com isto conseguirmos obter os vários conjuntos de fórmulas LEP de que precisamos, esta abordagem não nos parece epistemologicamente muito adequada. O problema é que, para quem está a modelar o conhecimento das preferências que o agente tem em relação às suas crenças, os vários espaços de crenças gerados a partir do contexto-meta não são evidentes. Se considerarmos que uma ordem parcial corresponde a um conjunto de preferências que provêm de uma forma coerente de alguma fonte de informação, uma determinada ordem parcial deve conter conhecimento que à partida é independente do das restantes ordens. Esta é, aliás, uma das razões porque na TRC existe a possibilidade de definir uma ordem entre as

[27] Essencialmente idêntica à SWMC.

ordens de preferência. Uma vez que, a partir de um contexto, os espaços de crenças gerados são muito semelhantes e não são intencionalmente controlados por quem está a fazer a modelação, a criação de uma ordem parcial a partir de cada um deles não permite obter ordens de preferência muito diferentes entre si, nem saber quantas ordens vão existir, o que elimina a possibilidade de definir uma ordem entre elas.

Por estes motivos, a partir de um contexto-meta, apenas vamos considerar a extracção de uma ordem parcial. Como um contexto pode gerar vários espaços de crenças, é necessário saber de qual é que se vai obter a ordem parcial. Para isso vamos introduzir um novo conceito — o de *espaço de crenças plausíveis* — que vai corresponder ao conjunto de fórmulas LEP donde se extrai a ordem.

Definição 7 Espaço de crenças plausíveis
Dado um contexto-meta β_M, e o conjunto dos espaços de crenças gerados por β_M, BS, o *espaço de crenças plausíveis* definido por β_M, PBS_{β_M}, é a intersecção de todos os espaços de crenças de BS: $PBS_{\beta_M} = \bigcap_{bs \in BS} bs$

A definição de espaço de crenças plausíveis corresponde a considerar uma abordagem céptica acerca das crenças do agente, isto é, apenas consideramos as crenças que existem em todos os espaços de crenças, o que na LEP (e SWMC) corresponde às consequências sólidas e plausíveis de um contexto. As crenças que estamos a deitar fora são crenças para as quais temos razões para acreditar, mas também para não acreditar, pelo que tomamos uma atitude cautelosa e ignoramo-las. As alternativas eram: considerar a reunião de todos os espaços de crenças, o que não faz sentido, uma vez que teríamos um conjunto inconsistente de fórmulas; ou escolher um dos espaços de crenças, para o que não temos informação (a menos que consideremos um outro nível-meta, acima deste, que especifique preferências entre as fórmulas deste nível de forma a escolher um dos espaços de crenças).

Com esta definição podemos agora dar a definição de *ordem parcial induzida por um contexto-meta*:

Definição 8 Ordem parcial induzida por um contexto-meta
Dado um contexto-meta β_M, a *ordem parcial induzida* por β_M, $\leq_{\beta_M}$, é definida da seguinte forma, onde PBS_{β_M} é o espaço de crenças plausíveis definido por β_M:

- para cada fórmula $Preferred(A, B) \in PBS_{\beta_M}$, em que A e B representam hipóteses ou suposições no nível-base, temos $B \leq_{\beta_M} A$.

- para cada fórmula $MostPreferred(A) \in PBS_{\beta_M}$, em que A representa uma hipótese ou suposição do nível-base, temos que $B \leq_{\beta_M} A$, para cada hipótese ou suposição B do nível-base.

Uma vez que um contexto-meta induz apenas uma ordem de preferências, de modo a obtermos mais do que uma precisamos de vários contextos-meta, ou seja, vamos considerar, não apenas um contexto no nível-meta, mas sim vários, tantos quantas as ordens parciais que desejarmos definir. Cada contexto corresponde, na prática, a um conjunto independente de crenças do agente. A configuração das crenças de um agente que utilize a SWMC para representar o seu conhecimento de base e a LEP para reflectir sobre as suas preferências pode ser representada pela figura 1: temos um nível-base onde estão representadas as crenças relativas ao "mundo exterior" que se está a modelar; e temos no nível-meta vários conjuntos de crenças que modelam as preferências do agente sobre as suas crenças do nível-base, cada um desses conjuntos dando origem a uma ordem parcial entre as crenças, ordens essas definidas ao nível da TRC.

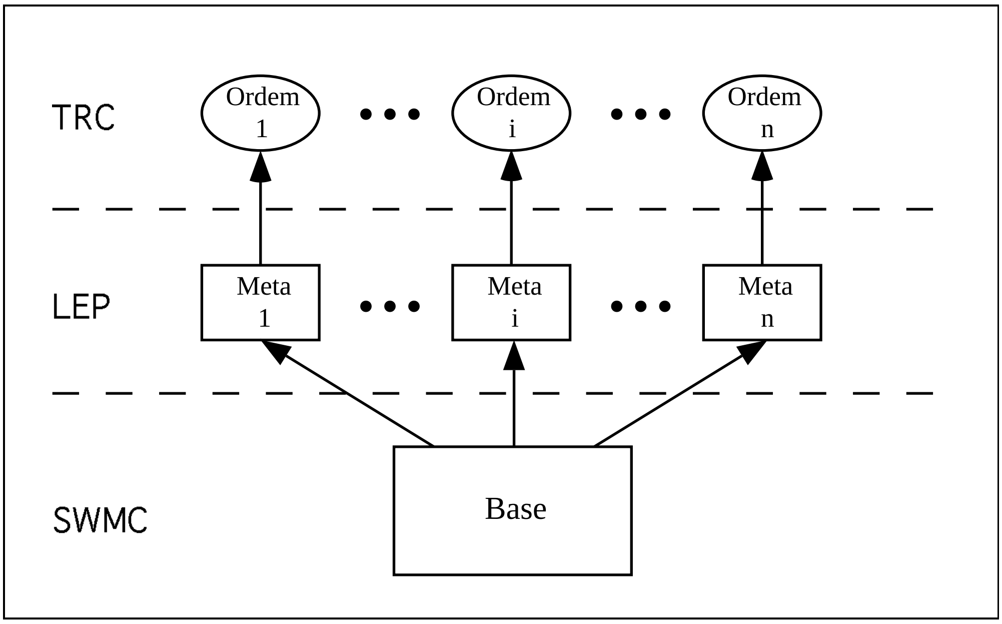

Figura 1: Configuração das crenças do agente

10 Integração entre a LEP e o SRC

Nesta secção vamos descrever como é que podemos integrar o conhecimento do nível-meta no SRC descrito na secção 4, bem como quando é que a inferência ao nível-meta deve ser desencadeada pelo SRC.

Para além disso, vamos discutir alguns dos problemas resultantes da implementação num sistema computacional do trabalho aqui apresentado.

10.1 Representação do conhecimento do nível-meta no SRC

A definição da linguagem para representar o conhecimento do nível-meta, a LEP, foi feita alterando a linguagem utilizada para representar o conhecimento do nível-base, a SWMC. Do ponto de vista sintáctico, não foram feitas alterações às regras de formação das fórmulas da linguagem; apenas foram introduzidas novas regras de inferência para lidar com um conjunto de predicados especiais.

Por este motivo, as estruturas utilizadas para representar o conhecimento do nível-meta são idênticas às utilizadas para a representação do conhecimento do nível-base. As alterações vão-se registar nos processos que manipulam essas estruturas:

- O motor de inferência vai ser aumentado com as novas regras de inferência, necessárias para manipular o conhecimento do nível-meta.

- O revisor de crenças vai ter que interpretar os espaços de crenças definidos pelos contextos do nível-meta, extraindo as ordens de preferência necessárias para as suas tarefas.

As regras de inferência adicionadas pela LEP apenas introduzem fórmulas atómicas que descrevem aspectos do nível-base. As regras de inferência para introduzir e/ou eliminar conectivas lógicas são as existentes em SWMC, pelo que os mecanismos de inferência utilizados no nível-meta são os mesmos que os utilizados no nível-base. Por este motivo, o motor de inferência do SRC vai poder ser utilizado também no nível-meta, necessitando apenas de um módulo adicional para efectuar as inferências relativas à ligação entre o nível-base e o nível-meta.

Apesar da uniformidade da representação do conhecimento nos dois níveis, eles são conceptualmente separados, quer no tipo de conhecimento que os constitui, quer nos objectivos da sua utilização. Por este motivo, o SRC vai ter duas bases de conhecimento, em vez de apenas uma: uma contém as fbfs do nível-base; a outra contém as fbfs do nível-meta. Na figura 2 está esquematizada a parte da estrutura do SRC que diz respeito às bases de conhecimento e ao motor de inferência.

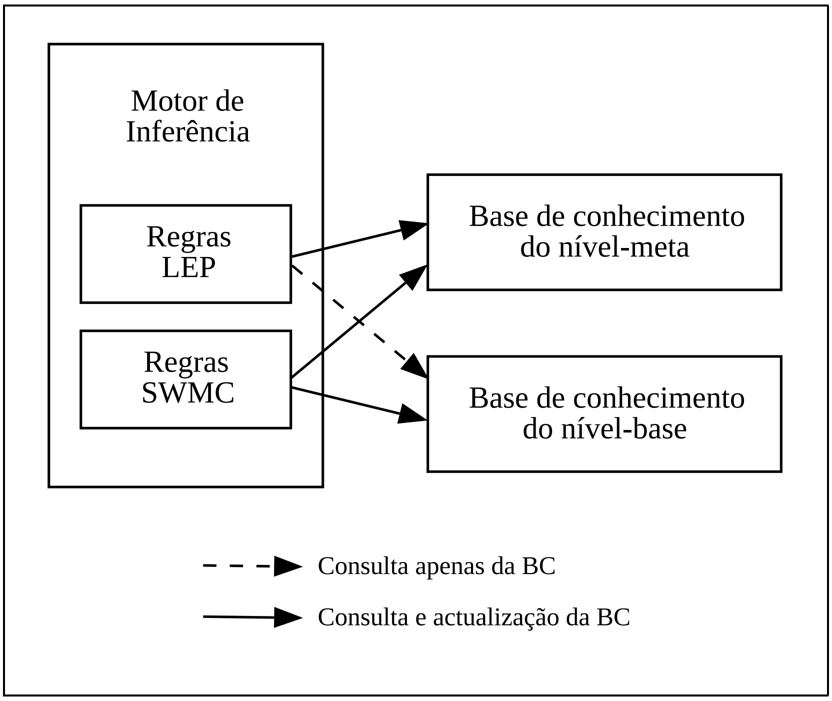

Figura 2: Estrutura parcial do SRC

Recorde-se que a arquitectura apresentada na secção 9 para um agente que utilize a SWMC e a LEP, pressupunha a existência de vários conjuntos de crenças ao nível-meta, cada um dos quais daria origem a uma ordem parcial de preferência. Esses vários conjuntos de crenças poderiam ser representados por várias bases de conhecimento no SRC, mas, uma vez que constituem conhecimento da mesma natureza, são representados todos numa única base de conhecimento. Para permitir a existência de vários conjuntos de crenças ao nível-meta, o SRC permite a existência simultânea de vários contextos no nível-meta, tal como já acontecia no nível-base.

10.2 Inferência ao nível-meta

O SRC utiliza as preferências entre as crenças de um agente em duas situações distintas: para resolver contradições reais e para escolher os espaços de crenças preferidos, de entre os que são definidos pelo contexto em consideração no nível-base.

Sempre que o SRC tiver que efectuar uma destas duas tarefas, vai ser desencadeada inferência no nível-meta, isto é, o SRC vai tentar provar ao nível-meta, usando a LEP, a existência de preferências entre as crenças do nível-base. A inferência no nível-meta tem que ser efectuada de cada vez que se quiserem saber as preferências entre as crenças do nível-base, não só porque o estado do nível-base pode ter sido alterado, mas também porque sempre que a inferência no nível-meta é desencadeada, apenas se tentam deduzir as preferências entre algumas das crenças do nível-base (como veremos mais à frente). Repare-se, no entanto, que pelo facto de estarmos a utilizar um sistema de revisão de crenças também ao nível-meta, preferências anteriormente deduzidas que se mantenham válidas dado o estado do nível-base são conhecidas do agente e não precisam de ser inferidas novamente. Da mesma forma, preferências anteriormente deduzidas que deixem de ser válidas, por alteração do estado do nível-base, deixam de ser acreditadas no nível-meta.

As deduções desencadeadas no nível-meta pelo SRC em cada um dos dois casos indicados acima vão ser descritas de seguida.

10.2.1 Resolução de contradições reais

Para resolver contradições reais, o revisor de crenças do SRC utiliza a TRC para determinar como deve ser resolvida a contradição. A TRC escolhe as hipóteses a retirar do contexto tendo em conta as relações de preferência existentes entre elas. Portanto, neste caso apenas interessa considerar as preferências entre as hipóteses na base da contradição.

Suponhamos que na base de conhecimento existem duas fbfs justificadas $<A, \tau_1, \alpha_1, \gamma_1>$ e $<\neg A, \tau_2, \alpha_2, \gamma_2>$, e que o contexto corrente do nível-base, β, é tal que $(\alpha_1 \cup \alpha_2) \subseteq \beta$. Neste caso, o revisor de crenças detecta uma contradição correspondente às fbfs A e $\neg A$, e para a resolver uma das hipóteses de $(\alpha_1 \cup \alpha_2)$ tem que ser removida do contexto β. A escolha da hipótese a retirar é tarefa da TRC, que utiliza as preferências existentes entre as crenças para efectuar essa escolha. Naturalmente, apenas as preferências entre as hipóteses de $(\alpha_1 \cup \alpha_2)$ são importantes para a TRC. Por este motivo, o SRC vai desencadear a inferência no nível-meta, tentando deduzir preferências entre as hipóteses de $(\alpha_1 \cup \alpha_2)$.

Assim, o SRC vai desencadear inferência para trás em cada um dos contextos do nível-meta, tentando provar $Preferred(A, B)$ e $Preferred(B, A)$ para cada par de hipóteses $A, B \in (\alpha_1 \cup \alpha_2)$. Depois disto, cada contexto do nível-meta define uma ordem parcial entre as hipóteses de $(\alpha_1 \cup \alpha_2)$. Estas ordens parciais, em conjunto com a ordem entre as ordens introduzida pelo utilizador, são usadas pelo SRC para calcular a ordem combinada, utilizada para escolher a hipótese de $(\alpha_1 \cup \alpha_2)$ a remover do contexto β.

10.2.2 Escolha dos espaços de crenças preferidos

Uma outra tarefa do SRC é escolher, de entre os vários espaços de crenças definidos por um contexto, qual ou quais são preferidos.

Cada espaço de crenças é definido univocamente por um dos núcleos definidos pelo contexto corrente do nível-base. A determinação dos núcleos preferidos é feita com base nas preferências existentes entre as suposições em conflito de cada par de núcleos. Neste caso, apenas as preferências entre estas suposições são relevantes para a tarefa do SRC, quer as definidas explicitamente pelo utilizador, quer as que são induzidas a partir da preferência entre as regras de omissão que lhes deram origem.

Suponhamos então que o SRC pretende saber qual de dois núcleos é preferido. Para isso começa por determinar o conjunto de suposições em conflito[28] nesses núcleos, $SupsConf$. Sabendo quais são as suposições relevantes para a escolha do núcleo preferido, o SRC vai desencadear inferência em cada um dos contextos do nível-meta, da seguinte forma: dadas duas suposições $A_1 = \mathit{Applicable}(D_1, c_1)$ e $A_2 = \mathit{Applicable}(D_2, c_2)$, pertencentes a núcleos diferentes, e tais que $A_1, A_2 \in SupsConf$, tenta provar $Preferred(A_1, A_2)$ e $Preferred(A_2, A_1)$; se não conseguir provar nenhuma das duas, então tenta provar também $Preferred(D_1, D_2)$ e $Preferred(D_2, D_1)$. Só vale a pena considerar as preferências entre as regras de omissão D_1 e D_2 se não existir uma preferência explícita entre as suposições. A preferência entre duas regras de omissão serve para induzir uma preferência entre duas suposições, mas quando existe uma preferência explícita entre essas suposições, isso constitui conhecimento mais específico, não se utilizando, neste caso, as preferências entre as regras de omissão (ver definição 6 na página 44).

10.3 Contradições no nível-meta

O conhecimento representado no nível-meta, sobre as preferências entre as crenças básicas de um agente, como discutimos na secção 5, constitui um componente importante do conhecimento de senso comum que um agente deve ter. Para além disso, pretendemos modelar um agente que raciocine num mundo em mudança, ou seja, as suas crenças básicas sofrem alterações à medida que o mundo sobre o qual ele está a raciocinar muda. Uma vez que o conhecimento do nível-meta é sobre as crenças do agente, e estas sofrem alterações, também o raciocínio efectuado ao nível-meta é sobre um "mundo" em mudança.

---

[28] Relembramos que, informalmente, as suposições em conflito de 2 núcleos são as que estão num e não estão no outro porque se estivessem o tornariam inconsistente.

O "problema" deste tipo de domínios, como já vimos, é o facto de o conhecimento ser incompleto e até mesmo contraditório. A LEP, por ser baseada na SWMC, uma lógica não-monótona, permite a utilização de regras de omissão, permitindo assim raciocinar sobre conhecimento incompleto. No entanto, devido à existência deste tipo de regras, podem existir vários espaços de crenças. Na LEP, para lidar com os vários espaços de crenças existentes seguimos uma abordagem céptica, isto é, apenas consideramos as crenças existentes em todos os espaços de crenças.[29]

Fica por resolver o problema da existência de conhecimento contraditório no nível-meta. O registo das dependências entre as fbfs no nível-meta permite ao SRC, tal como para o nível-base, determinar os culpados de uma contradição quando esta é detectada. O ponto mais delicado é sempre escolher qual deve ser a hipótese a retirar de forma a que a contradição seja resolvida. Esta é uma tarefa da TRC que, no caso de uma contradição no nível-base, utiliza as preferências entre as crenças do agente para escolher entre as várias alternativas.

No caso de uma contradição no nível-meta, temos também o problema de escolher qual a hipótese a retirar para a resolver, e podemos considerar que para o fazer utilizamos conhecimento sobre as crenças do nível-meta: as preferências existentes entre essas crenças. De acordo com esta linha de raciocínio, necessitamos de um novo nível-meta(-meta), representando esse conhecimento sobre as crenças do nível-meta anterior. Esta abordagem, apesar de interessante e elegante, precisa de ser cuidadosamente considerada, pois coloca alguns problemas conceptuais:

- Onde parar a criação de novos níveis-meta e, nesse caso, como resolver as contradições existentes no último nível.
- Qual a adequação epistemológica da existência de mais níveis-meta que representem conhecimento sobre o nível anterior: se é ou não possível e natural a representação desse conhecimento através de regras sobre as crenças do nível anterior.

A discussão destes problemas está fora do âmbito deste trabalho e constitui um tópico de trabalho futuro a desenvolver nesta área.

Na resolução de contradições ao nível-meta, a abordagem mais simples de seguir é considerar que a escolha da forma de resolver a contradição cabe ao utilizador, tal como acontecia no nível-base sem a existência de preferências.

[29] As consequências plausíveis de um contexto (definição 7 na página 47).

Repare-se, no entanto, que esta abordagem não constitui uma limitação muito grave do sistema, uma vez que é possível a existência de vários contextos simultâneos para o nível-meta. O que o utilizador precisa de garantir é que, em cada um desses contextos, isoladamente, o conhecimento não seja contraditório. Este requisito não é muito difícil de cumprir, uma vez que é natural que cada contexto represente conhecimento reduzido acerca de um determinado critério de preferência. Para além disso, o conhecimento existente no nível-meta diz respeito à especificação de preferências, e não nos parece muito natural a existência de conhecimento contraditório neste domínio — repare-se que, na LEP, $Preferred(A, B)$ não é contraditório com $Preferred(B, A)$.

10.4 Problemas computacionais

A implementação de qualquer lógica (pelo menos das de primeira ordem) num sistema computacional tem sempre o problema de não ser completa ([Pelletier 1991]). De facto, qualquer sistema computacional é necessariamente finito e tem limitações nos recursos que pode dispender para uma determinada tarefa.

O SRC, por ser a formalização de um sistema computacional, logo, finito e limitado, não pode representar os espaços de crenças da lógica SWMC. Em vez disso, mantém numa estrutura — a base de conhecimento — o conjunto (finito) das fbfs justificadas introduzidas no sistema até ao momento. Por este motivo, apesar de uma determinada fbf poder ser derivável, segundo a lógica, de um determinado contexto, uma implementação do SRC pode não conseguir apresentar uma derivação para essa fbf.

Os sistemas computacionais para os vários sistemas de prova existentes para a lógica clássica, apesar de serem incompletos, pretendem ser correctos, isto é, qualquer fórmula que o sistema consiga derivar é derivável também no sistema de prova. No caso dos sistemas computacionais baseados em lógicas não-monótonas, no entanto, isso não acontece: os sistemas computacionais deste tipo podem derivar fórmulas que não são deriváveis pela lógica, ou seja, não são correctos em relação à lógica.

Para poderem saltar para conclusões, as lógicas não-monótonas normalmente efectuam alguns testes de consistência, mas, uma vez que um sistema computacional não é completo, qualquer teste de consistência que ele faça tem que ser limitado, podendo saltar para conclusões que segundo a lógica não são admissíveis. Estas limitações computacionais, no entanto, não são muito graves, como se argumenta em [Cravo 1992, Cravo & Martins 1993].

A implementação da LEP no SRC, naturalmente, tem estes mesmos problemas: falta de completude e falta de correcção. A falta de completude é inevitável como já vimos. A falta de correcção deve-se não só ao facto de a LEP ser uma lógica não-monótona (como a SWMC), mas também à existência das regras de inferência de introdução dos predicados especiais negados. Consideremos, por exemplo, o caso da regra de "Introdução do ¬Sound": dizemos que podemos inferir $\neg Sound(A)$ a partir de um determinado contexto do nível-base β sempre que A não é uma consequência sólida de β. Para podermos aplicar esta regra de inferência, precisamos de verificar que A não é uma consequência sólida do contexto β. Como o sistema não é completo, podemos não conseguir encontrar uma derivação sólida para A, a partir do contexto β, mas ela existir segundo a lógica. Neste caso inferimos $\neg Sound(A)$, apesar de segundo a lógica devermos inferir $Sound(A)$.

Este problema manifesta-se também no caso de o SRC estar a utilizar o nível-meta para determinar as preferências entre as crenças do nível-base. Pode ser possível derivar, para duas hipóteses A e B, $Preferred(A, B)$ e $Preferred(B, A)$, mas o sistema apenas conseguir derivar uma delas, por exemplo, $Preferred(A, B)$. Neste caso, o SRC iria utilizar uma ordem parcial em que $B < A$, apesar de isso não estar correcto.

Esta limitação da implementação do sistema é uma consequência directa e inevitável do tipo de informação com que pretendemos raciocinar usando estes formalismos.

Repare-se, no entanto, que não inferimos uma determinada fórmula, $\neg Sound(A)$, por exemplo, sem antes termos feito alguns testes acerca dessa possibilidade. De facto, inferir $\neg Sound(A)$ quando conseguimos inferir no sistema $Sound(A)$ não é muito razoável, mas no caso de não conseguirmos inferir este último, julgamos que é lícito "saltar" para a conclusão $\neg Sound(A)$. O facto de mais tarde chegarmos à conclusão que não o deveríamos ter feito, visto isso ser contraditório com o resto das nossas crenças, leva-nos a que tenhamos que rever as nossas crenças face a essa nova informação. Isto é apenas mais uma consequência de o conhecimento de um agente ser incompleto, não só no que diz respeito ao "mundo" exterior, mas também em relação ao conhecimento sobre as suas próprias crenças.

10.4.1 Inferência no nível-base

Consideremos que o sistema utilizado para implementar o nível-base é ideal, isto é, permite a representação explícita dos espaços de crenças definidos segundo a lógica SWMC. Neste caso, a implementação das regras de inferência de ligação da LEP é trivial: basta fazer uma inspecção às estruturas do nível-base para verificar as condições de aplicabilidade impostas pelas regras. Por exemplo, na implementação da regra de "Introdução do Sound", para podermos inferir $Sound(A)$ no nível-meta basta-nos encontrar uma fbf suportada do nível-base $<A, \tau, \alpha>$ em que α esteja contido no contexto corrente do nível-base.[30]

O SRC, como vimos, não é um sistema ideal (no sentido acima): apesar de uma fbf ser uma consequência do contexto corrente do nível-base, pode não existir nenhuma fbf suportada na base de conhecimento do SRC para ela. Por este motivo, não basta, na implementação das regras de inferência de ligação da LEP, fazer consultas às estruturas utilizadas para representar a informação do nível-base. Quando, no nível-meta, precisamos de provar $Sound(A)$ para alguma fórmula A do nível-base, não basta verificar a existência de uma fbf suportada para A na base de conhecimento utilizada pelo SRC para representar o conhecimento do nível-base. Temos que tentar deduzir A no nível-base primeiro. Se A já tiver sido deduzida, então o SRC não precisa de fazer nada; senão, tenta deduzir A de forma a que essa fórmula apareça na base de conhecimento. Apenas após esta tentativa de deduzir A no nível-base se pode testar a condição de A ser uma consequência sólida do contexto corrente do nível-base.

No entanto, no caso de a inferência no nível-meta ter sido desencadeada pelo SRC para resolver uma contradição detectada no nível-base, o estado do nível-base é inconsistente. Nesta situação, de acordo com a definição de espaço de crenças dada pela SWMC, não existe nenhum espaço de crenças, não existindo, por isso, nenhumas consequências do contexto inconsistente. Obviamente que, nesta situação, a tentativa de, por exemplo, provar $Sound(A)$ no nível-meta, qualquer que seja o A, nunca será bem sucedida, não sendo necessário (nem desejável) tentar deduzir A no nível-base.

Suponhamos agora que a inferência no nível-meta tinha sido desencadeada pelo SRC para escolher os espaços de crenças preferidos, e que durante esse processo tinha sido despoletada a inferência no nível-base. Durante a inferência no nível-base pode ter sido detectada uma contradição, sendo necessário resolvê-la, ou essa inferência pode ter dado origem, mesmo sem a existência de uma contradição, a uma alteração nos espaços de crenças definidos pelo contexto corrente. Nestes casos, em que se verifica uma alteração nos espaços de crenças, o SRC terá que recomeçar a sua tarefa de escolher os espaços de crenças preferidos definidos pelo contexto do nível-base. Se nenhuma alteração destas se verificar no nível-base, então a inferência no nível-meta pode continuar assim que terminar a inferência no nível-base.[31]

---

[30] Assumindo que, mesmo tendo um conjunto infinito de crenças, a pesquisa nesse conjunto pode ser feita em tempo finito.

[31] A inferência no nível-meta não tem que ficar necessariamente "parada", pode ser continuada uma outra linha de raciocínio que não a que deu origem a esta inferência no nível-base.

11 Exemplos

Nesta secção vamos apresentar alguns exemplos de como é que a LEP poderia ser usada para representar o conhecimento que o agente tem acerca das suas crenças, quer para casos em que esse conhecimento é específico do domínio, quer em casos em que esse conhecimento pode ser usado na generalidade dos casos, independentemente do domínio.

11.1 Exemplo das reuniões

Recordemos o exemplo das reuniões, apresentado na página 18, tal como foi representado em SWMC (página 41):

Exemplo 4 (Exemplo das reuniões em SWMC)

$$fbf1 : \bigtriangledown(p,t)[(vaiReuniao(p) \wedge prefereReunioes(p,t)) \rightarrow ocorreReuniao(t)]$$
$$fbf2 : ocorreReuniao(Manha) \leftrightarrow \neg ocorreReuniao(Tarde)$$
$$fbf3 : general(Guerra)$$
$$fbf4 : capitao(Capucho)$$
$$fbf5 : prefereReunioes(Guerra, Manha)$$
$$fbf6 : prefereReunioes(Capucho, Tarde)$$
$$fbf7 : vaiReuniao(Guerra)$$
$$fbf8 : vaiReuniao(Capucho)$$

Tal como vimos anteriormente, a partir destas fbfs a SWMC iria gerar dois núcleos:

$$\Sigma_1 = \beta \cup \{Applicable(fbf1, Guerra, Manha)\}$$
$$\Sigma_2 = \beta \cup \{Applicable(fbf1, Capucho, Tarde)\}$$

sendo $\beta = \{fbf1, \ldots, fbf8\}$.

Para fazer com que o sistema prefira Σ_1 a Σ_2, é necessário estabelecer uma preferência entre as duas suposições, por exemplo com a fbf da LEP:

$$mfbf1 : Preferred(Applicable(fbf1, Guerra, Manha), Applicable(fbf1, Capucho, Tarde))$$

fazendo com que o agente prefira o espaço de crenças gerado a partir de Σ_1 ao gerado a partir de Σ_2.

Esta fórmula, no entanto, representa apenas um caso particular daquilo que nós sabemos acerca deste domínio. Não preferimos a aplicação da $fbf1$ apenas ao $Guerra$ em relação ao $Capucho$, mas sim de todos os Generais em relação a todos os Capitães. Para representar esse conhecimento na LEP, podemos substituir $mfbf1$ por:

$$\begin{aligned} mfbf2 : \bigtriangledown(g, c, t1, t2)[(Sound(general(g)) \land Sound(capitao(c))) \rightarrow \\ Preferred(Applicable(fbf1, g, t1), Applicable(fbf1, c, t2))] \end{aligned}$$

Se esta fórmula é mais geral que $mfbf1$, devemos conseguir derivar $mfbf1$ a partir de $mfbf2$. A derivação é a seguinte, considerando, para facilidade de escrita, as seguintes abreviaturas:

$$\begin{aligned} InBCtx &= InBaseContext \\ Pref &= Preferred \\ App &= Applicable \end{aligned}$$

1)	$<general(Guerra),\ hyp,\ \{general(Guerra)\}>$	Nível-base
2)	$\ll Sound(general(Guerra)),\ der,\ \{InBCtxt(general(Guerra))\}\gg$	ISound(1)
3)	$<capitao(Capucho),\ hyp,\ \{capitao(Capucho)\}>$	Nível-base
4)	$\ll Sound(capitao(Capucho)),\ der,$ $\{InBCtxt(capitao(Capucho))\}\gg$	ISound(4)
5)	$\ll Sound(general(Guerra)) \land Sound(capitao(Capucho)),\ der,$ $\{InBCtxt(general(Guerra)), InBCtxt(capitao(Capucho))\}\gg$	I$\land$(2, 4)
6)	$\ll mfbf2,\ hyp,\ \{mfbf2\}\gg$	Hip
7)	$\ll Sound(general(Guerra)) \land Sound(capitao(Capucho)) \rightarrow$ $Pref(App(fbf1, Guerra, Manha), App(fbf1, Capucho, Tarde)),$ $der,\ \{mfbf2, App(mfbf2, Guerra, Capucho, Manha, Tarde)\}\gg$	E$\bigtriangledown$(6)
8)	$\ll Pref(App(fbf1, Guerra, Manha), App(fbf1, Capucho, Tarde)),$ $der,\ \{mfbf2, App(mfbf2, Guerra, Capucho, Manha, Tarde),$ $InBCtxt(general(Guerra)), InBCtxt(capitao(Capucho))\}\gg$	E$\rightarrow$(5, 7)

Falta verificar se a fbf $Pref(App(fbf1, Guerra, Manha), App(fbf1, Capucho, Tarde))$ pertence ao espaço de crenças do nível-meta definido a partir de $mfbf2$ e do contexto de nível-base β. O contexto-meta, β_M, dado β e o conjunto de hipóteses do nível-meta $\{mfbf2\}$ é definido como sendo

$$\beta_M = \{mfbf2, InBaseContext(fbf1), \ldots, InBaseContext(fbf8), CurrentBaseContext(\beta)\}$$

Este contexto define apenas um núcleo

$$\Sigma_M = \beta_M \cup \{Applicable(mfbf2, Guerra, Capucho, Manha, Tarde\}$$

que define um espaço de crenças no qual está contida a fórmula pretendida:

$$Preferred(Applicable(fbf1, Guerra, Manha), Applicable(fbf1, Capucho, Tarde))$$

Logo, a ordem de preferência extraída deste espaço de crenças, $\leq$, contém a relação $Applicable(fbf1, Capucho, Tarde) \leq Applicable(fbf1, Guerra, Manha)$, permitindo ao sistema de revisão de crenças escolher o núcleo Σ_1, como era pretendido.

11.2 Extensão do exemplo das reuniões

Se pretendermos que o nosso domínio passe a incluir conhecimento sobre outras patentes da hierarquia militar, no nível-meta o conhecimento poderia ser representado através das seguintes fbfs:

$mfbf1 : superior(sargento, soldado)$
$mfbf2 : superior(tenente, sargento)$
$mfbf3 : superior(capitao, tenente)$
$mfbf4 : superior(general, capitao)$
$mfbf5 : \forall(x, y, z)[(superior(x, y) \wedge superior(y, z)) \rightarrow superior(x, z)]$
$mfbf6 : \triangledown(Pat1, Pat2, p1, p2, t1, t2)$
$[(superior(Pat1, Pat2) \wedge Sound(Pat1(p1)) \wedge Sound(Pat2(p2))) \rightarrow$
$Preferred(Applicable(fbf1, p1, t1), Applicable(fbf1, p2, t2))]$

Repare-se que, com estas fórmulas podemos inferir a preferência entre as suposições do exemplo anterior. Para além disso, se acrescentarmos militares de várias patentes ao nível-base, não necessitamos de alterar o conhecimento do nível-meta, continuando este a inferir as preferências desejadas.

Suponhamos agora que queremos representar, no nível-base, conhecimento acerca dos almoços dos militares. Isso corresponde a introduzir as seguintes hipóteses no nível-base:

$$fbf9 : \triangledown(p,t)[(vaiAlmoco(p) \wedge prefereAlmocar(p,t)) \rightarrow ocorreAlmoco(t)]$$
$$fbf10 : \forall(t1,t2)[t1 \neq t2 \rightarrow (ocorreAlmoco(t1) \leftrightarrow \neg ocorreAlmoco(t2))]$$
$$fbf11 : prefereAlmocar(Guerra, 12)$$
$$fbf12 : prefereAlmocar(Capucho, 14)$$
$$fbf13 : vaiAlmoco(Guerra)$$
$$fbf14 : vaiAlmoco(Capucho)$$

Para que possamos estender o nível-meta de forma a representar as preferências acerca desta informação, ou duplicamos a $mfbf6$ para o caso dos almoços, ou a substituimos por:

$$\begin{aligned} mfbf6' : & \triangledown(RO, Pat1, Pat2, p1, p2, t1, t2) \\ & [(superior(Pat1, Pat2) \wedge Sound(Pat1(p1)) \wedge Sound(Pat2(p2))) \rightarrow \\ & \quad Preferred(Applicable(RO, p1, t1), Applicable(RO, p2, t2))] \end{aligned}$$

Estamos, com isto, a quantificar sobre todas as regras de omissão com duas variáveis, o que, num caso mais complexo, não é adequado. Por exemplo, se estivermos a representar a informação de que “Tipicamente uma pessoa namora com quem gosta”, através da fbf $\triangledown(p1,p2)[gosta(p1,p2) \rightarrow namora(p1,p2)]$, não preferimos a aplicação desta regra a um general em detrimento de um capitão. Neste caso, uma forma alternativa de representar o conhecimento pretendido com a $mfbf6$, sem a duplicar, é substitui-la por:

$$\begin{aligned} mfbf6'' : & \triangledown(RO, Pat1, Pat2, p1, p2, t1, t2) \\ & [(regraGosto(RO) \wedge superior(Pat1, Pat2) \wedge \\ & \quad Sound(Pat1(p1)) \wedge Sound(Pat2(p2))) \rightarrow \\ & \quad Preferred(Applicable(RO, p1, t1), Applicable(RO, p2, t2))] \\ mfbf7 : & \; regraGosto(fbf1) \\ mfbf8 : & \; regraGosto(fbf9) \end{aligned}$$

A representação das preferências através de uma regra de omissão permite que existam excepções a essa regra. Consideremos o caso do capitão *Capucho*, que tem um problema de estômago e que portanto tem de almoçar às 14 horas. Para representar essa preferência no nível-meta, basta introduzir a seguinte hipótese:

$$\forall(p,t)[\ Preferred(Applicable(fbf9, Capucho, 14), Applicable(fbf9, p, t)) \wedge \\ \neg Preferred(Applicable(fbf9, p, t), Applicable(fbf9, Capucho, 14))]$$

o que invalidaria a aplicação da regra de omissão $mfbf6''$ no caso do capitão *Capucho*.

11.3 Regra de especificidade

A regra de especificidade definida pelo SRC (ver página 15) já está implementada no sistema, e em princípio constitui meta-conhecimento que é independente do domínio.

Uma situação em que essa regra é útil é quando se pretende representar conhecimento acerca de uma hierarquia, onde as regras mais específicas para um determinado objecto do "mundo real" devem ter prioridade sobre as regras mais gerais. Por exemplo, suponhamos que estava representada a seguinte informação no nível-base:

$$\forall(x) pinguim(x) \rightarrow ave(x)$$
$$\bigtriangledown(x) ave(x) \rightarrow voa(x)$$
$$\bigtriangledown(x) pinguim(x) \rightarrow \neg voa(x)$$
$$pinguim(Tweety)$$

Intuitivamente, preferimos acreditar que o *Tweety* não voa a acreditar que ele voa. É exactamente isso que acontece se usarmos a regra de especificidade implementada no SRC. Se esta regra não estivesse implementada no SRC, poderíamos representá-la explicitamente usando uma fórmula da LEP da seguinte forma:

$$\forall(\Phi_1, \Phi_2, x) preferred(\bigtriangledown(x) pinguim(x) \rightarrow \Phi_1\{x\}, \bigtriangledown(x) ave(x) \rightarrow \Phi_2\{x\})$$

A regra de especificidade na sua forma mais geral poderia ser representada na LEP por:

$$\forall(\Psi_1, \Psi_2, \Phi_1, \Phi_2, x)[(sound(\forall(x)\Psi_1\{x\} \rightarrow \Psi_2\{x\}) \wedge \neg sound(\forall(x)\Psi_2\{x\} \rightarrow \Psi_1\{x\})) \rightarrow \\ preferred(\bigtriangledown(x)\Psi_1\{x\} \rightarrow \Phi_1\{x\}, \bigtriangledown(x)\Psi_2\{x\} \rightarrow \Phi_2\{x\})]$$

A Regras de inferência da lógica SWMC

Regras clássicas

Hipótese (Hip)

Dada qualquer fbf $A \in (\mathcal{L}_{FOL} \cup \mathcal{L}_D \cup \mathcal{L}_E)$, podemos escrever a fbf suportada $<A,\ hyp,\ \{A\}>$.

Introdução da implicação (I$\rightarrow$)

A partir de $<B,\ \tau,\ \alpha>$, $B \in \mathcal{L}_{FOL}$ e qualquer hipótese $H \in (\alpha \cap \mathcal{L}_{FOL})$ podemos inferir $<H \rightarrow B,\ der,\ \alpha - \{H\}>$ desde que, para toda a suposição $Applicable(D,\ c) \in (\alpha\ - \{H\})$, se tenha $(\alpha\ - \{H\}) \vdash_{SWMC} Applicable(D,\ c)$. Se tal não acontecer, então podemos inferir $<H \rightarrow B,\ der,\ \alpha>$.

Modus Ponens – Eliminação da implicação (MP)

A partir de $<A,\ \tau_1,\ \alpha_1>$ e $<A \rightarrow B,\ \tau_2,\ \alpha_2>$ podemos inferir $<B,\ der,\ \alpha_1 \cup \alpha_2>$.

Modus Tollens – Introdução da implicação (MT)

A partir de $<\neg B,\ \tau_1,\ \alpha_1>$ e $<A \rightarrow B,\ \tau_2,\ \alpha_2>$ podemos inferir $<\neg A,\ der,\ \alpha_1 \cup \alpha_2>$.

Introdução da dupla negação (IDN)

A partir de $<A,\ \tau,\ \alpha>$, se $A \in \mathcal{L}_{FOL}$, podemos inferir $<\neg\neg A,\ der,\ \alpha>$.

Eliminação da dupla negação (EDN)

A partir de $<\neg\neg A,\ \tau,\ \alpha>$ podemos inferir $<A,\ der,\ \alpha>$.

Introdução da negação (I$\neg$)

A partir de $<A \wedge \neg A,\ \tau,\ \alpha>$, $\alpha \subset \mathcal{L}_{FOL}$ e qualquer hipótese $H \in \alpha$ podemos inferir $<\neg H,\ der,\ \alpha - \{H\}>$.

Introdução da conjunção (I$\wedge$)

A partir de $<A,\ \tau_1,\ \alpha_1>$, $<B,\ \tau_2,\ \alpha_2>$, $A \in \mathcal{L}_{FOL}$ e $B \in \mathcal{L}_{FOL}$ podemos inferir $<A \wedge B,\ der,\ \alpha_1 \cup \alpha_2>$.

Eliminação da conjunção (E$\wedge$)

A partir de $<A \wedge B,\ \tau,\ \alpha>$ podemos inferir $<A,\ der,\ \alpha>$ ou $<B,\ der,\ \alpha>$, ou ambas.

Introdução da disjunção (I$\vee$)

A partir de $<A,\ \tau,\ \alpha>$, $A \in \mathcal{L}_{FOL}$ e qualquer $B \in \mathcal{L}_{FOL}$ podemos inferir $<A \vee B,\ der,\ \alpha>$, ou $<B \vee A,\ der,\ \alpha>$, ou ambas.

Eliminação da disjunção (E$\vee$)

A partir de $<A \vee B,\ \tau_1,\ \alpha_1>$, $<A \to C,\ \tau_2,\ \alpha_2>$ e $<B \to C,\ \tau_3,\ \alpha_3>$ podemos inferir $<C,\ der,\ \alpha_1 \cup \alpha_2 \cup \alpha_3>$.

Introdução do quantificador universal (I$\forall$)

A partir de $<A(c),\ der,\ \alpha>$, $A(c) \in \mathcal{L}_{FOL}$, e $A(c)$ usa um símbolo individual c que não ocorre em nenhuma fbf em α podemos inferir $<\forall(x)A(x),\ der,\ \alpha>$.

Eliminação do quantificador universal (E$\forall$)

A partir de $<\forall(x)A(x),\ \tau,\ \alpha>$ e qualquer símbolo individual c podemos inferir $<A(c),\ der,\ \alpha>$.

Introdução do quantificador existencial (I$\exists$)

A partir de $<A(c),\ \tau,\ \alpha>$, em que c é um símbolo individual, e $A(c) \in \mathcal{L}_{FOL}$ podemos inferir $<\exists(x)A(x),\ der,\ \alpha>$.

Eliminação do quantificador existencial (E$\exists$)

A partir de $<\exists(x)A(x), \tau, \alpha_1>$, $<B, der, \{A(c)\} \cup \alpha_2>$, e $A(c)$ usa um símbolo individual c que não ocorre em B nem em nenhuma fbf em α_2 podemos inferir $<B,\ der,\ \alpha_1 \cup \alpha_2>$.

Regras estendidas

Suposição e eliminação do quantificador de omissão (Sup-E$\bigtriangledown$)

A partir de $<\bigtriangledown(x)A(x),\ hyp,\ \{\bigtriangledown(x)A(x)\}>$, podemos inferir, para qualquer símbolo individual c, $<Applicable(\bigtriangledown(x)A(x), c),\ asp,\ \alpha>$, $<A(c),\ der,\ \alpha>$, em que $\alpha = \{\bigtriangledown(x)A(x), Applicable(\bigtriangledown(x)A(x), c)\}$.

B Regras de inferência da LEP

Regras de ligação

Introdução de Sound

Dada uma fbf suportada do nível-base $<A,\ \tau,\ \alpha>$, em que α não contém suposições, podemos inferir $\ll Sound(A),\ der,\ \{InBaseContext(H) : H \in \alpha\}\gg$.

Introdução de Plausible

Dado um contexto β do nível-base, e uma fbf $A \in \mathcal{L}$, tal que $\beta \vdash_P A$, podemos inferir $\ll Plausible(A),\ der,\ \{CurrentBaseContext(\beta)\}\gg$.

Introdução de Conceivable

Dado um contexto β do nível-base, e uma fbf $A \in \mathcal{L}$, tal que $\beta \vdash_C A$, podemos inferir $\ll Conceivable(A),\ der,\ \{CurrentBaseContext(\beta)\}\gg$.

Introdução de Derivable

Dada uma fbf suportada do nível-base $<A,\ \tau,\ \alpha>$, podemos inferir $\ll Derivable(A,\gamma)$, $der,\ \{\}\gg$, para qualquer $\gamma \subset \mathcal{L}$ tal que $\alpha \subseteq \gamma$.

Introdução de ProofTree

Dada uma árvore de prova em SWMC, $\mathcal{P}$, para uma fbf suportada $\mathcal{A} = <A,\ \tau,\ \alpha>$ podemos inferir $\ll ProofTree(A,\mathcal{P}),\ der,\ \{\}\gg$

Regras sobre as ordens de preferência

Transitividade do Preferred

De $\ll Preferred(A,B),\ \tau_1,\ \alpha_1\gg$ e de $\ll Preferred(B,C),\ \tau_2,\ \alpha_2\gg$ podemos inferir $\ll Preferred(A,C),\ der,\ \alpha_1 \cup \alpha_2\gg$.

Reflexividade do Preferred

Dada uma fbf do nível-base $A \in \mathcal{L}$, podemos inferir $\ll Preferred(A,A),\ der,\ \{\}\gg$.

Introdução do MostPreferred

De $\ll MostPreferred(A),\ \tau_1,\ \alpha_1\gg$ e $\ll Preferred(B,A),\ \tau_2,\ \alpha_2\gg$, podemos inferir $\ll MostPreferred(B),\ der,\ \alpha_1 \cup \alpha_2\gg$.

Regras sobre a negação dos predicados especiais

Introdução de ¬InBaseContext

Dado um contexto β e uma hipótese H do nível-base, tal que $H \notin \beta$, podemos inferir $\ll \neg InBaseContext(H)$, der, $\{CurrentBaseContext(\beta)\}\gg$.

Introdução de ¬CurrentBaseContext

Dado um contexto β do nível-base, podemos inferir, para qualquer contexto $\beta' \neq \beta$, $\ll \neg CurrentBaseContext(\beta')$, der, $\{CurrentBaseContext(\beta)\}\gg$.

Introdução de ¬Sound

Dado um contexto β do nível-base, e uma fbf $A \in \mathcal{L}$, tal que $\beta \not\vdash A$, podemos inferir $\ll \neg Sound(A)$, der, $\{CurrentBaseContext(\beta)\}\gg$.

Introdução de ¬Plausible

Dado um contexto β do nível-base, e uma fbf $A \in \mathcal{L}$, tal que $\beta \not\vdash_P A$, podemos inferir $\ll \neg Plausible(A)$, der, $\{CurrentBaseContext(\beta)\}\gg$.

Introdução de ¬Conceivable

Dado um contexto β do nível-base, e uma fbf $A \in \mathcal{L}$, tal que $\beta \not\vdash_C A$, podemos inferir $\ll \neg Conceivable(A)$, der, $\{CurrentBaseContext(\beta)\}\gg$.

Introdução de ¬Derivable

Dada uma fbf $A \in \mathcal{L}$ e $\alpha \subset \mathcal{L}$, tal que não existe nenhuma fbf suportada $<A$, τ, $\alpha'>$, em que $\alpha' \subseteq \alpha$, podemos inferir $\ll \neg Derivable(A, \alpha)$, der, $\{\}\gg$.

Introdução de ¬ProofTree

Dada uma fbf $A \in \mathcal{L}$, podemos inferir, para qualquer árvore $\mathcal{P}$ que não constitua uma árvore de prova em SWMC para uma fbf suportada $\mathcal{A}$ qualquer com $wff(\mathcal{A}) = A$, $\ll \neg ProofTree(A, \mathcal{P})$, der, $\{\}\gg$.

C Axiomas próprios das relações usadas

Os axiomas próprios da relação $varsLivres$, de acordo com a representação dada no nível-meta às fórmulas do nível-base descrita na secção 6, são os seguintes:

$$\forall(a)[varsLivres(ct(a), \{\})]$$
$$\forall(x)[varsLivres(var(x), \{x\})]$$
$$\forall(f, args, vars)[varsLivresSeq(args, vars) \to varsLivres(function(f, args), vars)]$$
$$\forall(P, args, vars)[varsLivresSeq(args, vars) \to varsLivres(pred(P, args), vars)]$$
$$\forall(A, vars)[varsLivres(A, vars) \to varsLivres(not(A), vars)]$$
$$\forall(A, B, vars_A, vars_B)[(varsLivres(A, vars_A) \wedge varsLivres(B, vars_B)) \to varsLivres(and(A, B), vars_A \cup vars_B)]$$
$$\forall(A, B, vars_A, vars_B)[(varsLivres(A, vars_A) \wedge varsLivres(B, vars_B)) \to varsLivres(or(A, B), vars_A \cup vars_B)]$$
$$\forall(A, B, vars_A, vars_B)[(varsLivres(A, vars_A) \wedge varsLivres(B, vars_B)) \to varsLivres(imply(A, B), vars_A \cup vars_B)]$$
$$\forall(A, vars_A, x)[varsLivres(A, vars_A) \to varsLivres(forall(x, A), vars_A - x)]$$
$$\forall(A, vars_A, x)[varsLivres(A, vars_A) \to varsLivres(exists(x, A), vars_A - x)]$$
$$\forall(A, vars_A, x)[varsLivres(A, vars_A) \to varsLivres(default(x, A), vars_A - x)]$$
$$\forall(D, c)[varsLivres(applicable(D, c), \{\})]$$

sendo $\cup$ e $-$ as operações usuais sobre conjuntos: reunião e diferença, respectivamente.

C.1 Variáveis livres de uma sequência

Definimos o predicado $varsLivresSeq(seq, vars)$, que relacciona uma sequência de termos com o conjunto das suas variáveis livres da seguinte forma:

$$varsLivresSeq(NIL, \{\})$$
$$\forall(x, seq, vars_x, vars_{seq})[(varsLivres(x, vars_x) \wedge varsLivresSeq(seq, vars_{seq})) \to varsLivresSeq(cons(x, seq), vars_x \cup vars_{seq}]$$

Onde a constante NIL representa a sequência vazia e $cons(x, seq)$ representa o termo x à cabeça da sequência seq.

Referências

Cravo, M. R. [1992], Raciocínio por Omissão e Revisão de Crenças: Dois Aspectos do Senso Comum, PhD thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa, Lisboa, Portugal.

Cravo, M. R. [1993*a*], A belief revision theory based on SWMC, Technical Report GIA 93/03, Instituto Superior Técnico, Universidade Técnica de Lisboa, Lisboa, Portugal.

Cravo, M. R. [1993*b*], SWMC: A logic for default reasoning and belief revision (a new version), Technical Report GIA 93/02, Instituto Superior Técnico, Universidade Técnica de Lisboa, Lisboa, Portugal.

Cravo, M. R. [1994]. Comunicação pessoal.

Cravo, M. R. & Martins, J. P. [1993], 'SNePSwD, a newcomer to the SNePS family', *Journal of Experimental and Theoretical Artificial Intelligence* **5**, 135–148.

de Kleer, J. [1986], 'An assumption-based truth maintenance system', *Artificial Intelligence* **28**(2), 127–162.

Doyle, J. [1979], 'A truth maintenance system', *Artificial Intelligence* **12**(3), 231–272.

Doyle, J. [1991], Rational belief revision, *in* Allen, Fikes & Sandewall, eds, 'Proc. Second International Conference on Principles of Knowledge Representation and Reasoning', Morgan Kaufmann Inc., San Mateo, CA, pp. 163–174.

Gärdenfors, P. [1988], *Knowledge in flux: modeling the dynamics of epistemic states*, The MIT Press, Cambridge, MA.

Genesereth, M. [1988], Introspective fidelity, *in* P. Maes & D. Nardi, eds, 'Meta-Level Architectures and Reflection', Elsevier Science Publishers B.V., (North-Holland), pp. 75–86.

Genesereth, M. R. & Nilsson, N. J. [1987], *Logical Foundations of Artificial Intelligence*, Morgan Kaufmann, Los Altos, CA.

Levi, I. [1977], 'Subjunctives, dispositions, and chances', *Synthese* **34**, 423–455.

Martins, J. P. [1983], Reasoning in multiple belief spaces, PhD thesis, State University of New York at Buffalo, Buffalo, N.Y.

Nebel, B. [1989], A knowledge level analysis of belief revision, *in* Brachman, Levesque & Reiter, eds, 'Proc. First International Conference on Principles of Knowledge Representation and Reasoning', Morgan Kaufmann Inc., San Mateo, CA, pp. 301–311.

Pelletier, F. J. [1991], The philosophy of automated theorem proving, *in* 'Proc. of the 12 th IJCAI', Sidney, Australia, pp. 1039–1045.

Shapiro, S. C. [1979], The SNePS Semantic Network Processing System, *in* Findler, ed., 'Associative Networks: Representation and Use of Knowledge by Computers', Academic Press, New York, N.Y., pp. 179–203.

Shapiro, S. C. & Martins, J. P. [1990], Recent advances and developments: The SNePS 2.1 report, *in* Kumar, ed., 'Current Trends in SNePS - Semantic Network Processing System: Proc. of the First Annual Workshop', number 437 *in* 'Lecture Notes in Artificial Intelligence', Springer-Verlag, Heidelberg, Germany, pp. 1–13.

Shapiro, S. C. & Rapaport, W. J. [1993], The SNePS family, *in* F. Lehman, ed., 'Semantic Networks in Artificial Intelligence', Pergamon, pp. 243–275.

van Harmelen, F., Simpson, A., Giunchiglia, F., Serafini, L. & Smaill, A. [1994], A discussion about naming relations, Technical Report 9007-16, I.R.S.T, Trento, Italy.