**Evolução da Base de Conhecimento**

A evolução do conhecimento apresentou-se como o maior fator de discussão no seio do grupo na realização do trabalho. Para evoluir a base de conhecimento tínhamos várias possibilidades, desde uma mais simples até a mais complexa de todas. De seguida iremos abordar sobre cada uma das abordagens disponíveis.

* Sobre a adição de conhecimento negativo quando se tem conhecimento positivo contraditório

Um dos objetivos no controlo da evolução da base conhecimento é garantir que esta não fique com conhecimento que a torne inconsistente. Desta forma não deve ser possível, manter na mesma base de conhecimento, um conhecimento positivo e outro negativo que são opostos. Por exemplo: Não podemos permitir que existam, em simultâneo, o seguinte conhecimento: prestador( 1,jose,'ortopedia',['Hospital de Braga'] ) e **-**prestador( 1,jose,'ortopedia',['Hospital de Braga'] ). Para controlar esta situação discutimos duas abordagens possíveis.

A primeira abordagem não permite que tal conhecimento contraditório seja adicionado. Ou seja, caso se queira adicionar conhecimento negativo quando já se possui conhecimento positivo que contradiz o que se quer adicionar, então tal conhecimento não deve ser possível acrescentar, continuando a base, com o mesmo conhecimento.

A segunda abordagem passa por realizar uma evolução da base de conhecimento. Isto é, quando se pretende adicionar conhecimento negativo quando a base já possui conhecimento positivo a contradizer o que se pretende adicionar, então o conhecimento que se esta a adicionar é posto em evidência com a alteração do conhecimento presente na base de conhecimento do tipo positivo para o tipo Desconhecido. Caso se voltasse a colocar a mesma informação negativa então a base de conhecimento já deixaria adicionar tal conhecimento removendo o que possui do tipo Desconhecido.

As duas abordagens são corretas e cada uma tem os seus prós e contras. A segunda abordagem permitiria uma evolução da base de conhecimento, porém é muito mais complexa para implementar, utilizando uma ferramenta como o PROLOG, e ainda era necessário acrescentar um novo tipo de conhecimento Desconhecido (por factos que serão abordados de seguida). No mundo real alguém conhecimento que hoje é verdadeiro, amanhã pode não o ser. No entanto é necessário ter uma relação entre o que é verdade e o que não é, como por exemplo pesos para o que conhecemos e o que é o novo conhecimento, sem nunca descartar o que de novo há para aprender nem esquecer o que antes se conhecia. É necessário “misturar tudo” o que se sabe, e uma forma de o fazer é com análises estatísticas. Isso não é implementável em PROLOG ou é bastante complexo.

A nossa escolha para tratar desta situação foi a primeira abordagem. Para a implementar apenas tivemos de verificar se a base de conhecimento já possuía um conhecimento positivo que fizesse *match* com o conhecimento negativo que se pretendesse adicionar. Decidimos implementar por ser mais simples que a segunda, mas não só. Analisando o conhecimento que queremos representar, atos médicos, é um “conhecimento fixo”, ou seja, que em princípio quando a possuímos conhecimento positivo ou negativo, este não se vai alterar nunca mais, já que é um historial médico, ou então a informação de um utente. Assim sendo, decidimos optar pela primeira abordagem que nos pareceu ser a mais indicada para a evolução sobre o conhecimento que pretendemos representar.

De notar que poderia ainda se ter discutido sobre uma terceira abordagem em que quando se inserir conhecimento negativo, se retirava o conhecimento positivo contraditório. Mas esta abordagem não teve qualquer apoio por qualquer elemento do grupo, pelo que não foi motivo de discussão.

A inserção de conhecimento positivo quando já existe conhecimento negativo segue um raciocínio análogo.

* Sobre a adição de conhecimento quando existe um conhecimento do tipo Desconhecido:

Uma possível seria “à la Funcionário Público” onde qualquer conhecimento que tentássemos inserir, tal não seria possível caso a base de conhecimento já possuísse conhecimento relativo ao facto que queremos inserir. Por exemplo, estamos a tentar inserir o seguinte conhecimento positivo: “cuidado( data( 2018,04,18,10 ),1,1,’rotina’,20,’Hospital’ )” mas já existe, na base de conhecimento, uma exceção para este mesmo cuidado onde não se sabe se o custo do mesmo foi de 20 ou 30. Com a abordagem referida a cima, como então já existe um conhecimento “análogo” não deve ser permitido acrescentar o conhecimento positivo. Desta forma seria bastante fácil tratar da evolução da base de conhecimento, já que apenas evoluía para conhecimento que ainda não possuía. Apesar da sua simplicidade não retrata de forma ideal o conhecimento que queremos representar, já que, recuando ao exemplo dado em cima, a qualquer momento podia-se ter certeza de qual o valor do custo do cuidado, pelo que poderá ser vantajoso inserir o novo conhecimento.

Outra abordagem, que seria muito mais complexa, era analisar não só se podíamos inserir ou não tal conhecimento como também fazer uma análise do conhecimento na base, onde o conhecimento verdadeiro estaria sempre no intervalo do conhecimento do tipo Desconhecido, e tudo o que não fosse verdadeiro ou desconhecido, seria totalmente falso.

Quando temos conhecimento do tipo Incerto, caso acrescentássemos conhecimento negativo nenhuma alteração ia ser efetuada, porém caso acrescentássemos conhecimento positivo, então todo o conhecimento do tipo desconhecido Incerto e, eventual, conhecimento negativo seriam apagados da base de conhecimento.

No caso de possuirmos conhecimento do tipo Impreciso, na adição de conhecimento positivo, que estivesse dentro do intervalo delimitado (caso estivesse fora do intervalo que o Desconhecido abrange, não seria permitida a sua adição) a ação seria análoga ao do caso referido anteriormente. O caso especial seria se pretendêssemos adicionar conhecimento negativo. Mais uma vez, caso este conhecimento estivesse fora do intervalo abrangido pelo conhecimento Desconhecido, tal conhecimento não seria possível ser adicionado. Porém, se estivesse dentro do intervalo, então deixaria acrescentar. Mas, utilizando esta abordagem, caso houvesse apenas um e um único facto do tipo desconhecido (após a inserção de um conhecimento do tipo negativo) então esse facto desconhecido deveria ser tornado verdadeiro e todos os restantes conhecimentos negativos apagados da base de conhecimento já que seria conhecimento redundante(). Pegando mais uma vez no exemplo referido a cima em que na base de conhecimento temos: “exceção( cuidado( data( 2018,04,18,10 ),1,1,’rotina’,20,’Hospital’ ) )” e “exceção( cuidado( data( 2018,04,18,10 ),1,1,’rotina’,30,’Hospital’ ) )” e queremos inserir “-cuidado( data( 2018,04,18,10 ),1,1,’rotina’,20,’Hospital’ )”. Neste caso, ficaríamos apenas com uma exceção, pelo que o único conhecimento que deveria de estar presente na base de conhecimento era “cuidado( data( 2018,04,18,10 ),1,1,’rotina’,20,’Hospital’ )”. Isto, deve-se ao facto de se considerar que o conhecimento verdadeiro está dentro dos limites do conhecimento do tipo Desconhecido e tudo o resto ser falso.

Esta abordagem seria bastante diferente da inicial, e muito difícil de conseguir implementar usando a ferramenta pedida PROLOG, tornando-se um desafio bastante interessante.

Por fim, uma terceira abordagem, onde apesar de poder existir um conhecimento do tipo Desconhecido, não há certezas absolutas de que o conhecimento verdadeiro esteja dentro do intervalo do Desconhecido e tudo o que está fora é falso, mas não é uma negação forte, mas sim antes, uma negação por falha na prova de que o conhecimento é verdadeiro ou desconhecido.

Nesta abordagem, a inserção tanto de conhecimento verdadeiro como de conhecimento falso ia ser feito de forma semelhante. Ou seja, caso inseríssemos estes tipos de conhecimento, este iria coexistir com o conhecimento do tipo Desconhecido, quer estivesse dentro ou fora dos limites, já que o facto de ser Desconhecido não garante com certeza absoluta de que o conhecimento verdadeiro esteja definido no intervalo do Desconhecido. É uma abordagem menos intuitiva que as outras, e até nos leva a pensar, se não temos a certeza de que o valor verdadeiro esteja dentro do Desconhecido então para que queremos deste tipo de conhecimento Desconhecido? Ora, com certeza uma boa questão, que pode ser respondida pelo seguinte exemplo: Dois colegas estão a discutir a classificação de uma equipa de futebol, neste caso o Vitória SC, na temporada transata, onde chegam à conclusão de que como o Vitória é uma equipa mediana em Portugal (no que toca ao valor financeiro e não ao da equipa) dizem que muito provavelmente ficou entre 5º e 8º, já que os 4 primeiros lugares estariam dedicados ao Benfica, Porto, Sporting e Braga. Porém, a verdadeira classificação não se encontra no intervalo que os amigos determinaram, já que o Vitória ficou em 4º lugar. Tínhamos um conhecimento do tipo Desconhecido onde parecia que quase de certeza a informação verdadeira estava dentro do intervalo, mas na verdade estava fora, porém, aquele conhecimento do tipo Desconhecido parecia fazer sentido.

Quando inserimos um conhecimento negativo, tanto este como o conhecimento Desconhecido, seriam mantidos na base de conhecimento, porém nunca ficaria inconsistente. Apesar de poder dar duas respostas para a mesma pergunta (caso se pergunte diretamente à base de conhecimento), para nós a base só fica inconsistente caso tenha conhecimento positivo e ao mesmo tempo negativo a contradizer o positivo. Mas porquê manter os dois? Porque caso eliminássemos o conhecimento do tipo Negativo, este ficaria a ser Desconhecido e não falso por negação na prova. O mesmo raciocínio é feito para a inserção de conhecimento verdadeiro.

Pelo facto de esta abordagem ser menos complexa ao nível de implementação usando a ferramenta PROLOG e ser bastante completa na representação do conhecimento, que no fundo é o mais importante a considerar, decidimos usá-la como estratégia na evolução de conhecimento no caso de já existir conhecimento do tipo Desconhecido.

* Sobre a adição de conhecimento desconhecido

No que toca à adição de conhecimento do tipo Desconhecido, optámos por não adicionar, devido ao seu grau de complexidade em PROLOG, mas no entanto, deixámos qual deveria de ser um tipo de abordagem a utilizar.

Caso o conhecimento Desconhecido fosse do tipo Interdito, seria necessário eliminar da base de conhecimento todo o conhecimento positivo, negativo e Desconhecido, já que, para qualquer pergunta para o predicado a inserir, a resposta a dar teria de ser sempre “desconhecido”.

Caso pretendêssemos adicionar conhecimento Desconhecido, mas no entanto, já existisse um conhecimento positivo ou negativo, que fizesse match com o predicado a inserir, íamos permitir que tal acontecesse, já que, segundo abordagens anteriores, deixamos que exista em simultâneo conhecimento Desconhecido e negativo ou Desconhecido e positivo.

Porém quando existisse conhecimento Desconhecido do tipo Impreciso ou Incerto (no caso de ser Interdito, como é obvio, não deixamos adicionar) era necessário efetuar uma análise ao conhecimento que já existe o que se pretende inserir para que o intervalo de desconhecido fosse sempre coerente. Ou seja, um exemplo seria, efetuar uma disjunção, onde se ficava com o maior dos dois intervalos, ou então, uma conjunção disjunta (este último poderia seria o mais ideal a aplicar, mas também seria o mais difícil para efetuar).

Ora para que tal fosse possível, era necessário fazer um parse do que se estava a inserir (para poder identificar qual o tipo de Desconhecido), posteriormente analisar a base de conhecimento, e por ventura, era necessário tomar decisões de qual o intervalo a manter sobre o desconhecido e ainda criar um novo predicado (diferente do que queríamos inserir), caso tivéssemos de alterar a variação dos valores do desconhecido.

Para auxiliar no processo de adição de conhecimento construímos um meta-predicado denominado de “evolução”. Este tinha como principal objetivo fazer uma avaliação (testando invariantes) de que se o conhecimento poderia ser adicionado ou não. Caso o conhecimento fosse adicionado com sucesso o resultado seria \textit{true}, caso não fosse adicionado o conhecimento deveria dar como resposta \textit{false}.

COLOCAR AQUI O PREDICADO

Para que não fosse possível adicionar, por exemplo, um cuidado “negativo” quando já existia um cuidado igual, criamos um invariante que procurava na base de conhecimento se já existia um cuidado com todos os parâmetros iguais ao conhecimento negativo que estamos a inserir.

COLOCAR AQUI O INVARIANTE

O mesmo raciocínio é semelhante para o caso de se querer inserir conhecimento positivo.

A abordagem utilizada, no caso de termos conhecimento desconhecido, e querermos inserir conhecimento positivo ou negativo, fez com que para estas situações, nós deixemos adicionar, não tendo de nos preocupar com esta situação em particular.

Os predicados apresentados anteriormente, são muito idênticos aos construídos para os predicados utente/4 e prestador/4.

Para além destes invariantes construímos outros que controlam, não só a nível estrutural, como a nível referencial, como por exemplo, o facto de um prestador não puder ter mais do que 3 serviços à mesma hora, ou então, não prestar um serviço numa instituição da qual não faz parte, entre outros. De salientar, que estes invariantes, já forma utilizados no trabalho prático anterior, mas era importante, voltar a referir e colocar neste trabalho, já que implicitamente tratam da evolução do conhecimento.

**“Inevolução” da Base de Conhecimento**

A “Inevolução” consiste na remoção do conhecimento da base. Para realizar esta ação, tivemos de construir um predicado inevolução, muito semelhante ao predicado evolução, mas com o intuito de avaliar se era permitido remover tal conhecimento da base.

No caso de querer remover conhecimento negativo, permitimos sempre, já que não afeta em nada a base de conhecimento. Já a remoção de conhecimento positivo, já não é assim tão trivial. Decidimos que um cuidado, pode ser removido, porém caso se pretenda remover um prestador ou então um utente, tal só é permitido se o mesmo não possuir registos de cuidados. Para este caso apenas tivemos de construir um invariante que verificava se o utente ou prestador em causa tinha algum registo de um cuidado, quer esse registo seja positivo, negativo ou do tipo Desconhecido.

No caso de conhecimento de tipo desconhecido, não deixamos retirar, por ser, mais uma vez, uma ação algo complexa para ser tratada com uma ferramenta como o PROLOG. Por exemplo, caso existisse um conhecimento Desconhecido do tipo impreciso no parâmetro do “custo”, entre 10 e 50, caso se pretendesse remover que era desconhecido para o valor de 15, teríamos de dividir o predicado presente em dois, para que pudesse criar duas áreas de imprecisão, nos intervalos [10, 15[ e ]15,50].