

# Revisão de Conceitos e Lógicas de Descrição

Orientadora: Renata Wassermann

Aluno: Márcio Moretto Ribeiro

26 de Abril de 2006

## 1 Introdução

Revisão de crenças é a sub-área da inteligência artificial, extensivamente estudada durante os últimos vinte anos [Gär88, Har86, Han97], que lida com a atualização de uma base de conhecimento. Como mostraremos mais adiante, nem sempre uma nova informação pode ser simplesmente adicionada à base.

Lógicas de Descrição [BCM<sup>+</sup>03] compõem uma família de linguagens para representação de conhecimento que são adequadas para definir ontologias que, por sua vez, servem para descrever domínios de forma não ambígua.

A necessidade da revisão de crenças nesse contexto decorre do fato de que ao se descrever domínios é comum aparecerem inconsistências. Por exemplo [SC03] mostra um caso real tirado da terminologia DICE <sup>1</sup>

$$\begin{aligned} Brain &\sqsubseteq CentralNervousSystem \sqcap \\ \exists systempart.NervousSystem \sqcap BodyPart \sqcap & \\ \exists region.HeadAndNeck \sqcap \forall region.HeadAndNeck & \end{aligned} \quad (1)$$

$$CentralNervousSystem \sqsubseteq NervousSystem \quad (2)$$

Mas nessa ontologia os conceitos NervousSystem e BodyPart são definidos como disjuntos e logo essa ontologia é inconsistente.

Nas próximas seções mostraremos as principais motivações do estudo da revisão de crenças em lógicas de descrição. Faremos um pequeno resumo dessas duas áreas da inteligência artificial, apresentaremos alguns resultados referentes a compatibilidade entre essas áreas encontrados na literatura ou

---

<sup>1</sup><http://kik.amc.uva.nl/dice/home.jsp>

que obtivemos durante a iniciação científica e por fim descreveremos como pretendemos dar continuidade a esse trabalho durante o mestrado.

## 2 Motivações

Com o avanço da tecnologia, um grande desafio passou a ser a capacidade de lidar com uma quantidade enorme de informações. Qualquer agente que possa ser chamado de “inteligente” necessita de um mecanismo para decidir o que fazer com novas informações recebidas. Nem sempre uma nova informação pode ser simplesmente adicionada a uma base de dados, sem que vários aspectos sejam verificados, por exemplo:

- se a fonte da informação é confiável;
- se a informação é consistente com dados anteriores;
- quais implicações podem ser derivadas da nova informação;
- quais dados devem ser atualizados em vista da nova informação;
- quais as conseqüências destas atualizações.

Dado um agente com um conjunto de crenças (dados, conhecimento), como ele deve modificar estas crenças ao receber uma nova informação? Essa é a formulação mais genérica do problema de Revisão de Crenças. Um agente para nós é qualquer tipo de sistema ao qual se pode atribuir crenças e do qual se espera reações racionais. Esse é um problema multidisciplinar, com aplicações em diversas áreas. Podemos citar alguns exemplos de revisão de crença do modo como o problema aparece em:

- Dia-a-dia: Eu acreditava que em Amsterdã sempre chovia. Um dia, ao acordar em Amsterdã, percebi que fazia sol. Eu acreditei que naquele dia não estava chovendo, contradizendo minha crença anterior. Eu tive de deixar de acreditar que sempre chovia lá.
- Banco de dados: Num banco de dados contendo informações sobre os clientes de uma livraria, há uma entrada para João da Silva com a sua data de nascimento, 20/2/67. A livraria recebe um novo pedido, vindo de João da Silva, mas onde como data de nascimento consta 20/2/76.

Não é possível adicionar uma nova data de nascimento e a data de nascimento de João não pode ter mudado com o tempo. O que fazer? Manter a data anterior? Substituí-la pela nova data? Ou esse pedido vem de um outro João da Silva que deve ser adicionado ao banco de dados?

- Robótica: Um robô possui um mapa da região por onde ele deve se mover. No mapa, não há nada a sua frente, então o robô deveria ser capaz de seguir em frente. Mas os seus sensores indicam a presença de um obstáculo. O robô deve duvidar de seus sensores e continuar tentando seguir em frente? Ou ele deve acreditar nos seus sensores e duvidar da exatidão do mapa?
- Diagnósticos: Eu acredito que ao colocar um artigo na posição correta em uma fotocopiadora, eu obtenho cópias do artigo. Eu coloco um artigo na posição correta, mas tudo o que eu obtenho são páginas em branco. Eu devo deixar de acreditar que aquela é a posição correta? Ou que a fotocopiadora está funcionando corretamente?

Modelos clássicos de revisão de crenças [Gär88] assumem o agente muito idealizado, por exemplo com memória e tempo de raciocínio ilimitados. Agentes reais, porém, claramente não possuem tais propriedades, logo, para se modelar revisão de crenças em agentes com recursos limitados, o modelo de alguma forma deve ser modificado [Was99]. Estudos com bases de crenças [Han97], por exemplo, tratam de conjuntos de crenças finitos, tornando a área de revisão de crenças bem mais interessante do ponto de vista computacional.

Em [Was03] foram propostas três formas de se fazer revisão de crenças em agentes menos idealizados: a primeira seria limitando a capacidade de acesso a memória, a segunda limitando o poder de inferência de um agente e por último, tolerando inconsistências. Uma outra forma de se tentar fazer revisão em agentes reais seria diminuindo a expressividade da lógica usada, usando, por exemplo, alguma lógica de descrição no lugar da lógica de primeira ordem. Lógicas de descrição, por serem menos expressivas do que a lógica de primeira ordem, geralmente são decidíveis e, na prática, inclusive computacionalmente viáveis [LB87].

Na literatura tem sido amplamente defendido que o conhecimento deve ser representado *funcional* e não *estruturalmente* [Neb90]. Para tanto um sistema de inferência deve ser especificado por uma interface do tipo “Tell &

Ask”, em que a função *Tell* adiciona conhecimento a base de conhecimentos e *Ask* faz consultas na mesma.

Seria interessante que a interface “Tell & Ask” dos sistemas de representação de conhecimento pudessem lidar com novas entradas inconsistentes ao seu conjunto de asserções atual ou pudessem remover (interface “Tell, Ask & Forget”) conhecimento de tal conjunto de uma maneira funcional. Para tanto é necessário que tais sistemas possam realizar alguma forma de revisão de crenças.

Uma outra motivação atual para o estudo proposto é o aparecimento da Web-Semântica [BLHL01], com o uso de ontologias para marcar páginas web. As linguagens adotadas como padrão para descrever ontologias na web, OWL e seus fragmentos OWL-DL e OWL-Lite [DSB<sup>+</sup>04] são baseadas em lógicas de descrição. Problemas de inferência em ontologias podem ser transformados em problemas de inferência nas lógicas correspondentes: SHOIN(D) [BHS05, HPS03] para OWL-DL e SHIF(D) no caso de OWL-Lite. Seria interessante para a realização da web-semântica que fosse possível importar ontologias, ou seja, mesclá-las. Porém, ao fazer isso, é possível que apareçam inconsistências e portanto seria indispensável um mecanismo para revisão de crenças.

Em [Was98] foi proposta uma forma de interpretar revisão de conceitos, porém esse trabalho só se preocupou com a semântica. Durante a iniciação científica começamos a estudar aspectos lógicos da compatibilidade entre lógicas de descrição e revisão de crenças. Mostramos que, apesar de não ser possível em qualquer lógica de descrição encontrar uma contração que satisfaça os postulados AGM [FPA04, FPA05], modificando-se um dos postulados, como veremos, o problema desaparece. O que pretendemos agora é prosseguir o estudo sendo feito durante a iniciação científica e implementar as idéias estudadas.

## 3 Resumo da Bibliografia

### 3.1 Lógicas de Descrição

Em vários problemas de inteligência artificial é interessante em algum ponto poder de alguma forma armazenar conhecimento adquirido e extrair conhecimento explícita ou implicitamente armazenado. Para isso é necessário um formalismo que represente simbolicamente o conhecimento e seja capaz de raciocinar sobre esse conhecimento simbolicamente armazenado, ou seja inferir

consequências implícitas à base de conhecimento.

As lógicas de descrição têm se tornado um importante formalismo para representação de conhecimento, principalmente para descrever conceitos e ontologias. Dentre as vantagens de se usar lógicas de descrição para representação de conhecimento está o fato delas possuírem uma semântica bem definida (são subconjuntos da lógica de primeira ordem) e serem decidíveis.

O preço pago pela decidibilidade, porém, é uma diminuição na expressividade de tais lógicas [LB87]. Ultimamente muitos trabalhos tem sido apresentados no sentido de achar as lógicas mais expressivas que ainda se mantêm decidíveis (para mais detalhes consulte [BCM<sup>+</sup>03]).

Seguiremos a convenção de usar palavras com letras maiúsculas (BIANCA, CAIO ...) para representar indivíduos, usaremos palavras começadas por maiúscula (Homem, Pai ...) para representar conceitos e palavras em minúsculas (temfilho ...) para papéis (“roles”).

Uma base de conhecimento em lógica de descrição é tipicamente dividida em duas partes, um *TBox* que contém conhecimento dito “intensional”, ou seja, define as propriedades dos conceitos e um *ABox* que contém conhecimento dito “extensional”, ou conhecimentos específicos sobre indivíduos.

Um *TBox* é constituído de axiomas como:

- $\text{Homem} \sqsubseteq \text{Pessoa}$
- $\text{Pai} \equiv \text{Homem} \sqcap \exists \text{temfilho.Pessoa}$

Ou seja, todo homem é uma pessoa e um pai é um homem que tem (pelo menos) um filho que é pessoa.

A principal forma de inferência que se pode fazer em uma terminologia (*TBox*) é verificar se um determinado conceito é mais geral do que outro, ou seja se  $C \sqsubseteq D$ . Tal forma de inferência é chamada de subordinação de conceitos (concept subsumption). Todo sistema gerenciador de conhecimento em lógica de descrição deve ser capaz de fazer tal inferência.

Um *ABox* contém sentenças sobre indivíduos, que podem ser afirmações sobre conceitos como:

- Mulher(BIANCA)

ou sobre relações entre indivíduos (“roles”) como:

- irmão(MÁRCIO, CAIO)

A principal inferência que um *ABox* deve ser capaz de realizar é chamada de *instanciação*, ou seja, verificar se determinado indivíduo é ou não instância de um conceito.

Como colocamos anteriormente as lógicas de descrição formam uma família de formalismos e não um único. O que diferencia umas das outras é o conjunto de construtores permitidos. Por exemplo: a lógica ALC [BCM<sup>+</sup>03] permite o uso da união ( $\sqcup$ ) e intersecção ( $\sqcap$ ) entre conceitos, complemento ( $\neg$ ) de um conceito e restrição de valores dos papéis ( $\forall$ ); a lógica SH permite tudo que a ALC permite mais papéis (“roles”) transitivos e hierarquia de papéis.

Lógicas de descrição diferentes podem possuir expressividade ou complexidade diferentes. A lógica de descrição SHOIN(D) é bastante expressiva, mas ainda assim é decidível, por essa razão ela foi escolhida para ser a lógica relacionada à linguagem OWL-DL. Por esse motivo daremos uma importância especial a essa lógica.

## 3.2 Revisão de Crenças

Revisão de crenças consiste no estudo dos *estados epistêmicos* e sua dinâmica, fornecendo uma representação para os *elementos epistêmicos* (crenças) e um *critério de racionalidade*. Grande parte da literatura segue o paradigma proposto em [AGM85], conhecido como AGM devido às iniciais dos autores Alchourrón, Gärdenfors e Makinson. Nos focaremos primeiramente na representação baseada em *conjuntos de crenças* [Gär88, AGM85] e depois no modelo de *bases de crenças* [Han97].

### 3.2.1 Conjuntos de Crenças

Usaremos letras gregas ( $\alpha, \beta, \gamma, \dots$ ) para representar sentenças, letras maiúsculas para conjuntos e os símbolos  $\top$  para tautologia e  $\perp$  para contradição. O operador  $Cn$  indicará o conjunto das consequências lógicas (lógica proposicional clássica) de um conjunto de fórmulas.

**Definição 1** *Um conjunto de sentenças  $K$  é chamado de conjunto de crenças sse:*

$$K = Cn(K), \text{ ou seja, se } K \text{ for logicamente fechado} \quad (3)$$

### 3.2.2 Expansão, Revisão e Contração

Uma mudança em um conjunto de crenças pode ser de três tipos:

**(expansão)** Quando simplesmente adicionamos uma proposição  $\alpha$  ao conjunto de crenças  $K$ . Notação:  $K + \alpha$

**(contração)** Quando queremos retirar uma proposição  $\alpha$  do conjunto de crenças  $K$ . Notação:  $K - \alpha$

**(revisão)** Quando queremos adicionar uma proposição  $\alpha$  ao conjunto, mas queremos manter a consistência do conjunto de crenças  $K$ . Notação:  $K * \alpha$

Seja  $K$  nosso conjunto de crenças, ao adicionar uma sentença nova a ele devemos também adicionar todas as conseqüências dessa nova sentença junto com nossas crenças antigas, ou seja:

$$K + A = Cn(K \cup A) \quad (4)$$

Isso é o que chamamos de expansão.

Fazer uma revisão ou contração não é tão simples quanto uma expansão. No paradigma AGM, são apresentados conjuntos de postulados que uma operação de revisão ou contração deve satisfazer, chamados de *postulados de racionalidade*. São propostas também algumas construções que satisfazem os postulados. Estas construções têm a característica de que, além de satisfazerem os postulados, qualquer operação que satisfaça os postulados pode ser construída desta forma. Os teoremas que provam isto são chamados de *teoremas de representação* e demonstram a equivalência entre as diversas construções.

Neste texto, nos concentraremos na apresentação da operação de contração, pois no paradigma AGM a revisão pode ser derivada a partir das operações de contração e expansão utilizando-se a *Identidade de Levi*:  $K * \alpha = (K - \neg\alpha) + \alpha$ .

Sempre que desejamos desistir de uma crença devemos fazer uma *contração*. Repare que se desejamos tirar uma sentença  $\alpha$  de nosso conjunto de crenças não basta apenas tirar  $\alpha$ , pois o conjunto remanescente pode implicar em  $\alpha$  e simplesmente tirá-la implicaria que o novo conjunto não seria logicamente fechado.

Esses são os axiomas que, segundo o paradigma AGM, uma contração deve obedecer:

- (K<sup>-1</sup>)  $K - \alpha$  é um conjunto de crença
- (K<sup>-2</sup>)  $K - \alpha \subseteq K$
- (K<sup>-3</sup>) Se  $\alpha \notin K$ , então  $K - \alpha = K$
- (K<sup>-4</sup>) Se  $\alpha \notin Cn(\emptyset)$ , então  $\alpha \notin K - A$
- (K<sup>-5</sup>) Se  $\alpha \in K$ , então  $K \subseteq (K - \alpha) + \alpha$
- (K<sup>-6</sup>) Se  $Cn(\alpha) = Cn(\beta)$ , então  $K - \alpha = K - \beta$

### 3.2.3 Contração Partial Meet

Os postulados descritos tanto para *revisão* quanto para *contração* não caracterizam totalmente uma função, eles colocam restrições a possíveis construções. Vamos mostrar agora uma possível maneira de se construir uma *função de contração* que satisfaça os postulados propostos. Trataremos da *contração partial meet* que consiste em tomar a intersecção de alguns dos subconjuntos de K maximais que não implicam  $\alpha$ .

**Definição 2** *Seja K um conjunto de crenças. K' é um subconjunto maximal de K que não implica  $\alpha$  sse:*

- $K' \subseteq K$
- $\alpha \notin Cn(K')$
- Se  $K' \subset K'' \subseteq K$  então  $K'' \vdash \alpha$

$K \perp \alpha$  denota o conjunto de todos os subconjuntos maximais de K que não implicam  $\alpha$ .

Escolhendo alguns conjuntos de  $K \perp \alpha$  e fazendo a intersecção deles, temos o que chamamos de *partial meet*. Mais formalmente:

**Definição 3** *Seja K um conjunto de sentenças. Uma função de seleção para K é uma função  $\gamma$  tal que para toda sentença  $\alpha$ :*

- Se  $K \perp \alpha \neq \emptyset$  então  $\gamma(K \perp \alpha) \neq \emptyset$  e  $\gamma(K \perp \alpha) \subseteq K \perp \alpha$
- Caso contrário,  $\gamma(K \perp \alpha) = \{K\}$

**Definição 4** *Sejam  $K$  um conjunto de sentenças,  $\alpha$  uma sentença e  $\gamma$  uma função seleção. Chamamos o conjunto  $\bigcap \gamma(K \perp \alpha)$  de contração parcial meet em  $K$  gerada a partir de  $\gamma$ .*

Alchourrón, Gärdenfors e Makinson provaram o seguinte teorema de representação:

**Teorema 1** [AGM85] *Uma contração é partial meet se e somente se satisfizer  $(K^-1)$ - $(K^-6)$ .*

## 4 Resultados Anteriores

Nesta seção, mostramos alguns resultados obtidos na aplicação de revisão de crenças a lógicas de descrição. Começamos mostrando rapidamente algumas definições e resultados de Flouris, Plexousakis e Antoniou em seguida, nossos resultados preliminares.

Seguindo [FPA04] chamaremos de lógica o par  $\langle L, C \rangle$  onde  $L$  é o conjunto de símbolos nessa lógica e  $C$  é o operador consequência lógica, substituindo Cn por C.

Usando essa definição de lógica podemos reescrever os postulados AGM sem ter de supor que a lógica usada seja proposicional clássica de forma análoga a que foi apresentada anteriormente.

O operador consequência  $C$  pode possuir diversas propriedades <sup>2</sup>, em particular estamos interessados em operadores tarskianos, ou seja:

**Definição 5** *Seja  $\langle L, C \rangle$  uma lógica e sejam  $A$  e  $B$  conjuntos de fórmulas em  $L$ . Chamaremos o operador  $C$  de tarskiano sse ele satisfaz as seguintes propriedades para todo  $A$  e  $B$ :*

- $C(C(A)) = C(A)$  (idempotência)
- $A \subseteq C(A)$  (inclusão)
- $A \subseteq B \Rightarrow C(A) \subseteq C(B)$  (monotonicidade)

---

<sup>2</sup>para uma lista de propriedades de um operador consequência descritas na literatura consulte [Was99].

Em [FPA04, FPA05] foi mostrado que nem toda lógica tarskiana possui operador de contração satisfazendo os seis postulados AGM. Os autores apresentaram um critério para decidir se uma lógica é AGM compatível (uma lógica onde existe operador de contração satisfazendo os postulados AGM).

**Definição 6** [FPA04] *Uma lógica  $\langle L, C \rangle$  é dita decomponível sse para todo  $A \subseteq L$  e para todo  $B$  tal que  $C(\emptyset) \subset C(B) \subset C(A)$  existe  $D$  tal que  $C(D) \subset C(A)$  e  $C(B \cup D) = C(A)$ .*

**Teorema 2** [FPA04] *Uma lógica é AGM compatível sse ela for decomponível*

Em particular, a lógica de descrições SHOIN não é decomponível. Este foi um resultado negativo obtido pelo grupo que queria aplicar o paradigma AGM diretamente sobre a lógica SHOIN, base da linguagem OWL. Um exemplo de lógica que é decomponível é a lógica ALC.

No nosso trabalho, conseguimos mostrar que utilizando um conjunto alternativo de postulados para a contração, é possível definir operações de contração no estilo AGM para qualquer lógica tarskiana. Para isso, basta substituir o postulado da recuperação pelo seguinte postulado:

**(relevância)** Se  $\beta \in K \setminus K - \alpha$  então existe  $K'$  tal que  $K - \alpha \subseteq K' \subseteq K$  e  $\alpha \notin Cn(K')$  mas  $\alpha \in Cn(K' \cup \{\beta\})$ .

**Lema 1** *Toda lógica  $\langle L, C \rangle$  (tarskiana) possui operador contração satisfazendo os postulados AGM porém trocando o postulado recuperação pelo critério da relevância.*

O conjunto  $\bigcap(K \perp \alpha)$  (intersecção de todos os subconjuntos maximais de  $K$  que não implicam em  $\alpha$ ), por exemplo, satisfaz esses postulados. Em [Gär88] foi provado que esse conjunto satisfaz todos os postulados para lógica proposicional clássica, baseando-se nessa demonstração provamos que  $\bigcap(K \perp \alpha)$  satisfaz os postulados  $(K^{-1})$  até  $(K^{-4})$  e  $(K^{-6})$ . Para provar  $(K^{-5})$  basta notar que como  $K' \in K \perp \alpha$ ,  $\alpha \notin K'$  e como  $K' \subset K' \cup \{\beta\} \subseteq K$  então  $\alpha \in C(K' \cup \{\beta\})$  e por fim temos que  $\bigcap(K \perp \alpha) \subseteq K' \subseteq K$ .

Essa troca de postulados se justifica pois, em primeiro lugar, o critério da relevância já é amplamente aceito em bases de crenças [Han97]. Além disso, o postulado da recuperação é o mais polêmico e já recebeu diversas críticas na literatura [Mak87, Fer01]. Hansson mostrou que no caso da lógica proposicional, os dois conjuntos de postulados são equivalentes:

**Proposição 1** [Han97] *Para lógica proposicional clássica vale que se um operador de contração satisfaz os postulados AGM menos o postulado da recuperação e satisfaz o critério da relevância então esse operador satisfaz o postulado da recuperação.*

Como as lógicas de descrição são tarskianas, elas possuem operador de contração que satisfaz esse novo conjunto de postulados, em particular a lógica SHOIN também possui. A consequência disso é que podemos aplicar revisão de crenças no estilo AGM a ontologias descritas em OWL.

## 5 Bases de Crenças

Como estamos preocupados com aspectos computacionais do problema, logo notamos que o estudo de conjuntos de crenças não é totalmente adequado, pois conjuntos de crenças tendem a ser muito grandes (muitas vezes infinitos).

Para contornar esse, e outros problemas, foi proposta a idéia das bases de crenças [Han97].

Bases de crenças são apenas conjuntos de proposições, porém a idéia de contração aqui não é apenas retirar uma sentença da base, mas tirá-la do conjunto das consequências dela.

Como foi feito com conjuntos de crenças podemos definir uma contração partial meet em bases de crenças:

$$K - \alpha = \bigcap \gamma(K \perp \alpha) \quad (5)$$

Porém agora não exigimos que  $K$  seja logicamente fechado.

Podemos também definir um conjunto de postulados que caracterizam essa contração totalmente, ou seja, pode-se provar o seguinte teorema da representação:

**Teorema 3** [Han97] *Uma contração em bases de crenças é uma contração partial meet ( $\bigcap \gamma(K \perp \alpha)$ ) sse ela satisfaz os seguintes postulados:*

**(sucesso)** *Se  $\alpha \notin C(\emptyset)$  então  $\alpha \in K - \alpha$*

**(inclusão)**  *$K - \alpha \subseteq K$*

**(relevância)** *Se  $\beta \in K \setminus K - \alpha$  então existe  $K'$  tal que  $K - \alpha \subseteq K' \subseteq K$  e  $\alpha \notin Cn(K')$  mas  $\alpha \in Cn(K' \cup \{\beta\})$ .*

**(uniformidade)** Se para todo  $K' \subseteq K$  vale que  $\alpha \in C(K')$  sse  $\beta \in C(K')$  então  $K - \alpha = K - \beta$ .

Uma outra forma de se construir o operador contração em bases de crenças é chamado de contração “kernel” (núcleo). A idéia basicamente é tirar da base de crenças  $K$  um elemento de cada um dos conjuntos minimais que não implicam  $\alpha$ , ou seja:

**Definição 7**  $X \in K \perp\!\!\!\perp \alpha$  sse:

- $X \subseteq K$
- $\alpha \in Cn(X)$
- Se  $Y \subset X$  então  $\alpha \notin Cn(Y)$

**Definição 8** Chamamos uma função  $\delta$  de função de incisão sse:

- $\delta(K \perp\!\!\!\perp \alpha) \subseteq \bigcup (K \perp\!\!\!\perp \alpha)$
- Se  $X \neq \emptyset$  e  $X \in K \perp\!\!\!\perp \alpha$  então  $X \cap \delta(K \perp\!\!\!\perp \alpha) \neq \emptyset$

**Definição 9** Uma contração kernel é definida como:

$$K -_{\sigma} \alpha = K \setminus \sigma(K \perp\!\!\!\perp \alpha) \tag{6}$$

A contração kernel também pode ser caracterizada por um conjunto de postulados, porém um conjunto um pouco diferente daquele que caracteriza a contração partial meet. Na verdade a única diferença é que o postulado da relevância deve ser substituído pelo seguinte postulado:

**(core-retainment)** Se  $\beta \in K \setminus K - \alpha$  implica que exista  $K'$  tal que  $K' \subseteq K$  e  $\alpha \notin Cn(K')$  mas  $\alpha \in Cn(K' \cup \{\beta\})$ .

Então pode-se provar o respectivo teorema da representação.

## 6 Proposta de Trabalho

Durante o mestrado, pretendemos dar continuidade ao estudo teórico da aplicação de revisão de crenças a lógica de descrições e implementar um protótipo de revisor de ontologias.

Acreditamos que isso é viável, pois diferente das contrações em conjuntos de crenças, contrações em bases de crenças podem ser aplicadas a qualquer lógica monotônica e compacta, conforme provado em [HW02]. Lógicas de descrição são monotônicas, resta mostrar se elas são compactas para podermos aplicar contração em bases de crenças.

Como mostramos acima, uma forma interessante de se fazer uma contração ( $K - \alpha$ ) é a chamada contração kernel que consiste em pegar um elemento de cada subconjunto minimal de  $K$  que implica em  $\alpha$  e tirá-lo de  $K$ . A parte complicada desse processo é encontrar os subconjuntos minimais de  $K$  que não implicam em  $\alpha$ .

Em [SC03, KPSH05] é apresentado um algoritmo (para ALC e SHOIN respectivamente) para encontrar o que foi chamado de MUPS (Minimal Unsatisfiability Preserving Sub-TBox) de uma TBox, ou seja, os menores subconjuntos de uma TBox inconsistente que ainda são inconsistentes. A idéia do algoritmo é usar um reasoner para achar inconsistências em um TBox, mapear esse processo e assim identificar as “causas” (MUPS) da inconsistência. Como foi mostrado em [SC03] isso não altera a complexidade do problema de encontrar inconsistências em uma TBox.

Acreditamos que algumas alterações nesse algoritmo façam com que ele encontre o “kernel” de uma TBox em relação a uma fórmula. Poderíamos, então, usar esse “kernel” para fazer uma contração no TBox. Pretendemos desenvolver melhor essa idéia durante o mestrado e então implementar um protótipo de revisor de ontologias.

## 7 Cronograma

### [Primeiro Semestre]

Prosseguir o estudo de bases de crenças [Han97, Was99] (contração, revisão, consolidação e semi-revisão). Verificar em que condições podemos aplicar contração de bases de crenças em TBox. Para isso temos que mostrar se as lógicas de descrição são compactas (em particular a

lógica SHOIN), estudar em que situações poderíamos aplicar as contração de bases de crenças em lógicas de descrição.

### [Segundo Semestre]

Estudar o algoritmo apresentado em [SC03, KPSH05] e como teríamos que adaptá-lo para que ele encontrasse as menores TBox que implicam em uma sentença arbitrária.

### [Terceiro Semestre]

Implementar um revisor de ontologias aplicando os estudos anteriormente.

### [Quarto Semestre]

Terminar a implementação, testar o programa e escrever a dissertação.

Os créditos necessários serão cursados durante os três primeiros semestres, juntamente com a pesquisa.

## Referências

- [AGM85] Carlos Alchourrón, Peter Gärdenfors, and David Makinson. On the logic of theory change. *Journal of Symbolic Logic*, 50(2):510–530, 1985.
- [BCM<sup>+</sup>03] Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge University Press, 2003.
- [BHS05] F. Baader, I. Horrocks, and U. Sattler. Description logics as ontology languages for the Semantic Web. In D. Hutter and W. Stephan, editors, *Mechanizing Mathematical Reasoning: Essays in Honor of Jörg H. Siekmann on the Occasion of His 60th Birthday*, volume 2605 of *Lecture Notes in Artificial Intelligence*, pages 228–248. Springer-Verlag, 2005.
- [BLHL01] Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web: A new form of web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American*, May, 2001.

- [DSB<sup>+</sup>04] Mike Dean, Gus Schreiber, Sean Bechhofer, Frank van Harmelein, Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein. OWL web ontology language reference. W3C Recommendation, 2004. Disponível em <http://www.w3c.org/TR/owl-ref>.
- [Fer01] Eduardo Fermé. Five faces of recovery. In Hans Rott and Mary-Anne Williams, editors, *Frontiers in Belief Revision*. Kluwer, 2001.
- [FPA04] Giorgos Flouris, Dimitris Plexousakis, and Grigoris Antoniou. Generalizing the AGM postulates: preliminary results and applications. In *In Proc. of the 10th International Workshop on Non-Monotonic Reasoning 2004 (NMR-04)*, pages 171–179, Westin Whistler Resort and Spa, Whistler BC, Canada, June 6-8 2004.
- [FPA05] Giorgos Flouris, Dimitris Plexousakis, and Grigoris Antoniou. Updating description logics using the AGM theory. In *In Proceedings of the 7th International Symposium on Logical Formalizations of Commonsense Reasoning*, Corfu, Greece, May 2005.
- [Gär88] P. Gärdenfors. *Knowledge in Flux: Modeling the Dynamics of Epistemic States*. MIT Press, Cambridge, 1988.
- [Han97] Sven-Ove Hansson. *A Textbook of Belief Dynamics*. Kluwer Academic Publishers, 1997.
- [Har86] Gilbert Harman. *Change In View: Principles of Reasoning*. Bradford Books. MIT Press, 1986.
- [HPS03] I. Horrocks and P. Patel-Schneider. Reducing OWL entailment to description logic satisfiability. In *Proc. of the 2nd International Semantic Web Conference (ISWC)*, 2003.
- [HW02] Sven Ove Hansson and Renata Wassermann. Local change. *Studia Logica*, 70(1):49–76, 2002.
- [KPSH05] Aditya Kalyanpur, Bijan Parsia, Evren Sirin, and James Hendler. Debugging unsatisfiable classes in OWL ontologies. *Journal of*

*Web Semantics - Special Issue of the Semantic Web Track of WWW2005*, 3(4), 2005.

- [LB87] Hector J. Levesque and Ronald J. Brachman. Expressiveness and tractability in knowledge representation and reasoning. *Computational Intelligence*, 3:78–93, 1987.
- [Mak87] David Makinson. On the status of the postulate of recovery in the logic of theory change. *Journal of Philosophical Logic*, 16:383–394, 1987.
- [Neb90] Bernhard Nebel. *Reasoning and revision in hybrid representation systems*. Springer-Verlag New York, Inc., New York, NY, USA, 1990.
- [SC03] S. Schlobach and R. Cornet. Non-standard reasoning services for debugging of description logic terminologies. In *Proceedings of the 15th Belgium-Netherlands Conference on Artificial Intelligence*, 2003.
- [Was98] Renata Wassermann. Revising concepts. In *Proceedings of the Fifth Workshop on Logic, Language, Information and Communication*, São Paulo, SP, BR, 1998.
- [Was99] Renata Wassermann. *Resource Bounded Belief Revision*. PhD thesis, Universiteit van Amsterdam, 1999.
- [Was03] Renata Wassermann. Generalized change and the meaning of rationality postulates. *Studia Logica*, 73(2):299–319, 2003.