



Revisão de Crenças em Lógicas de Descrição

MACo499 - Trabalho de Formatura Supervisionado

Luís Felipe de Melo Costa Silva

Orientadora: Prof^a Dra. Renata Wassermann

Instituto de Matemática e Estatística

Universidade de São Paulo

luis.melo.silva@usp.br



1. Introdução

A **ontologias** são sistemas usados para representar conhecimento de algum domínio, como a saúde ou o cinema. Elas são baseadas em classes, propriedades das classes e relações entre elas. As **Lógicas de Descrição** são usadas para formalizar as ontologias. Para construir as ontologias, pode-se usar um editor, como o *Protégé*¹.

A inclusão de algum novo conhecimento no sistema pode tornar a base de dados já existente inconsistente. Para consertar isso, podem ser usadas técnicas de **Revisão de Crenças**.

Para esse trabalho foi feito um estudo sobre as ontologias, sobre Lógicas de Descrição e Revisão de Crenças, com o objetivo de fazer um **plug-in** que implementasse algumas operações de Revisão de Crenças.

2. Ontologias e Lógicas de Descrição

Uma ontologia pode ser analisada, computacionalmente, como uma reunião de sentenças lógicas que exibem alguma informação sobre uma parte do mundo analisada. O objetivo delas é resolver problemas relacionados ao seu domínio, por isso, não se recomenda a construção de ontologias sobre algum problema específico. São modeladas com **Classes, Propriedades, Relações, Restrições e Instâncias**.

Para codificá-las, são necessárias linguagens formais. As linguagens naturais, como o português e o espanhol, não podem ser usadas para tal. A razão é simples: as linguagens naturais possuem figuras de linguagem, ou seja, um discurso pode ter diversas interpretações.

Portanto, as linguagens formais usadas para ontologias são as Lógicas de Descrição, sublinguagens da Lógica de Primeira Ordem. Para organizar o conhecimento, fazem uso de **Conceitos, Papéis e Indivíduos**, equivalentes às *Classes, Propriedades e Instâncias* referidos anteriormente, respectivamente. Pode-se ver que elas encaixam muito bem com as ontologias.

3. Revisão de Crenças

A área que cuida das alterações do estado de crenças, que é o que um agente acredita em um determinado momento, é chamada de Revisão de Crenças.

Quando se deseja editar uma ontologia, é necessário saber se essa alteração influencia no que já existe na ontologia, ou seja, se ela se torna inconsistente ou não. A inconsistência acontece quando existe um conceito e a sua negação ao mesmo tempo na ontologia. Existem três operações básicas nessa área, que são:

- **Expansão:** A mais básica das operações. Representa a adição de uma informação sem a preocupação com a consistência.

- **Contração:** Representa a retirada de um conhecimento e tudo que o implica, caso necessário.

- **Revisão:** Consiste na adição de uma informação com preopunção com a consistência, retirando sentenças problemáticas, se preciso.

Apenas a Expansão é definida diretamente. Para cada uma das outras duas operações básicas, são utilizados **oito** postulados de racionalidade. Computacionalmente, são usados construtores para a implementação das operações.

4. Construtores das operações

Para um conjunto de crenças K e uma sentença α , se definem os seguintes construtores:

Contração Kernel: Definida da seguinte forma:

$$K -_{\sigma} \alpha = K \setminus \sigma(K \perp \alpha),$$

onde $-_{\sigma}$ é a operação chamada de Contração *Kernel*, σ , uma **função de incisão**, que seleciona no mínimo uma fórmula de cada elemento de um conjunto-*Kernel*, e \perp , a representação de um **conjunto-Kernel**, que é o conjunto de todos os subconjuntos minimais de K que implicam α .

Contração Partial Meet: Escrita assim:

$$K -_{\gamma} \alpha = \bigcap \gamma(K \perp \alpha),$$

onde $-_{\gamma}$ é a operação chamada de Contração *Partial Meet*, γ é uma **função de seleção** que seleciona algumas crenças do conjunto, e \perp é a representação de um **conjunto-resíduo**, que é o conjunto de todos os subconjuntos maximais de K que não implicam α .

Pseudocontração SRW: Com a seguinte fórmula:

$$K -_{*} \alpha = \bigcap \gamma(\text{Cn}^*(K) \perp \alpha),$$

que é muito semelhante ao construtor anterior, exceto pelo uso de um operador de consequência lógica mais fraco do que o usual. Um operador de consequência é usado para inferir todo o conhecimento possível de uma base de conhecimento.

Revisão Kernel: Apresentada como:

$$K *_{\sigma} \alpha = K \setminus \sigma(K \Downarrow \alpha),$$

onde $*_{\sigma}$ é a operação chamada de Revisão *Kernel*, σ é uma função de incisão, e \Downarrow é a representação de um conjunto-*kernel* para a revisão, que é o conjunto de todos os subconjuntos minimais inconsistentes de K .

5. Implementação e resultados

O *plug-in* construído tem a interação com o usuário pela linha de comando. Ele precisa definir o

arquivo de entrada, o arquivo de saída, a operação e a fórmula de interesse.

O *plug-in* e seu código-fonte estão disponíveis online².

Para exemplificar a execução, foi feito um exemplo com os seguintes axiomas: $\text{MusicaBelica} \sqsubseteq \text{HinoNacional}$ e $\text{Marcha} \sqsubseteq \text{MusicaBelica}$. Sua hierarquia pode ser observada na Figura 1.



Figura 1: A hierarquia de classes da ontologia, no Protégé.

As fórmulas utilizadas para a Contração *Kernel*, Contração *Partial Meet* e Pseudocontração SRW foram a mesma "Marcha SubClassOf HinoNacional". As operações devolveram resultados diferentes.

Tanto a Contração *Kernel* quanto a Pseudocontração SRW forneceram o mesmo resultado, deixando todas as classes no mesmo nível hierárquico, como exibido na Figura 2.



Figura 2: A estrutura de classes após as operações Contração *Kernel* e Pseudocontração SRW.

Já a Contração *Partial Meet* fornece um resultado mais brando, mantendo o axioma $\text{Marcha} \sqsubseteq \text{MusicaBelica}$. Esse resultado parece ser mais promissor do que o anterior. Na Figura 3, é possível observar como fica a ontologia.

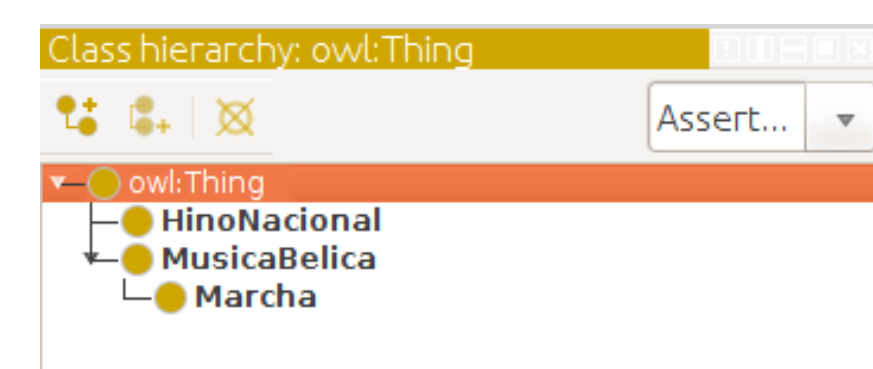


Figura 3: Hierarquia das classes após a Contração *Partial Meet*.

No outro oposto, temos a Revisão *Kernel*. Seu resultado é o mais radical. A fórmula utilizada foi "Marcha DisjointWith HinoNacional". Pode-se observar, na Figura 4, que esse axioma está presente na ontologia. No entanto, os outros dois axiomas não estão mais na ontologia, e a classe *MusicaBelica* também não existe mais na estrutura.

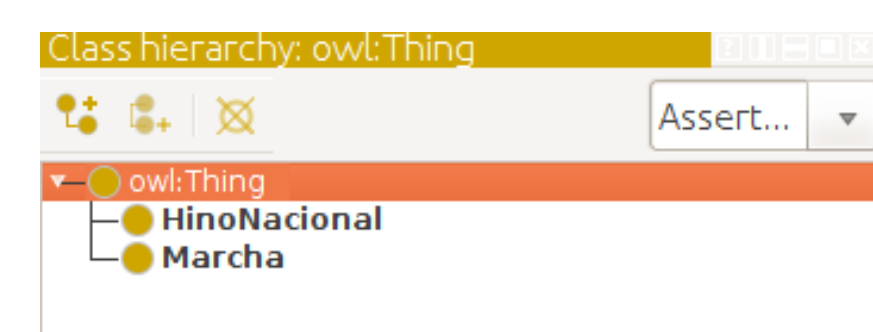


Figura 4: A estrutura da ontologia, radicalmente alterada após a Revisão *Kernel*.

¹<https://protege.stanford.edu/>

²<https://github.com/lslfp/ontology-repair>