Revisão de Crenças em Lógica de Descrição

MAC0499 - 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

As **ontologias** são sistemas usados para representar conhecimento de algum domínio, como a saúde ou o cinema. As **Lógicas de Descrição** são usadas para formalizar as ontologias.

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

Para este trabalho, foi feito um estudo sobre as ontologias, sobre Lógicas de Descrição e sobre Revisão de Crenças, com o objetivo de fazer um ***plug-in*** que implementasse 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 informações sobre uma parte do mundo analisada. O objetivo delas é resolver problemas relacionados ao seu domínio. São modeladas com **Classes**, **Propriedades**, **Relações**, **Restrições** e **Instâncias**.

Para formalizar as ontologias são usadas as Lógicas de Descrição, sublinguagens da Lógica de Primeira Ordem. Para organizar o conhecimento, elas fazem uso de **Conceitos**, **Papéis** e **Indivíduos**, equivalentes às *Classes*, *Propriedades* e *Instâncias* referidos anteriormente, respectivamente.

Para construir as ontologias e povoá-las com instâncias, pode-se usar um editor, por exemplo, o *Protégé*, cujo logo está na Figura 1. Com ele, é possível salvar as ontologias na linguagem OWL (sigla para *Ontology Web Language*).

Figura 1: *O logotipo do Protégé.*

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 essa alteração causa uma inconsistência. Isso acontece quando é possível inferir uma asserção de 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 na ontologia, sem a preocupação com a consistência.
- **Contração**: Representa a retirada de um conhecimento da base e tudo que o implica, caso necessário.
- **Revisão**: Consiste na adição de uma informação no sistema com preocupação com a consistência, retirando sentenças problemáticas, se preciso.

Apenas as Expansão é definida diretamente. Para cada uma das outras duas operações básicas, são utilizados **oito** postulados de racionalidade.

Computacionamente, são usados construtores para a implementação das operações. Tais construtores são espécies de fórmulas, que tentam passar uma "receita" de como fazer cada operação.

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:

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

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 $\perp\!\!\!\perp$, 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:

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

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.

Revisão *Kernel*: Apresentada como:

$$K *_\sigma \alpha = K \setminus \sigma(K \downarrow\!\!\downarrow \alpha),$$

onde $*_\sigma$ é a operação chamada de Revisão *Kernel*, σ é uma **função de incisão**, e $\downarrow\!\!\downarrow$ é a representação de um **conjunto-*kernel* para a Revisão**, que é o conjunto de todos os subconjuntos minimais inconsistentes de K após a adição de α.

5. Implementação e resultados

O *plug-in* construído tem a interação com o usuário pela linha de comando. Ele precisa, obrigatoriamente, definir o arquivo de entrada, o arquivo de saída, a operação e a fórmula de interesse.

Para exemplificar a execução, foi feita uma ontologia com os seguintes axiomas: `MusicaBelica` $\sqsubseteq$ `HinoNacional` e `Marcha` $\sqsubseteq$ `MusicaBelica`. Sua hierarquia pode ser observada na Figura 2.

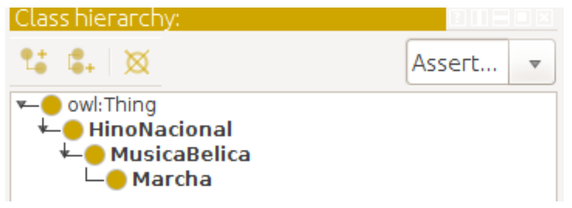

Figura 2: *A hierarquia de classes da ontologia, no Protégé.*

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 3.

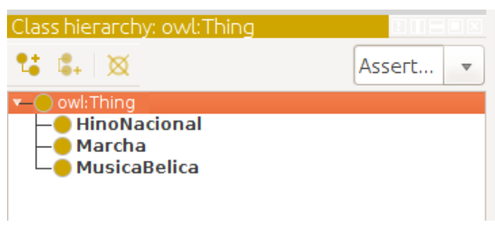

Figura 3: *A estrutura de classes após as operações Contração Kernel e Pseudocontração SRW.*

Já a Contração *Partial Meet* fornece um resultado mais brando, mantendo o axioma `Marcha` $\sqsubseteq$ `MusicaBelica`. Esse resultado parece ser mais promissor do que o anterior. Na Figura 4, é possível observar como fica a ontologia.

Figura 4: *Hierarquia das classes após a Contração Partial Meet.*

No outro oposto, temos a Revisão *Kernel*. Seu resultado é o mais radical. A fórmula utilizada aqui foi `"Marcha DisjointWith HinoNacional"`. Pode-se observar, na Figura 5, 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.

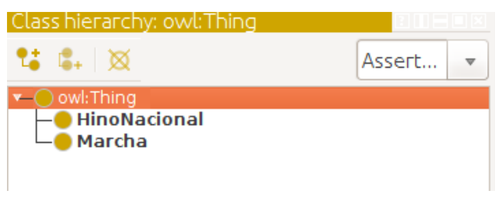

Figura 5: *A estrutura da ontologia, radicalmente alterada após a Revisão Kernel.*