

Implementação de Revisão de Modelos CTL

DUILIO FELIX OLIVEIRA DA SILVA

Universidade de São Paulo, Brasil
duilio@linux.ime.usp.br

Revisão de crenças é uma área da inteligência artificial que trata da atitude de um agente inteligente rever ou atualizar sua base de conhecimentos. Esta área e suas variantes são muito utilizadas em diversos tipos de projetos, tais como na programação de robôs ou em alterações em bancos de dados. Neste trabalho, utilizaremos o framework criado por Renato Urquiza Lundberg, o **bcontractor** [2], para implementar os algoritmos de revisão de modelos CTL descritos na tese de Paulo de Tarso [1] e assim, fazer uma ligação entre a teoria e a prática da revisão de crenças.

Revisão de Crenças

Revisão de crenças é uma área de inteligência artificial que estuda como um agente inteligente (o programa de interesse) deve se comportar ao receber informações novas para a sua base de conhecimentos. Em revisão de crenças, o conhecimento de um sistema é representado por uma base de conhecimento, que é um conjunto de asserções lógicas, ou crenças.

Um conjunto de crenças consistente pode ter três atitudes em relação a uma crença: Aceitar, rejeitar ou tomá-la como indeterminada. Normalmente, esse conjunto é mutável, e uma mudança nele pode ser resumida a uma mudança de atitude para determinada crença.

Essas mudanças podem ser divididas em três tipos:

- Quando uma crença α é indeterminada e passa a ser aceita ou rejeitada, a mudança é denominada de **expansão**.
- Quando a crença α é aceita ou rejeitada e passa a ser indeterminada, chama-se **contração**.
- Quando a crença α muda de aceita para rejeitada ou vice-versa, a mudança é denominada de **revisão**.

Para se fazer essas operações, geralmente seguimos os postulados de racionalidade descritos na Teoria AGM, cujo nome vem de seus criadores, Carlos E. Alchourrón, Peter Gärdenfors e David Makinson. Estes postulados definem uma série de propriedades que fazem com que as operações sigam o princípio de mudança mínima, ou seja, as operações fazem somente a quantidade necessária de mudanças no conjunto de crenças sem remover ou adicionar informações desnecessariamente.

Algoritmos

Olhando para a parte prática, existem algumas construções desses operadores como a **partial meet** e o sistema de esferas. Algumas destas construções podem ser implementadas para diversas lógicas diferentes (não só a proposicional).

Porém, aqui estamos interessados em uma variante da revisão de crenças, a revisão de modelos. Mais especificamente, a revisão de modelos CTL com os algoritmos propostos por Paulo. O trabalho consiste em implementar este algoritmo e, se possível, analisá-lo do ponto de vista prático.

CTL

Antes de discutirmos sobre o algoritmo proposto, vamos falar sobre a “linguagem” que ele atende, a Lógica de Árvore Computacional, ou CTL (do inglês, **computation tree logic**). A CTL é uma lógica modal capaz de representar e raciocinar sobre o tempo, ou seja, ela é capaz de expressar acerções que envolvem condições temporais. Seu nome vem do fato de que a CTL modela o futuro como ramificações de uma árvore. Sua definição é dada pela seguinte **Backus Naur Form**:

$$\phi ::= \top \mid \perp \mid p \mid (\neg\phi) \mid (\phi \vee \phi) \mid (\phi \wedge \phi) \mid (\phi \rightarrow \phi) \mid EX\phi \mid AX\phi \mid EF\phi \mid AF\phi \mid EG\phi \mid AG\phi \mid E[\phi U \phi] \mid A[\phi U \phi]$$

onde p é um átomo proposicional, \neg , \vee , \wedge e \rightarrow são os operadores clássicos de lógica proposicional.

Os operadores temporais da CTL são constituídos de um quantificador de caminhos (E , “existe um caminho”, ou A , “para todos os caminhos”) seguido de um operador de estado (X , “próximo estado no caminho”, U , “até que no caminho”, G , “globalmente no caminho”, ou F , “futuramente no caminho”).

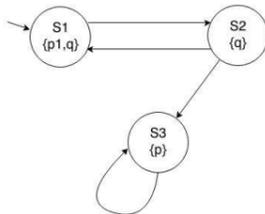
Modelos CTL

Para definir modelos de fórmulas CTL, precisamos da estrutura de Kripke.

Essa estrutura é, basicamente, constituída de um conjunto de estados, um conjunto de estados iniciais, uma relação entre os estados e uma associação entre cada estado e um conjunto de valores proposicionais.

Os modelos CTL são estruturas de Kripke com algumas relações definidas para formalizar quando uma fórmula é válida ou não em determinados estados.

Abaixo segue um exemplo de uma estrutura de Kripke:



Revisão de Modelos CTL

Neste trabalho, nos concentraremos em usar uma variante da revisão de crenças especializada em modelos CTL. Essa variante se faz necessária pois muitos dos resultados de revisão de crenças não se aplicam à lógica temporal.

Em um problema de revisão de crenças é comum usarmos algumas suposições para que a Teoria AGM seja válida, as chamadas suposições AGM. Mas, Hansson e Wasserman provam que apenas compacidade e monotonicidade já são suficientes para que as construções AGM comuns possam ser aplicadas.

Contudo, não podemos garantir que as construções AGM típicas possam ser aplicadas pois a lógica CTL não é compacta. Este é o motivo que levou à modificação das aborgens típicas e à criação dos algoritmos descritos na tese do Paulo. Estes são os algoritmos que implementaremos neste trabalho.

Bcontractor

Para implementar os algoritmos propostos, trabalharemos sobre o **framework** criado por Renato Lundberg, o **bcontractor** [2]. Este framework tem como objetivo dar uma forma prática de programar algoritmos de revisão em Java.

O **bcontractor** é uma plataforma genérica que nos dá uma ligação entre a parte matemática e a parte computacional da teoria. Isso é feito através de várias estruturas que definem entes como conjuntos, sentenças, provedores e outros.

Nele já existem algumas implementações de algoritmos feitas para testes do framework e que também nos serviram como base para o desenvolvimento dos nossos algoritmos. A saber, existem algumas implementações do **partial meet** e uma implementação de revisão para lógica descritiva (ou DL) feita através da OWL (**Web Ontology Language**).

Para o nosso trabalho, utilizaremos a estrutura de conjuntos e sentenças para definir a CTL e implementaremos um provedor e os algoritmos de interesse nos mesmos moldes dos já implementados.

Implementação

A ideia do algoritmo de revisão descrito em [1] é gerar modelos que satisfazem uma dada fórmula CTL e que sejam próximos a um determinado conjunto de modelos, ou seja, que possa ser construído a partir deste conjunto de modelos com poucas operações. Com isso, o algoritmo deve ser capaz de receber um conjunto de modelos inconsistentes e retornar um conjunto semelhante à entrada, porém consistente. Outro ponto a se ressaltar, é que este algoritmo de revisão segue os postulados de revisão AGM.

O algoritmo se baseia em chamadas recursivas de subfunções que tratam especificamente de cada tipo de operador temporal. Todas as escolhas que o algoritmo faz são determinísticas e todas as iterações são feitas sobre conjuntos finitos, como o conjunto de estados ou o conjunto de transições. Isso garante que o algoritmo eventualmente termine.

Além disso, as subfunções podem chamar a função principal novamente (de onde surge a recursividade) de forma a explorar os aninhamentos dos operadores temporais.

Referências

- [1] Paulo de Tarso Guerra Oliveira. Revisão de Modelos CTL. **São Paulo: Instituto de Matemática e Estatística, Universidade de São Paulo**, 2010. Dissertação de Mestrado em Ciência da Computação.
- [2] Renato Urquiza Lundberg. Análise Empírica de Algoritmos de Revisão de Crenças. **São Paulo: Instituto de Matemática e Estatística, Universidade de São Paulo**, 2013. Dissertação de Mestrado em Ciência da Computação.