



IME-USP

Implementação e análise de algoritmos para revisão de crença



Universidade de São Paulo

Aluno

Heitor Reis Ribeiro

Orientadora

Profa. Doutora Renata Wassermann

Teoria

Revisão de Crenças

A revisão de crenças é utilizada em atualizações de bancos de dados, diagnósticos, robótica, etc. Nestes campos, em frente a novas informações, é preciso mapear mudanças no estado epistêmico e, para isto, temos três operações: revisão (*), contração (÷) e expansão (+), relacionadas abaixo:

$$K * p = (K \div \neg p) + p$$

As construções mais conhecidas para contração foram testadas e comparadas.

Partial Meet

Construções Partial Meet usam a estrutura de remainder sets: Para um conjunto A e sentença p, $A \perp p$ é o conjunto dos subconjuntos maximais de A que não implicam p.

$$A \div p = \bigcap \gamma(A \perp p)$$

Kernel

A contração Kernel utiliza a estrutura de p-kernels: $A \perp\!\!\!\perp p$ são os subconjunto minimais de A que implicam p.

$$A \div p = A \setminus \sigma(A \perp\!\!\!\perp p)$$

Belief Change Extension

O algoritmo utiliza Belief Change Scenarios $B = (K, R, C)$, em que K é a base de crenças, R e C são conjuntos de sentenças em uma lógica com alfabeto P (Lp) e $P' = \{p' \mid p \in P\}$ um alfabeto disjunto.

Define-se um conjunto maximal de equivalências:

$$EQ \subseteq \{p \equiv p' \mid p \in P\}$$

$$Cn(K' \cup R \cup EQ) \cap (C \cup \{\perp\}) = \emptyset$$

gera $E = Cn(K' \cup R \cup EQ) \cap L_p$ a Belief Change Extension de B.

Algoritmos

Kernel

Kernel (B, p)

```

1 Cut ← queue ← ∅
2 S ← KernelBlackBox(B, a)
3 kernel ← {S}
4 for s ∈ S
5   queue.enqueue {s}
6 while queue is not empty
7   Hn ← queue.remove
8   continue if ∃C ∈ Cut | C ⊆ Hn
10  else if B \ Hn ⊢ p
11    S ← KernelBlackBox(B \ Hn, p)
12    kernel = kernel ∪ {S}
13  for s ∈ S
14    queue.enqueue Hn ∪ {s}
15  else
16    Cut ← Cut ∪ Hn
17  return kernel

```

Partial Meet

Partial Meet (B, p)

```

1 queue ← ∅
2 S ← RemBlackBox(B, p, ∅)
3 remainder ← {S}
4 for s ∈ B \ S
5   queue.enqueue {s}
6 while queue is not empty
7   Hn ← queue.remove
8   continue if Hn ⊢ p
9   S ← RemBlackBox(B, p, Hn)
10  remainder ← remainder ∪ {S}
11  for s ∈ B \ S
12    queue.enqueue Hn ∪ {s}
13  return remainder

```

Belief Change Extension

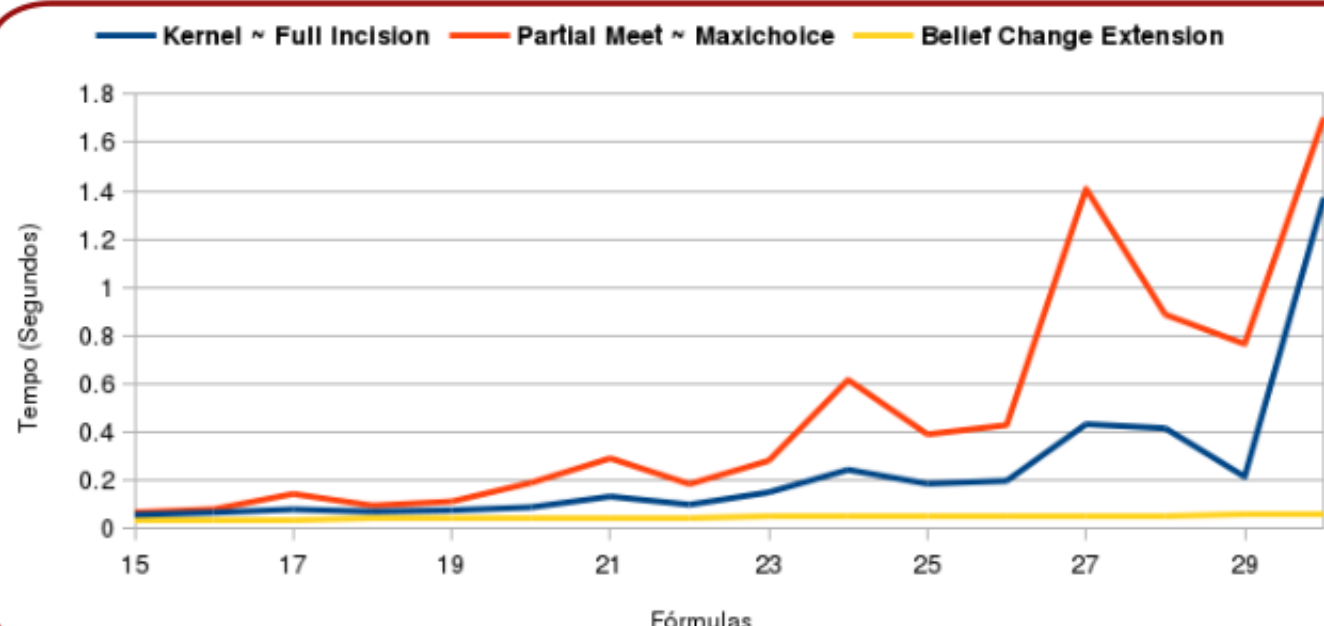
Belief Change (K, R, C)

```

1 if K ⊢ ⊥ or R ⊢ ⊥ then return ⊥
2 in ← out ← ∅
3 At ← Atoms(K) ∩
3.1 (Atoms(R) ∪ Atoms(C))
4 K' ← Prime(K, At)
5 for a ∈ At
6   if for b ∈ C ∪ {⊥}
6.1   we have K' ∪ R ∪
6.2   {p ≡ p' | p ∈ in ∪ {a}} ⊢ b
7     then in ← in ∪ {a}
8     else out ← in ∪ {a}
9 for p ∈ in
10  K' ← Replace(K', p', p)
11 for p ∈ out
12  K' ← Replace(K', p', ¬p)
13 return ((∧ a ∈ K' a) ∧ (∧ b ∈ R b))

```

Testes



Os testes foram feitos a partir de cenários gerados aleatoriamente consistindo de uma base de crenças B e uma fórmula f tal que $B \vdash f$, sendo que cada valor obtido é a média de 20 execuções de cada algoritmo.

Note que a Belief Change Extension é uma operação de revisão, portanto não respeitando as mesmas propriedades que as outras respeitam.

Trabalhos Referenciados

A Consistency-based approach for belief change (James P Delgrade, Torsten Schaub)
 Revisão de Crenças em Lógicas de Descrição - Um Plug-In para o Protégé (Fillipe Resina)
 Revisão de Crenças em Lógicas de Descrição e em outras lógicas não clássicas (Márcio Ribeiro)
 Análise Empírica de Algoritmos de Revisão sobre Bases de Crenças (Renato Lundberg)

Código disponível em:

https://bitbucket.org/marcheing/belief_change_rb