SCC-5774 - Capítulo 5
Prova Automática de Teoremas

João Luís Garcia Rosa[1]

[1]Departamento de Ciências de Computação
Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo - São Carlos

2020

Sumário

Sumário

Corpo de Conhecimento

Exemplo: Suponha o seguinte corpo de conhecimento:

1. Marco era um homem.
2. Marco era um pompeiano.
3. Todos os pompeianos eram romanos.
4. César era um soberano.
5. Todos os romanos ou eram leais a César ou o odiavam.
6. Todos são leais a alguém.
7. As pessoas somente tentam assassinar soberanos aos quais elas não são leais.
8. Marco tentou assassinar César.

Representação através de Fórmulas da LPPO

Representando este conhecimento através de fórmulas da lógica de primeira ordem:

1. $homem(marco)$
2. $pompeiano(marco)$
3. $\forall X(pompeiano(X) \rightarrow romano(X))$
4. $soberano(cesar)$
5. $\forall X(romano(X) \rightarrow (leal(X, cesar) \vee odiar(X, cesar)))$
6. $\forall X \exists Y leal(X, Y)$
7. $\forall X \forall Y((pessoa(X) \wedge tentarassassinar(X, Y) \wedge soberano(Y)) \rightarrow \neg leal(X, Y))$
8. $tentarassassinar(marco, cesar)$

Inclusão de Conhecimento de Senso Comum

- Suponha que se deseje usar este conhecimento para responder à questão "Marco era leal a César?"
- Parece que usando 7 e 8, dá para concluir que Marco não era leal a César (ignorando a distinção entre passado e presente).
- Há a necessidade de inclusão de conhecimento de senso comum:
  9. Todos os homens são pessoas.
  $\forall X(homem(X) \rightarrow pessoa(X))$

Conversão de Sentenças da LN para a LPPO

Deste exemplo simples, pode-se perceber três pontos importantes, na conversão de sentenças da língua natural (português) em fórmulas da lógica:

1. Muitas sentenças da língua natural são ambíguas. A escolha da interpretação correta pode ser difícil.
2. Existe uma escolha de como representar o conhecimento. Representações simples são desejáveis mas podem impedir certos tipos de raciocínio.
3. Mesmo em situações muito simples, um conjunto de sentenças pode não conter toda a informação necessária para raciocinar sobre o tópico em questão. Muitas vezes é necessário ter acesso a um outro conjunto de fórmulas que representam fatos considerados óbvios demais para mencionar (senso comum).

Quais Comandos Deduzir?

- Um outro problema surge em situações onde não se conhece de antemão quais comandos deduzir.
- No exemplo apresentado, o objetivo era responder a questão "Marco era leal a César?"
- Como um programa poderia decidir se deveria tentar provar

$$leal(marco, cesar)$$
ou
$$\neg leal(marco, cesar)$$

Sumário

O que é Resolução?

- O sistema formal da resolução trabalha exclusivamente com cláusulas e contém apenas uma regra de inferência, chamada de regra da resolução (RE):
  - RE gera uma nova cláusula a partir de duas outras.
- Dado um conjunto **S** de cláusulas e uma cláusula c:
  - Uma *dedução* de c a partir de **S** neste sistema formal consiste de uma seqüência de cláusulas terminando em c e gerada aplicando-se repetidamente a regra da resolução.
  - Uma *refutação* a partir de **S** é uma dedução da cláusula vazia a partir de **S**.
- A regra da resolução é definida de tal forma que **S** é insatisfazível se e somente se existe uma refutação a partir de **S**.

O que é Resolução?

- Na definição da regra da resolução, tratar-se-á uma cláusula não-vazia "$l_1 ... l_n$" como o conjunto finito $\{l_1, ..., l_n\}$ e a cláusula vazia "$\square$" como o conjunto vazio.
- Assim, utilizar-se-á as operações usuais de teoria dos conjuntos para definir novas cláusulas a partir de outras.
    - Por exemplo, se "$l\ m\ n$" e "$n\ p$" são cláusulas, a expressão "$(l\ m\ n) \cup (n\ p)$" denota a cláusula "$l\ m\ n\ p$" (a ordem dos literais no resultado é irrelevante em face da semântica das cláusulas).

Instanciação

- Uma cláusula A é uma *instância* de B se e somente se existir uma substituição $\beta = \{X_1/t_1, ..., X_n/t_n\}$ de variáveis por termos tal que A é obtida substituindo-se simultaneamente X_i por t_i em B, para $i = 1, ..., n$. Usar-se-á $B\beta$ para denotar o resultado da substituição.
  - Exemplo: Seja a cláusula $B = p(X)\ q(X, Y)$. Seja uma substituição $\beta = \{X/a, Y/f(b)\}$. A instanciação de B por β, denotada por $B\beta$, é a cláusula instância $A = p(a)\ q(a, f(b))$.
- A *regra da resolução* combina:
  - uma adaptação para cláusulas da regra *Modus Ponens* (regra R1)
  - um processo de "unificação" de literais de duas cláusulas (regra R2)
  - um processo de "unificação" de literais de uma mesma cláusula (regra da resolução RE).

Regra R1

Regra R1

se A' possui um literal l e A'' possui um literal $\neg l$, derive $A = (A' - l) \cup (A'' - \neg l)$.

- Exemplo 1: Seja o seguinte conjunto de cláusulas:
  1. $p(X)\ \neg q(Y)$
  2. $q(Y)\ r(Z)$

  dá para obter, usando a regra R1, a cláusula

  3. $p(X)\ r(Z)$
- Exemplo 2: Seja agora o seguinte conjunto de cláusulas:
  1. $p(X)\ \neg q(Y)$
  2. $q(W)\ r(Z)$

  não é possível mais obter a cláusula 3, pois as variáveis são diferentes. A regra R2 vai resolver este problema.

Regra R2

Regra R2

se A' possui um literal l' e A'' possui um literal $\neg l''$ e existe uma substituição β tal que $l'\beta = l''\beta$, derive
$A = (A'\beta - l'\beta) \cup (A''\beta - \neg l''\beta)$

- Exemplo: Retomando o conjunto anterior
  1. $p(X)\ \neg q(Y)$
  2. $q(W)\ r(Z)$

  existe uma substituição $\beta = \{W/Y\}$, que aplicada às duas cláusulas, resulta no seguinte
  1. $p(X)\ \neg q(Y)$
  2. $q(Y)\ r(Z)$

  que obviamente produz a cláusula abaixo
  3. $p(X)\ r(Z)$

u.m.g. e Fatoração

- O processo de tornar idênticos os literais em uma cláusula c através de uma substituição de variáveis por termos é chamado de *unificação* e a substituição é chamada de um *unificador* de c.
- Um *unificador mais geral* (*u.m.g.*) é aquele que, intuitivamente, especifica as substituições mais simples possíveis.
- O processo de unificação deverá então utilizar sempre um unificador mais geral para não bloquear outras unificações.
- Diz-se que uma cláusula B é um *fator* de uma cláusula A se e somente se existe um conjunto $\mathbf{L}$ de literais de A e existe um unificador mais geral φ para $\mathbf{L}$ tal que $B = A\varphi$. Note que uma cláusula A é um fator dela mesma.
- O processo de obter fatores de cláusulas é chamado de *fatoração*.

Regra da Resolução

Regra RE

se B' e B'' são fatores de cláusulas A' e A'' tais que B' possui um literal l' e B'' um literal $\neg l''$ e existe um unificador mais geral β para l' e l'', derive $A = (B'\beta - l'\beta) \cup (B''\beta - \neg l''\beta)$

Neste caso, diz-se que a cláusula A é um *resolvente* de A' e A'', que são as cláusulas *pais*.

Regra da Resolução

- Exemplo: Seja o seguinte conjunto de cláusulas:

1. $p(X)\ q(Z)$
2. $\neg r(Y)\ p(T)\ \neg r(W)$
3. $r(V)\ \neg q(U)$

4. $p(T)\ \neg r(W)\ \neg q(U)$ R2: 2,3 $\beta = \{V/Y\}$
5. $p(X)\ p(T)\ \neg r(W)$ R2: 1,4 $\beta = \{Z/U\}$

2'. $\neg r(Y)\ p(T)$ fator de 2, com $\varphi = \{W/Y\}$

4. $p(T)\ \neg q(U)$ RE: 2',3 $\beta = \{Y/V\}$
5. $p(X)$ RE: 1, 4 $\beta = \{U/Z\}$ e
fator de $p(X)\ p(T)$; $\varphi = \{T/X\}$

O Sistema Formal da Resolução

- O sistema formal da resolução, RE, consiste de:
  - *Classe de Linguagens*: linguagens de cláusulas
  - *Axiomas*: nenhum
  - *Regra de Inferência Lógica*: Regra da Resolução (RE)

Regra RE

se B' e B'' são fatores de cláusulas A' e A'' tais que B' possui um literal l' e B'' um literal $\neg l''$ e existe um unificador mais geral β para l' e l'', derive $A = (B'\beta - l'\beta) \cup (B''\beta - \neg l''\beta)$

Dedução e Refutação

- Seja **S** um conjunto de cláusulas e c uma cláusula.
  - Uma *dedução* de c a partir de **S** no sistema formal da resolução ou, simplesmente, uma *R-dedução* de c a partir de **S**, é uma seqüência $\mathbf{D} = (D_1, ..., D_n)$ de cláusulas tal que:
    - $D_n = c$
    - para todo $i \in [1, n]$, D_i pertence a **S** ou D_i é um resolvente de D_j e D_k, para algum j, $k < i$.

    Para cada $i \in [1, n]$, D_i é uma *cláusula de entrada* em **D** se e somente se D_i pertence a **S**; caso contrário, D_i é uma *cláusula derivada*.
  - Uma *refutação* a partir de **S** no sistema formal da resolução ou, simplesmente, uma *R-refutação* a partir de **S**, é uma R-dedução de $\Box$ a partir de **S**.

Sumário

Refutação

- No problema da prova de teorema tem-se um conjunto de fórmulas **S**, a partir do qual deseja-se provar alguma fórmula meta, f.
- Em uma refutação por resolução, primeiro nega-se a fórmula meta, e então adiciona-se a negação ao conjunto **S**.
- Este conjunto expandido é então convertido a um conjunto de cláusulas, e usa-se resolução para derivar uma contradição, representada pela cláusula vazia, $\Box$.

Refutação

- Um simples argumento pode ser dado para justificar o processo de prova por refutação.
  - Suponha uma fórmula f, que segue logicamente de um conjunto de fórmulas **S**; então, por definição, nenhuma interpretação que satisfaz **S** pode satisfazer $\neg f$, e, portanto, nenhuma interpretação pode satisfazer a união de **S** e $\{\neg f\}$.
  - Portanto, se f segue logicamente de **S**, o conjunto **S** $\cup\{\neg f\}$ é insatisfazível.

Refutação

- Se a resolução é aplicada repetidamente a um conjunto de cláusulas insatisfazíveis, em algum momento a cláusula vazia, $\Box$, será produzida.
- Portanto, se f segue logicamente de **S**, então a resolução em algum momento produzirá a cláusula vazia a partir da representação de cláusulas do conjunto $\mathbf{S} \cup \{\neg f\}$.
- Por outro lado, se a cláusula vazia é produzida, a partir da representação de cláusulas $\mathbf{S} \cup \{\neg f\}$, então f segue logicamente de **S**.

Refutação

- Considere um exemplo simples. Observe as seguintes frases:
  - (1) Qualquer um que possa ler é alfabetizado.
    fórmula: $\forall X(l(X) \rightarrow a(X))$
  - (2) Os golfinhos não são alfabetizados.
    fórmula: $\forall X(g(X) \rightarrow \neg a(X))$
  - (3) Alguns golfinhos são inteligentes.
    fórmula: $\exists X(g(X) \wedge i(X))$
- A partir destes quer-se provar a frase:
  - (4) Alguns que são inteligentes não podem ler.
    fórmula: $\exists X(i(X) \wedge \neg l(X))$

Refutação

- O conjunto de cláusulas que correspondem às frases 1 a 3 é:
  - 1. $\neg l(X)\ a(X)$
  - 2. $\neg g(Y)\ \neg a(Y)$
  - 3a. $g(a)$
  - 3b. $i(a)$

  onde a é a constante de Skolem. A negação do teorema a ser provado, convertido à forma de cláusula, é:
  - 4'. $\neg i(Z)\ l(Z)$

Refutação

- Provar este teorema através da refutação por resolução envolve gerar resolventes a partir do conjunto de cláusulas 1-3 e 4', adicionando estes resolventes ao conjunto, e continuando até que a cláusula vazia seja produzida. Uma prova possível (existe mais de uma) produz a seguinte seqüência de resolventes:
  - 5. $l(a)$ resolvente de $3b$ e $4'$, $\beta = \{Z/a\}$
  - 6. $a(a)$ resolvente de 5 e 1, $\beta = \{X/a\}$
  - 7. $\neg g(a)$ resolvente de 6 e 2, $\beta = \{Y/a\}$
  - 8. $\square$ resolvente de 7 e $3a$, ϵ.

Procedimento Resolução

- Procedimento RESOLUÇÃO
  1. CLÁUSULAS $\leftarrow$ **S**
  2. até que $\Box$ seja um membro de CLÁUSULAS, faça:
     1. selecione duas cláusulas distintas c_i e c_j em CLÁUSULAS
     2. calcule um resolvente, r_{ij}, de c_i e c_j
     3. CLÁUSULAS $\leftarrow$ o conjunto produzido adicionando r_{ij} a CLÁUSULAS
- Observe que o procedimento Resolução acima é muito similar ao procedimento *Produção* do capítulo 1.

Procedimento Resolução

- As decisões sobre quais cláusulas em CLÁUSULAS resolver (comando 1 do *loop*) e qual resolução destas cláusulas realizar (comando 2 do *loop*) são tomadas através da estratégia de controle.
- É útil para a estratégia de controle usar uma estrutura chamada de grafo de derivação.
- Os nós neste grafo são rotulados pelas cláusulas; inicialmente, existe um nó para toda cláusula no conjunto base.
- Quando duas cláusulas c_i e c_j produzem um resolvente r_{ij}, cria-se um novo nó, descendente, rotulado r_{ij}, ligado com os nós pais c_i e c_j.

Procedimento Resolução

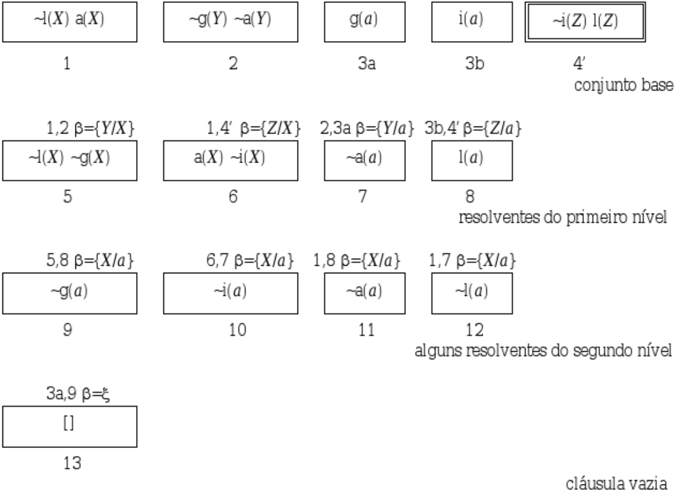
- Uma refutação por resolução pode ser representada como uma árvore de refutação (dentro do grafo de derivação) tendo um nó folha rotulado por $\square$.
- A estratégia de controle busca por uma refutação crescendo o grafo de derivação até que uma árvore seja produzida com um nó folha rotulado pela cláusula vazia, $\square$.
- Uma estratégia de controle para um sistema de refutação é completa se seu uso resulta num procedimento que achará uma contradição (eventualmente) onde existir.

Sumário

Busca em Largura

- Na estratégia de *busca em largura*, todos os resolventes de primeiro nível são calculados primeiro, depois os resolventes de segundo nível, e assim por diante.
- Um resolvente de primeiro nível está entre as cláusulas do conjunto base; um resolvente do i-ésimo nível é aquele cujos pais são resolventes do (i - 1)-ésimo nível.
- A estratégia de busca em largura é completa, mas é muito ineficiente.
- Exemplo: Exemplo do golfinho:
  - 1. $\neg l(X)\ a(X)$
  - 2. $\neg g(Y)\ \neg a(Y)$
  - 3a. $g(a)$
  - 3b. $i(a)$
  - 4'. $\neg i(Z)\ l(Z)$

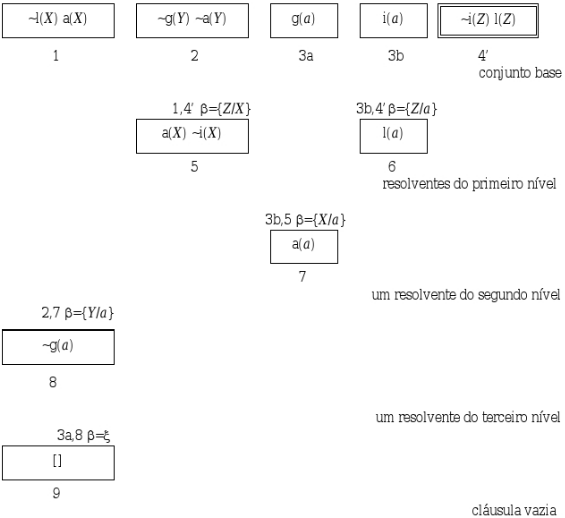
Busca em Largura



Conjunto de Suporte

- Uma refutação por conjunto de suporte é aquela na qual no mínimo um pai para cada resolvente é selecionado entre as cláusulas resultantes da negação da fórmula meta ou dos seus descendentes (o conjunto de suporte).
- A estratégia precisa garantir a busca de todos as refutações por conjunto de suporte possíveis (na forma por largura).
- Além de completa, a estratégia do conjunto de suporte é mais eficiente que a busca em largura.

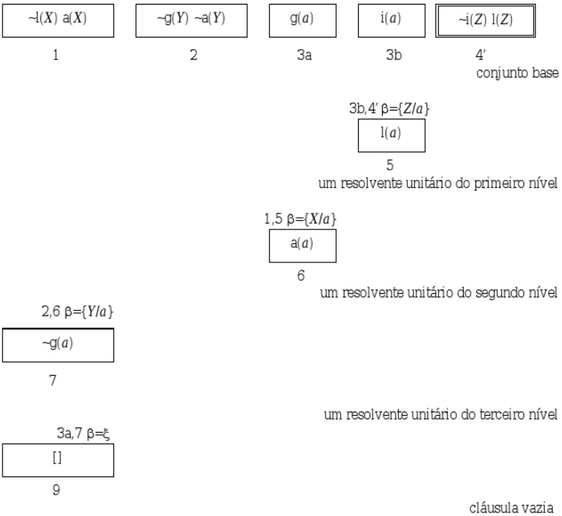
Conjunto de Suporte



Preferência Unitária

- A estratégia por preferência unitária é uma modificação da estratégia por conjunto de suporte na qual, ao invés de preencher cada nível na forma por largura, tenta-se selecionar uma cláusula de um único literal (chamado de unidade) para ser um pai numa resolução.
- Cada vez que as unidades são usadas na resolução, os resolventes têm menos literais do que seus outros pais.
- Este processo ajuda a dirigir a busca para produzir a cláusula vazia e, então, tipicamente, aumentar a eficiência. Mas não é completa.

Preferência Unitária

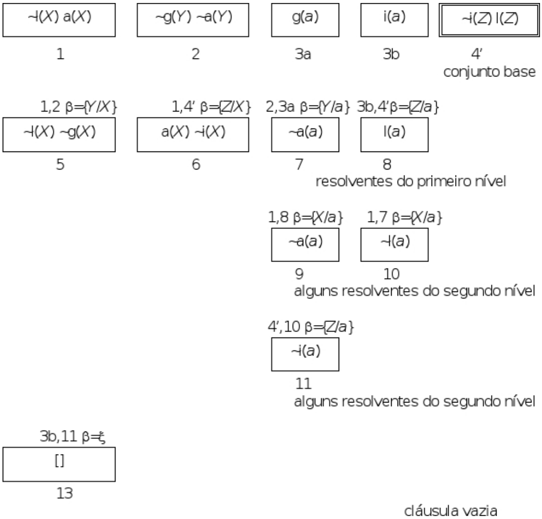


Sumário

Forma de entrada linear

- Uma refutação por forma de entrada linear é aquela na qual cada resolvente tem no mínimo um pai pertencente ao conjunto base.
- Esta estratégia não é completa, ou seja, existem casos nos quais uma refutação existe mas uma refutação por forma de entrada linear não.

Forma de entrada linear

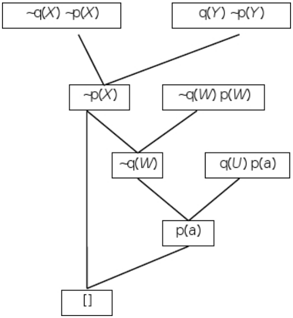


Forma "ancestral filtrada"

- Uma refutação por forma "ancestral filtrada" é aquela onde cada resolvente tem um pai que está no conjunto base ou que é um ancestral do outro pai.
- Portanto, a forma "ancestral filtrada" é muito parecida com a forma linear. É uma estratégia completa.
- Exemplo: Conjunto Base **S**:
  1. $\neg q(X)\ \neg p(X)$
  2. $q(Y)\ \neg p(Y)$
  3. $\neg q(W)\ p(W)$
  4. $q(u)\ p(a)$

Forma “ancestral filtrada”

- Obs.: Para a estratégia forma de entrada linear, não se chega à cláusula vazia, pois o conjunto **S** deve ter pelo menos uma cláusula unitária. A árvore de refutação abaixo está simplificada, ou seja, não estão explícitas todas as derivações possíveis.



Simplificação

- Algumas vezes um conjunto de cláusulas pode ser simplificado pela eliminação de certas cláusulas ou pela eliminação de certos literais dentro das cláusulas.
- Estas simplificações são tais que o conjunto de cláusulas simplificado é insatisfazível se e somente se o conjunto original for insatisfazível.
- Portanto, o emprego destas estratégias de simplificação ajuda a reduzir a taxa de crescimento de novas cláusulas.

Sumário

Eliminação de Tautologias

- Qualquer cláusula contendo um literal e sua negação (chama-se tal cláusula uma tautologia) pode ser eliminada, desde que qualquer conjunto insatisfazível contendo uma tautologia ainda seja insatisfazível depois de sua remoção e vice-versa.

Sumário

Incorporação Procedimental

- Algumas vezes é possível e mais conveniente calcular os valores verdade de literais (instâncias concretas de predicados computáveis) do que incluir estes literais, ou suas negações, no conjunto base.

Sumário

Eliminação por subjugação

- Por definição, uma cláusula A subjuga uma cláusula B se existe uma substituição β tal que $A\beta$ é um subconjunto de B. Como exemplos:
  - $p(X)$ subjuga $p(Y)\ q(Z)$, para $\beta = X/Y$
  - $p(X)$ subjuga $p(a)$, para $\beta = X/a$
  - $p(X)$ subjuga $p(a)\ q(Z)$, para $\beta = X/a$
  - $p(X)\ q(a)$ subjuga $p(f(a))\ q(a)\ r(Y)$, para $\beta = X/f(a)$
- Uma cláusula num conjunto insatisfazível que é subjugada por uma outra cláusula no conjunto pode ser eliminada sem afetar a insatisfazibilidade do resto do conjunto.
- A eliminação de cláusulas subjugadas por outras freqüentemente leva a reduções substanciais no número de resoluções necessárias para encontrar uma refutação.

Referências I

[1] Rosa, J. L. G.
Fundamentos da Inteligência Artificial.
Editora LTC. Rio de Janeiro, 2011.

[2] Casanova, M. A., Giorno, F. A. C., Furtado, A. L.
Programação em Lógica e a Linguagem Prolog.
Ed. Edgard Blücher Ltda., 1987

[3] Nilsson, N. J.
Principles of Artificial Intelligence.
Springer-Verlag; 1982.