Inteligência Artificial
IA

Prof. João Luís Garcia Rosa

III. PROVA AUTOMÁTICA DE TEOREMAS

Parte 2

2004

Cláusulas e resolventes

Cláusulas pais	Resolventes(s)	Comentários
p e ¬p q (isto é, p → q)	q	Modus Ponens
p q e ¬p q	q	A cláusula q q se reduz a q.
p q e ¬p ¬q	q ¬q ou p ¬p	Aqui, existem dois resolventes possíveis: ambos tautologias
¬p e p		Esta cláusula vazia é sinal de contradição
¬p q (ou p → q) e ¬q r (ou q → r)	¬p r (ou p → r)	Encadeamento

- Uma *renomeação* para B em presença de A é uma renomeação de variáveis β tal que A e $B\beta$ não possuem variáveis em comum. Recorde que se l é um literal da forma p ou da forma $\neg p$, então p é o átomo de l, denotado por $|l|$.
- Uma cláusula A é um *fator* de uma cláusula A' se e somente se existe um conjunto **L'** de literais de A' e um u.m.g. β de **L'** tais que $A = A'\beta$.

- Uma cláusula A é um *resolvente binário* de cláusulas A' e A" se e somente se existem literais l' e l" de A' e A", respectivamente, e uma substituição θ tais que
  - l' e l" têm sinais opostos e θ é um u.m.g. de $\{|l'|,|l''|\}$
  - $A = (A'\theta - l'\theta) \cup (A''\theta - l''\theta)$
- Diz-se ainda que l' e l" são os *literais resolvidos* e que θ é a *substituição de resolução*.

- Portanto, por convenção, o resolvente A é formado:
  - eliminando $l'\theta$ de $A'\theta$ e $l''\theta$ de $A''\theta$;
  - listando os literais restantes de $A'\theta$ e $A''\theta$ em qualquer ordem;
  - eliminando os literais repetidos.
- Sejam A' e A'' cláusulas e β uma renomeação para A'' em presença de A'. Uma cláusula A é um *resolvente* de A' e A'' se e somente se A é um resolvente binário de fatores de A' e $A''\beta$.

- O *sistema formal da resolução, RE*, consiste de:
  - *Classe de Linguagens*: linguagens de cláusulas
  - *Axiomas*: nenhum
  - *Regra de Inferência*: Regra da Resolução (RE)
    - *RE*: se A' e A'' são cláusulas e A é um resolvente de A' e A'', então derive A de A' e A''.

- 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 **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 a partir de **S**.

Refutação por Resolução

- No problema da prova de teorema tem-se um conjunto de fórmulas **S**, a partir do qual deseja-se provar alguma fórmula meta, W. Os sistemas baseados em resolução produzem provas por contradição, ou *refutação*. 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, .
- Um simples argumento pode ser dado para justificar o processo de prova por refutação. Suponha uma fórmula W, que segue logicamente de um conjunto de fórmulas **S**; então, por definição, nenhuma interpretação que satisfaz **S** pode satisfazer ¬W, e, portanto, nenhuma interpretação pode satisfazer a união de **S** e {¬W}. Portanto, se W segue logicamente de **S**, o conjunto **S** ∪ {¬W} é insatisfatível.

Refutação por Resolução

- Se a resolução é aplicada repetidamente a um conjunto de cláusulas insatisfatíveis, em algum momento a cláusula vazia, , será produzida. Portanto, se W segue logicamente de **S**, então a resolução em algum momento produzirá a cláusula vazia a partir da representação da cláusula **S** ∪ {¬W}. Por outro lado, se a cláusula vazia é produzida a partir da representação de cláusula **S** ∪ {¬W}, então W segue logicamente de **S**.

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

- 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 a forma de cláusula, é:
  - (4') $\neg i(Z)\ l(Z)$.

Refutação por Resolução

- 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) I(a)	resolvente de 3b e 4'
(6) a(a)	resolvente de 5 e 1
(7) ¬g(a)	resolvente de 6 e 2
(8)	resolvente de 7 e 3a.

Refutação por Resolução

- Procedimento *RESOLUÇÃO*

1 *CLÁUSULAS* ⇐ **S**
2 *até que* □ seja um membro de *CLÁUSULAS*, *faça*:
3 *início*
 4 selecione duas cláusulas distintas C*i* e C*j* em *CLÁUSULAS*
 5 calcule um resolvente, R*ij* de C*i* e C*j*
 6 CLÁUSULAS ⇐ o conjunto produzido adicionando R*ij* a *CLÁUSULAS*
7 *fim*

- Observe que o procedimento Resolução acima é muito similar ao procedimento Produção do capítulo 1.

Estratégias de Controle para Métodos de Resolução

- As decisões sobre quais cláusulas em *CLÁUSULAS* resolver (comando 4) e qual resolução destas cláusulas realizar (comando 5) 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.

Estratégias de Controle para Métodos de Resolução

- Uma refutação por resolução pode ser representada como uma *árvore de refutação* (dentro do grafo de derivação) tendo um nó folha rotulado por .
- 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, . Uma estratégia de controle para um sistema de refutação é *completa* se seu uso resulta num procedimento que achará uma contradição (eventualmente) onde existir. (A completude de uma *estratégia* não deve ser confundida com a completude lógica de uma regra de inferência).

Estratégias para Provar uma Meta através da Refutação

- *a) A ESTRATÉGIA DE 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. I(a) : corresponde à cláusula 3b
  2. ¬I(*Z*) r(*Z*) : corresponde à negação da meta (4´)
  3. ¬r(*X*) L(*X*) : corresponde à cláusula 1
  4. ¬D(*Y*) ¬L(*Y*) : corresponde à cláusula 2
  5. D(a) : corresponde à cláusula 3a

Estratégias para Provar uma Meta através da Refutação

i(a) | ¬i(*Z*) r(*Z*) | ¬r(*X*) l(*X*) | ¬d(*Y*) ¬l(*Y*) | d(a)

---

r(a) | ¬i(*Z*) l(*Z*) | ¬r(*Y*) ¬d(*Y*) | ¬l(a)

---

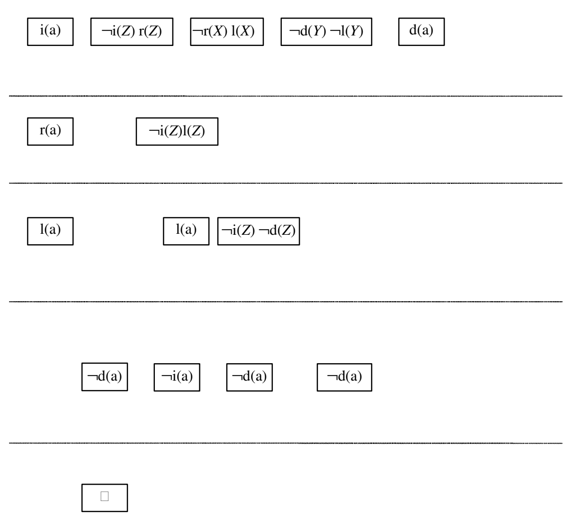
l(a) | ¬d(a) | l(a) | ¬i(*Z*) ¬d(*Z*) | ¬i(*Z*) ¬d(*Z*) | ¬r(a)

□

Estratégias para Provar uma Meta através da Refutação

- *b) A ESTRATÉGIA DO 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.

Estratégias para Provar uma Meta através da Refutação



Estratégias para Provar uma Meta através da Refutação

- *c) A ESTRATÉGIA POR 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.

Estratégias para Provar uma Meta através da Refutação

i(a) | ¬i(Z) r(Z) | ¬r(X) l(X) | ¬d(Y) ¬l(Y) | d(a)

---

r(a)

---

l(a)

¬d(a)

□

Estratégias para Provar que um Conjunto de Cláusulas é Insatisfatível

- *d) A ESTRATÉGIA POR 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.

Estratégias para Provar que um Conjunto de Cláusulas é Insatisfatível

$\boxed{i(a)}$ $\boxed{\neg i(Z)\ r(Z)}$ $\boxed{\neg r(X)\ l(X)}$ $\boxed{\neg d(Y)\ \neg l(Y)}$ $\boxed{d(a)}$

$\boxed{r(a)}$ $\boxed{\neg i(Z)\ l(Z)}$ $\boxed{\neg r(Y)\ \neg d(Y)}$ $\boxed{\neg l(a)}$

$\boxed{l(a)}$ $\boxed{l(a)}$ $\boxed{\neg i(Z)\ \neg d(Z)}$ $\boxed{\neg i(Y)\ \neg d(Y)}$ $\boxed{\neg r(a)}$

$\boxed{\neg i(a)}$

$\boxed{\square}$

Estratégias para Provar que um Conjunto de Cláusulas é Insatisfatível

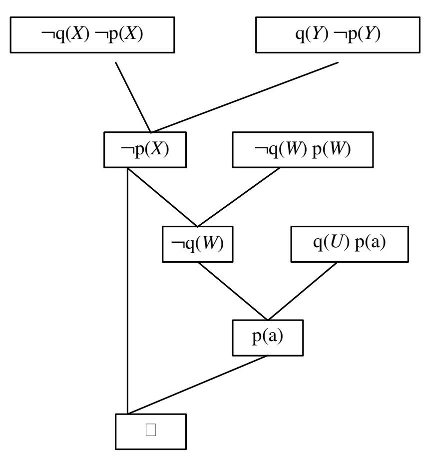
- *e) A ESTRATÉGIA POR FORMA "ANCESTRAL FILTRADA"*
  - Uma refutação por forma "ancestral filtrada" é aquela onde cada resolvente tem um pai que está no conjunto base ou que é um ancestral do outro pai. Portanto, a forma "ancestral filtrada" é muito parecida com a forma linear. É uma estratégia completa.

  Exemplo: Conjunto Base **S**:

  1. ¬q(*X*) ¬p(*X*)
  2. q(*Y*) ¬p(*Y*)
  3. ¬q(*W*) p(*W*)
  4. q(*u*) p(a)

  - 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 explícitos todas as derivações possíveis.

Estratégias para Provar que um Conjunto de Cláusulas é Insatisfatível



Estratégias de 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 insatisfatível se e somente se o conjunto original for insatisfatível. Portanto, o emprego destas estratégias de simplificação ajuda a reduzir a taxa de crescimento de novas cláusulas.

Estratégias de Simplificação

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

- *b) INCORPORAÇÃO PROCEDIMENTAL*
  - Algumas vezes é possível e mais conveniente *calcular* os valores verdade de literais do que incluir estes literais, ou suas negações, no conjunto base. Tipicamente, os cálculos são realizados para instâncias concretas. Uma *instância concreta* é uma instância de uma expressão onde não ocorrem variáveis.
  - Mas o que significa realmente "calcular" uma expressão. As expressões do cálculo de predicados são construções lingüísticas que denotam valores verdade, elementos, funções ou relações num domínio. Tais expressões podem ser interpretadas com referência a um modelo que associa entidades lingüísticas com entidades de domínio apropriadas. O resultado final é que os valores V ou F tornam-se associados com sentenças na linguagem.

- *c) ELIMINAÇÃO POR SUBJUGAÇÃO*
  - Por definição, uma cláusula A*i* *subjuga* uma cláusula B*i* se existe uma substituição β tal que A*i* β é um subconjunto de B*i* . Como exemplos:
    - p(*X*) subjuga p(*Y*) q(*Z*), para β = { *X* / *Y* }
    - p(*X*) subjuga p(a), para β = { *X* / a }
    - p(*X*) subjuga p(a) q(*Z*), para β = { *X* / a }
    - p(*X*) q(a) subjuga p(f(a)) q(a) r(*Y*), para β = {*X* / f(a)}
  - Uma cláusula num conjunto insatisfatível que é subjugada por uma outra cláusula no conjunto pode ser eliminada sem afetar a insatisfatibilidade 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.

Exemplo 1

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.
9. Todos os homens são pessoas.

Exemplo 1

- Fórmulas da lógica de primeira ordem:
  1. *homem(marco)*
  2. *pompeiano(marco)*
  3. $\forall X\ (pompeiano(X) \rightarrow romano(X))$
  4. *soberano(césar)*
  5. $\forall X\ (romano(X) \rightarrow (leal(X,césar) \vee odiar(X,césar)))$
  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,césar)*
  9. $\forall X\ (homem(X) \rightarrow pessoa(X))$

Exemplo 1

- Cláusulas:
  1. *homem(marco)*
  2. *pompeiano(marco)*
  3. ¬ *pompeiano(X) romano(X))*
  4. *soberano(césar)*
  5. ¬ *romano(X) leal(X,césar) odiar(X,césar)*
  6. *leal(X,f(Y))*
  7. ¬ *pessoa(X)* ¬ *tentarassassinar(X,Y)*
     ¬ *soberano(Y))* ¬ *leal(X,Y))*
  8. *tentarassassinar(marco,césar)*
  9. ¬ *homem(X) pessoa(X)*

Exemplo 1

- Meta:
  "Marco não era leal a César"
- Cláusula:
  ¬ leal(marco, césar)
- Nega-se a meta:
  leal(marco, césar)
- Inclui-se a meta negada no conjunto de cláusulas:
  10. leal(marco, césar)