INTRODUÇÃO À INTELIGÊNCIA ARTIFICIAL

PARTE 3. PROVA AUTOMÁTICA DE TEOREMAS

3.1. RESOLUÇÃO

3.1.1. 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, que 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 insatisfatível se e somente se existe uma refutação a partir de **S**.

Convém recordar alguns pontos antes de prosseguir. 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 “ ” 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).

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 Bβ para denotar o resultado da substituição.

<u>Exemplo</u>: 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 Bβ, 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).

Para o primeiro passo recorde que a regra Modus Ponens ditava que de P e de P $\rightarrow$ Q, pode-se derivar Q ou, equivalentemente, de P e de $\neg$P $\vee$ Q pode-se derivar Q. Uma adaptação imediata de Modus Ponens para cláusulas seria então:

*Regra R1***:** se A' possui um literal L e A" possui um literal $\neg$L, derive A = (A' - L) $\cup$ (A" - $\neg$L).

Ou seja, A é uma lista de literais contendo, em qualquer ordem, os literais em A' e A", exceto L e ¬L.

<u>Exemplo</u>: Seja o seguinte conjunto de cláusulas:

1. P(*x*) ¬Q(*y*)
2. Q(*y*) R(*z*)

dá para obter, usando a regra R1, a cláusula

3. P(*x*) R(*z*)

Agora, imagine o seguinte conjunto de cláusulas, parecido com o anterior:

1. P(*x*) ¬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***:** se A' possui um literal L' e A" possui um literal ¬L" e existe uma substituição β tal que L'β = L"β, derive A = (A'β - L'β) ∪ (A"β - ¬L"β)

Ou seja, A é uma lista de literais formada tomando-se, em qualquer ordem, os literais em A' e A", exceto L' e ¬L", e aplicando-se a substituição β a todos os literais restantes.

<u>Exemplo</u>: Retomando o conjunto acima

1. P(*x*) ¬Q(*y*)
2. Q(*w*) R(*z*)

existe uma substituição β = { *w*/*y* }, que aplicada às duas cláusulas, resulta no seguinte

1. P(*x*) ¬Q(*y*)
2. Q(*y*) R(*z*)

que obviamente produz a cláusula abaixo

3. P(*x*) R(*z*)

O processo de tornar idênticos os literais em um conjunto **E** através de uma substituição de variáveis por termos é chamado de *unificação* e a substituição é chamada de um *unificador* de **E**. Um *unificador mais geral* 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. Por exemplo, β = {*x*/f(a),*y*/a} e θ = {*x*/f(*y*)} unificam **E** = {P(*x*), P(f(y))} pois **E**β = {P(f(a))} e **E**θ = {P(f(y))}. Porém θ é um unificador mais geral do que β. A substituição θ é então preferível a β pois, por exemplo, o literal "P(f(b))" é unificável com o literal em **E**θ, mas não é unificável com o literal em **E**β.

Em geral, diz-se que uma cláusula B é um *fator* de uma cláusula A se e somente se existe um conjunto **L** de literais de A e existe um unificador mais geral β para **L** tal que B = Aβ. Note que uma cláusula A é um fator dela mesma. O processo de obter fatores de cláusulas é chamado de *fatoração*.

Regra RE: se B' e B" são fatores de cláusulas A' e A" tais que B' possui um literal L' e B" um literal ¬L" e existe um unificador mais geral β para L' e L", derive A = (B'β - L'β) ∪ (B"β - ¬L"β)

Neste caso, diz-se que a cláusula A é um *resolvente* de A' e A", que são as *cláusulas pais*. A tabela 3.1 mostra alguns casos interessantes de resolução.

Exemplo: Seja o seguinte conjunto de cláusulas:

1. P(x) Q(z)
2. ¬R(y) P(t) ¬R(w)
3. R(v) ¬Q(u)

Se usarmos a regra R1, não obteremos nada, pois esta regra não permite substituições. Se usarmos a regra R2, obteremos as seguintes cláusulas:

4. P(t) ¬R(w) ¬Q(u)	R2: 2,3 β = {v/y}
5. P(x) P(t) ¬R(w)	R2: 1,4 β = {z/u}

e assim por diante. Mas se usarmos a Regra da Resolução (RE), podemos simplificar a cláusula 2, fatorando-a:

2'. ¬R(y) P(t)	fator de 2, com β = {w/y}

e aí podemos continuar aplicando RE:

4. P(t) ¬Q(u)	RE: 2',3 β = {y/v}
5. P(x)	RE: 1, 4 β = {u/z} e fator de P(x) P(t)

Em resumo, o sistema formal de resolução trabalha apenas com cláusulas, que são objetos com uma sintaxe bem simples, e possui apenas uma regra de inferência, a regra da resolução. Um procedimento de *refutação baseado em resolução* é então um procedimento que, dado um conjunto qualquer de cláusulas **S**, procura sistematicamente derivar a cláusula vazia utilizando apenas a regra da resolução e tendo como ponto de partida as cláusulas em **S**. A construção de tais procedimentos requer, porém, métodos especiais para tentar contornar a explosão combinatorial gerada pela liberdade de escolha de cláusulas, fatores e literais.

Tabela 3.1 - 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

3.1.2. Unificação

Definição 3.1: (a) Um par (x,t) é uma *substituição simples* (lê-se "x substituído por t") se e somente se x é uma variável e t é um termo.

(b) Um conjunto finito β de substituições simples é uma *substituição* se e somente se duas substituições simples em β não coincidem no primeiro elemento.

(c) Uma substituição β é uma *substituição básica* se e somente se, para todo (x,t) em β, t é um termo sem ocorrências de variáveis.

(d) β é a *substituição vazia* se e somente se β for o conjunto vazio.

(e) Uma substituição β é uma *renomeação de variáveis* ou, simplesmente, uma *renomeação*, se e somente se cada par (x,t) em β for tal que t é uma variável e não existirem dois pares (x,u) e (y,v) em β tais que $x \neq y$ e $u = v$.

A expressão "x/t" denotará uma substituição simples (x,t). Letras gregas denotarão substituições e, em especial, "ε" denotará a substituição vazia.

Uma *expressão* é qualquer seqüência de símbolos de um alfabeto de primeira ordem, e uma *expressão simples* é qualquer literal ou termo sobre o alfabeto.

Definição 3.2: (a) Sejam E uma expressão e $\beta = \{x_1/t_1,...,x_n/t_n\}$ uma substituição. A *instanciação* de E por β ou, simplesmente, uma *instância* de E é a expressão Eβ obtida substituindo-se simultaneamente em E cada ocorrência de x_i por t_i, $i = 1,...,n$.

(b) Sejam **E** um conjunto de expressões e $\beta = \{x_1/t_1,...,x_n/t_n\}$ uma substituição. O conjunto de expressões $\mathbf{E}\beta = \{E\beta \,/\, E \in \mathbf{E}\}$ é a *instanciação* de **E** por β ou, simplesmente, uma *instância* de **E**.

(c) Seja C uma cláusula e β uma substituição. A *instanciação* de C por β, denotada por $C\beta$, é a cláusula obtida instanciando-se C por β e eliminando-se as ocorrências repetidas do mesmo literal, exceto a ocorrência mais à esquerda.

Definição 3.3: A *composição* de substituições é a função, denotada por "°", que mapeia pares de substituições em uma substituição e é definida da seguinte forma:

Para todo par de substituições

$\beta = \{x_1/t_1,..., x_n/t_n, y_1/s_1,..., y_k/s_k\}$
$\theta = \{y_1/r_1,...,y_k/r_k, z_1/q_1,..., z_m/q_m\}$

onde $x_1,..., x_n$, $y_1,..., y_k$, $z_1,..., z_m$ são variáveis distintas, a composição de β com θ será a substituição:

$$\beta \circ \theta = \{x_1/(t_1)\theta,..., x_n/(t_n)\theta, y_1/(s_1)\theta,..., y_k/(s_k)\theta, z_1/q_1,..., z_m/q_m\}$$

Definição 3.4: Seja $\mathbf{E} = \{E_1,...,E_n\}$ um conjunto de expressões simples e β uma substituição.

(a) β é um *unificador* de **E** se e somente se $(E_1)\beta = ... = (E_n)\beta$.
(b) β é um *unificador mais geral* (ou, abreviadamente, um *u.m.g.*) de **E** se e somente se β é um unificador de **E** e, para todo unificador θ de **E**, existe uma substituição φ tal que $\theta = \beta \circ \varphi$.
(c) O conjunto **E** é *unificável* se e somente se existe um unificador para **E**.

3.1.3. O Algoritmo da Unificação

Definição 3.5: Um conjunto de termos **D** é o *conjunto de discórdia* de um conjunto de expressões simples $\mathbf{E} = \{E_1,...,E_n\}$ se e somente se
(i) $\mathbf{D} = \varnothing$, se $n = 1$;
(ii) $\mathbf{D} = \{t_1,...,t_n\}$, se $n > 1$ e todas as expressões em **E** são idênticas até o i-ésimo símbolo, exclusive, e t_j é o termo ocorrendo em **E** que começa no i-ésimo símbolo, para $j = 1,...,n$.

Definição 3.6: (a) Uma substituição simples x/t *satisfaz o teste de ocorrência* se e somente se x não ocorre em t.
(b) Um conjunto de discórdia **D** *satisfaz o teste de ocorrência* se e somente se existem uma variável x e um termo t em **D** tais que x não ocorre em t.

Algoritmo 3.1: Algoritmo da Unificação

entrada: um conjunto **E** de expressões simples
saída: - um u.m.g. β de **E**, se **E** for unificável
- 'NÃO', se **E** não for unificável

início

$\beta := \varepsilon$;
W := **E**;
D := conjunto de discórdia de W;
enquanto (número de literais de W) > 1 e **D** satisfaz o teste de ocorrência *faça*
    início
    selecione uma variável *x* e um termo *t* em **D** tais que *x* não ocorre em *t*;
    $\beta := \beta \circ \{x/t\}$;
    $W := W\{x/t\}$;
    D := conjunto de discórdia de W;
    fim;
se (número de literais de W) = 1
    então retorne β
    senão retorne 'NÃO'
fim

3.1.4. O Sistema Formal da Resolução

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|.

Definição 3.7: 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'β.

Definição 3.8: 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

(i) L' e L'' têm sinais opostos e θ é um u.m.g. de {|L'|,|L''|}
(ii) $A = (A'\theta - L'\theta) \cup (A''\theta - L''\theta)$

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'θ de A'θ e L"θ de A"θ;
- listando os literais restantes de A'θ e A"θ em qualquer ordem;
- eliminando os literais repetidos.

Definição 3.9: 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"β.

Definição 3.10: 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".

Definição 3.11: Seja **S** um conjunto de cláusulas e C uma cláusula.

(a) 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:

(i) D_n = C
(ii) 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*.

(b) 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**.

Definição 3.12: Uma *dedução* de C a partir de um conjunto de cláusulas **S** é um conjunto finito de grafos acíclicos direcionados (GADs) {(t_1, R_1), ..., (t_n, R_n)} onde C rotula a raiz de algum t_i e todo t_i é tal que
(i) seus nós folha são rotulados com elementos de **S**,
(ii) todo nó não folha é rotulado com uma dupla $\langle R, \beta \rangle$ e tem exatamente dois nós pais rotulados com cláusulas P_1 e P_2, onde R é um resolvente de P_1 e P_2 formado usando o u.m.g. β.

Uma *refutação* é uma dedução de C =   , e uma dedução é *simples* se contém apenas um GAD (t, R).

3.2. Refutação por Resolução

3.2.1. Introdução

No problema da prova de teorema tem-se um conjunto de fórmulas **S**, a partir das quais 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, toda 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.

Se a resolução é aplicada repetidamente a um conjunto de cláusulas insatisfatíveis, eventualmente a cláusula vazia, , será produzida. Portanto, se W segue logicamente de **S**, então a resolução eventualmente 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.
$\forall x\ (\mathrm{L}(x) \rightarrow \mathrm{A}(x))$
(2) Os golfinhos não são alfabetizados.
$\forall x\ (\mathrm{G}(x) \rightarrow \neg\mathrm{A}(x))$
(3) Alguns golfinhos são inteligentes.
$\exists x\ (\mathrm{G}(x) \wedge \mathrm{I}(x))$

A partir destes quer-se provar a frase:

(4) Alguns que são inteligentes não podem ler.
$\exists x\ (\mathrm{I}(x) \wedge \neg\mathrm{L}(x))$

O conjunto de cláusulas que correspondem às frases 1 a 3 é:

(1) $\neg\mathrm{L}(x)\ \mathrm{A}(x)$
(2) $\neg\mathrm{G}(y)\ \neg\mathrm{A}(y)$
(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') ¬I(*z*) L(*z*).

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'
(6) A(a)	resolvente de 5 e 1
(7) ¬G(a)	resolvente de 6 e 2
(8)	resolvente de 7 e 3a.

3.2.2. Sistemas de Produção para refutação por resolução

Suponha um conjunto **S** de cláusulas chamado de *conjunto base*. O algorítmo básico para um sistema de produção de refutação por resolução pode ser escrito como:

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*

3.2.3. 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.

Uma refutação por resolução pode ser representada como uma *árvore de refutação* (dentro do grafo de derivação) tendo um nó raiz rotulado por .

A estratégia de controle busca por uma refutação crescendo o *grafo* de derivação até que uma *árvore* seja produzida com um nó raiz 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).

3.2.3.1. *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

I(a)	¬I(*z*) R(*z*)	¬R(*x*) L(*x*)	¬D(*y*) ¬L(*y*)	D(a)
1	2	3	4	5

Cláusulas originais **S**

1,2	2,3	3,4	4,5
R(a)	¬I(*z*) L(*z*)	¬R(*y*) ¬D(*y*)	¬L(a)
6	7	8	9

Resolventes de 1▪ nível

3,6	6,8	1,7	4,7	2,8	5,8	7,9
L(a)	¬D(a)	L(a)	¬I(*z*) ¬D(*z*)	¬I(*z*) ¬D(*z*)	¬R(a)	¬I(a)
10	11	12	13	14	15	16

Resolventes de 2▪ nível

6,15
ð
17

Resolventes de 3▪ nível

Ilustração da estratégia de busca em largura

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.

Exemplo: Exemplo do golfinho (anterior):

negação da meta

I(a)	¬I(z) R(z)	¬R(x) L(x)	¬D(y) ¬L(y)	D(a)
1	2	3	4	5

Cláusulas originais **S**

1,2	2,3
R(a)	¬I(z)L(z)
6	7

Resolventes de 1▪ nível

3,6	1,7	4,7
L(a)	L(a)	¬I(z) ¬D(z)
8	9	10

Resolventes de 2▪ nível

4,8	5,10	1,10	4,9
¬D(a)	¬I(a)	¬D(a)	¬D(a)
11	12	13	14

Resolventes de 3▪ nível

5,11
ð
15

Resolventes de 4▪ nível

Ilustração da estratégia de conjunto de suporte

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.

Exemplo: O exemplo do golfinho (anterior):

negação da meta

I(a)	¬I(*z*) R(*z*)	¬R(*x*) L(*x*)	¬D(*y*) ¬L(*y*)	D(a)
1	2	3	4	5

Cláusulas originais **S**

1,2

R(a)

6

Resolvente de 1▪ nível

3,6

L(a)

7

Resolvente de 2▪ nível

4,7

¬D(a)

8

Resolvente de 3▪ nível

5,8

ð

9

Resolvente de 4▪ nível

Ilustração da estratégia de preferência unitária

3.2.3.2. 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.

Exemplo: Exemplo do golfinho (anterior):

I(a)	¬I(*z*) R(*z*)	¬R(*x*) L(*x*)	¬D(*y*) ¬L(*y*)	D(a)
1	2	3	4	5

Cláusulas originais **S**

1,2	2,3	3,4	4,5
R(a)	¬I(*z*) L(*z*)	¬R(*y*) ¬D(*y*)	¬L(a)
6	7	8	9

Resolventes de 1▪ nível

3,6	1,7	4,7	2,8	5,8
L(a)	L(a)	¬I(*z*) ¬D(*z*)	¬I(*y*) ¬D(*y*)	¬R(a)
10	11	12	13	14

Resolventes de 2▪ nível

2,14
¬I(a)
15

Resolventes de 3▪ nível

1,15
ð
16

Resolventes de 4▪ nível

Ilustração da estratégia forma de entrada linear

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.

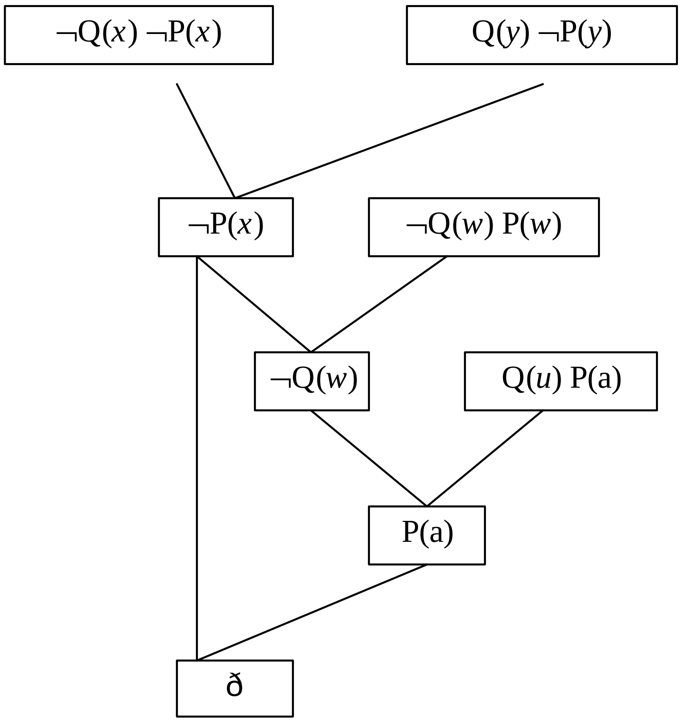

Ilustração da estratégia forma ancestral filtrada

3.2.4. 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.

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 $A_i\ \beta$ é um subconjunto de B_i . 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 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.

EXERCÍCIOS PROPOSTOS

3.1. Considere as cláusulas obtidas no exercício 2.1:

a) Prove que Josualdo gosta de amendoim utilizando a resolução.
b) Utilize a resolução para responder a pergunta "Que alimento Solange come?"

3.2. Suponha os seguintes fatos:

a) Elesbão gosta apenas de cursos fáceis.
b) Os cursos de ciência são difíceis.
c) Todos os cursos do departamento de fabricação de cestas são fáceis.
d) BK301 é um curso de fabricação de cestas.

Utilize a resolução para responder à pergunta "Que curso Elesbão gostaria de fazer?"

3.3. Considere os fatos seguintes:

a) Os sócios do Clube de Bolinha de Gude de São Petersburgo são Paul, Linda, Maguila e Sharon Stone.
b) Paul é casado com Linda.
c) Maguila é irmão de Sharon Stone.
d) O esposo de toda pessoa sócia do clube também é sócio do clube.
e) A última reunião do clube foi na casa de Paul.

1) Represente estes fatos na lógica de predicados.
2) Dos fatos acima, a maioria das pessoas seria capaz de decidir sobre a verdade das declarações adicionais seguintes:

f) A última reunião do clube foi na casa de Linda.
g) Sharon Stone não é casada.

Pode-se construir provas de resolução para demonstrar a verdade de cada uma dessas declarações, dados os cinco fatos listados acima? Faça-o, se possível. Caso contrário, acrescente os fatos que você precisar e depois construa as provas.

3.4. Aplique o princípio da resolução nas formas normais conjuntivas:

$$1.\ (P \wedge \neg Q) \vee M \ \text{ e } \ (P \rightarrow Q) \vee R$$
$$2.\ (P \vee \neg Q \vee M) \ \text{ e } \ R \vee \neg P \vee Q$$

3.5. Determine se cada um dos seguintes conjuntos de cláusulas é satisfatível:

a) $\{ \neg P \vee Q \vee R, \neg Q \vee S, P \vee S, \neg R, \neg S \}$
b) $\{ P \vee \neg Q, P \vee Q, \neg P \}$
c) $\{ \neg P \vee Q, P \vee \neg R, \neg Q, \neg R \}$

d) { $P \vee Q$, $\neg P \vee Q$, $P \vee \neg Q$, $\neg P \vee \neg Q$ }

3.6. Determine se cada um dos seguintes conjuntos de expressões é unificável:

a) { P(x, f(y), y), P(w, z, g(a,b)) }
b) { P(x, z, y), P(x, z, x), P(a, x, x) }
c) { P(a, x, f(x)), P(x,y,z) }
d) { P(z, f(x), b), P(x, f(a), b), P(g(x), f(a), y) }
e) { P(x, f(y), y), P(w, z, g(a,b)) }

3.7. Procure uma refutação para os seguintes conjuntos (inconsistentes) de cláusulas conectadas conjuntivamente usando a estratégia de controle chamada "forma de entrada linear":

P:
1. $\neg P \vee \neg Q \vee R$
2. $\neg S \vee T$
3. $\neg T \vee P$
4. S
5. $\neg R$
6. $\neg S \vee U$
7. $\neg U \vee Q$

Q:
1. $\neg P \vee R$
2. $\neg Q \vee \neg R$
3. $\neg S \vee T$
4. $\neg T \vee P$
5. S
6. $\neg S \vee J$
7. $\neg J \vee Q$

3.8. Se um curso é fácil, alguns estudantes no curso são felizes. Se um curso tem exame, nenhum estudante no curso é feliz. Use resolução para mostrar que, se um curso tem exame, o curso não é fácil.

3.9. Usando refutação mostre que o conjunto S de cláusulas:

1. $\neg A(x) \vee F(x) \vee G(f(x))$
2. $\neg F(x) \vee B(x)$
3. $\neg F(x) \vee C(x)$
4. $\neg G(x) \vee B(x)$
5. $\neg G(x) \vee D(x)$
6. $A(g(x)) \vee F(h(x))$

implica na "meta": $\exists x \exists y ((B(x) \wedge C(x)) \vee (D(y) \wedge B(y)))$

Não deixe de individualizar a variável de cada cláusula. Desenhe a árvore de refutação indicando claramente cada substituição.

3.10. Use resolução para mostrar que o conjunto de cláusulas abaixo é insatisfatível (inconsistente):

a) $\{ P(x,y) \vee Q(a,f(y)) \vee P(a,g(z)), \neg P(a,g(x)) \vee Q(a,f(g(b))), \neg Q(x,y) \}$

3.11. Indique quais das seguintes cláusulas são subjugadas por P(f(x),y):

a) $P(f(a), f(x)) \vee P(z, f(y))$
b) $P(z,a) \vee \neg P(a,z)$
c) $P(f(f(x)), z)$
d) $P(f(z), z) \vee Q(x)$
e) $P(a,a) \vee P(f(x), y)$

3.12. Prove por refutação que: $((A \rightarrow B) \rightarrow C) \rightarrow (A \rightarrow (B \rightarrow C))$

3.13. Mostre por refutação que a fórmula "$\exists x\ P(x)$" segue logicamente da fórmula "$P(a1) \vee P(a2)$". Entretanto, a forma skolemizada de "$\exists x\ P(x)$", denominada "P(a)", não segue logicamente de "$P(a1) \vee P(a2)$". Explique.

BIBLIOGRAFIA

- Casanova, M. A. & Giorno, F. A. C. & Furtado, A. L. (1987). Programação em Lógica e a Linguagem Prolog. Editora Edgard Blücher Ltda.

- Nilsson, N. J. (1982). Principles of Artificial Intelligence. Springer-Verlag.

- Stabler Jr., E. P. (1993). Parsing as non-Horn deduction. *Artificial Intelligence* 63, 225-264.