

MAC0444 - Sistemas Baseados em Conhecimento

Lista de Exercícios No. 2

Mateus Agostinho dos Anjos
NUSP 9298191

26 de Setembro de 2019

1 -

Predicados:

$fezEx(x)$ = x fez os exercícios

$vaiBem(x)$ = x vai bem na prova

$mediaAlta(x)$ = x fica com media alta

$aprovado(x, y)$ = x é aprovado em y

Formalizando as sentenças do enunciado chegamos em:

$$\forall x (fezEx(x) \rightarrow vaiBem(x))$$

$$\forall y (vaiBem(y) \rightarrow mediaAlta(y))$$

$$\forall z (mediaAlta(z) \rightarrow aprovado(z, mac444))$$

$$fezEx(Jo\tilde{a}o)$$

$$vaiBem(Maria)$$

Base de conhecimento (KB):

1. $[\neg fezEx(x), vaiBem(x)]$
2. $[\neg vaiBem(y), mediaAlta(y)]$
3. $[\neg mediaAlta(z), aprovado(z, mac444)]$
4. $[fezEx(Jo\tilde{a}o)]$
5. $[vaiBem(Maria)]$
6. $[\neg aprovado(Jo\tilde{a}o, mac444)]$

Veja que inserimos $[\neg aprovado(Jo\tilde{a}o, mac444)]$ na base de conhecimento, pois é a negação do nosso objetivo. Sendo assim, se chegarmos na cláusula vazia a partir desta base de conhecimento estará provado que $aprovado(Jo\tilde{a}o, mac444)$ é consequência lógica das sentenças do enunciado.

Utilizando a **resolução SLD** temos:

$$\begin{array}{ll}
\neg aprovado(Jo\tilde{a}o, mac444) & \text{(resolve com 3. e } z/Jo\tilde{a}o) \\
\downarrow & \\
\neg mediaAlta(Jo\tilde{a}o) & \text{(resolve com 2. e } y/Jo\tilde{a}o) \\
\downarrow & \\
\neg vaiBem(Jo\tilde{a}o) & \text{(resolve com 1. e } x/Jo\tilde{a}o) \\
\downarrow & \\
\neg fezEx(Jo\tilde{a}o) & \text{(resolve com 4.)} \\
\downarrow & \\
[] &
\end{array}$$

Sendo assim provamos que: $KB \cup \{\neg aprovado(Jo\tilde{a}o, mac444)\}$ é insatisfazível, portanto $aprovado(Jo\tilde{a}o, mac444)$ é consequência lógica de nossa base de conhecimento.

A **resolução SLD** será semelhante para Maria, portanto temos:

Base de conhecimento (KB):

1. $[\neg fezEx(x), vaiBem(x)]$
2. $[\neg vaiBem(y), mediaAlta(y)]$
3. $[\neg mediaAlta(z), aprovado(z, mac444)]$
4. $[fezEx(Jo\tilde{a}o)]$
5. $[vaiBem(Maria)]$
6. $[\neg aprovado(Maria, mac444)]$

Utilizando a **resolução SLD** temos:

$$\begin{array}{ll}
 \neg aprovado(Maria, mac444) & \text{(resolve com 3. e } z/Maria) \\
 \downarrow & \\
 \neg mediaAlta(Maria) & \text{(resolve com 2. e } y/Maria) \\
 \downarrow & \\
 \neg vaiBem(Maria) & \text{(resolve com 5.)} \\
 \downarrow & \\
 [] &
 \end{array}$$

2 -

Temos a Base de Conhecimento (KB) reescrita com variáveis renomeadas para evitar confusões na resolução do exercício:

1. $[\neg A_1(x), \neg A_2(x), P(x)]$
2. $[\neg B_1(y), \neg B_2(y), A_1(y)]$
3. $[\neg B_3(z), \neg B_4(z), A_2(z)]$
4. $[B_1(a)]$
5. $[B_2(a)]$
6. $[B_3(a)]$
7. $[B_4(a)]$

a)

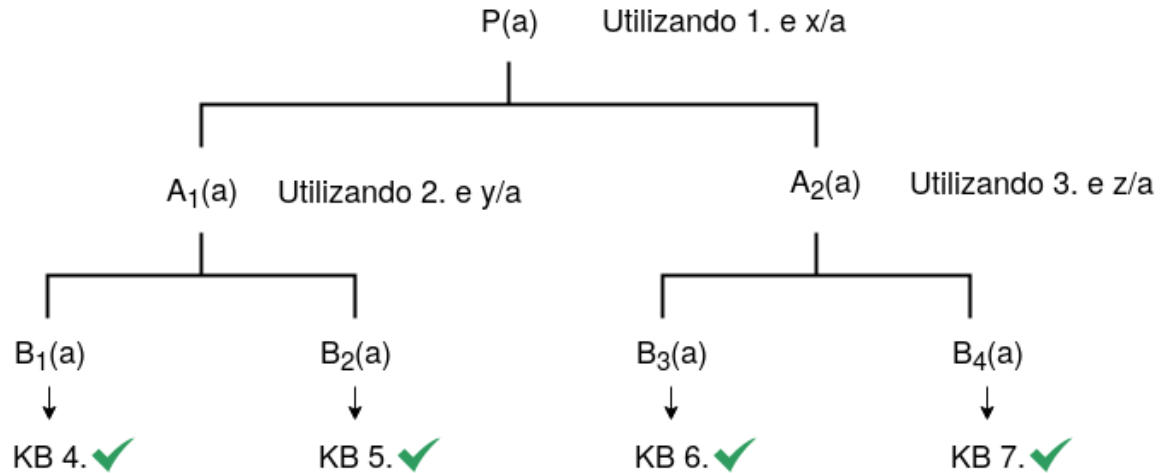
Para mostrar o passo a passo do procedimento de encadeamento para trás (backward chaining) devemos começar identificando as implicações da Base de Conhecimento.

Seguindo a ordem acima temos:

(note que utilizamos \leftarrow nas implicações)

1. $\forall x(P(x) \leftarrow A_1(x) \wedge A_2(x))$
2. $\forall y(A_1(y) \leftarrow B_1(y) \wedge B_2(y))$
3. $\forall z(A_2(z) \leftarrow B_3(z) \wedge B_4(z))$
4. $B_1(a)$
5. $B_2(a)$
6. $B_3(a)$
7. $B_4(a)$

A partir destas implicações o passo a passo pode ser mostrado a partir da figura abaixo, sendo que cada passo gera pelo menos uma sub-árvore.



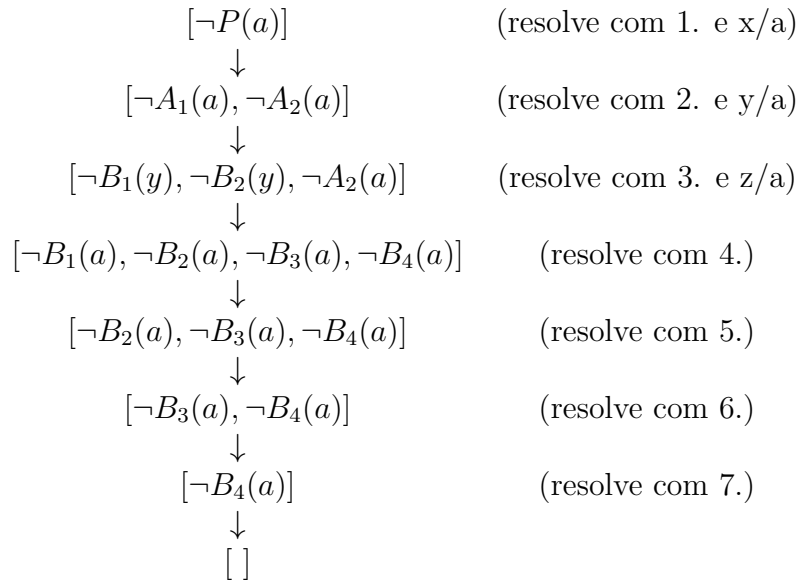
Todas as folhas da árvore estão na base de conhecimento em uma cláusula unitária, portanto são verdadeiras. Sendo assim podemos marcá-las com um ✓.

A partir disso podemos concluir que $A_1(a)$ e $A_2(a)$ estão provados e depois que $P(a)$ está provado, portanto mostramos que o encadeamento para trás (backward chaining) produz resposta SIM com objetivo $P(a)$.

b)

Utilizando a **resolução SLD** com a base de conhecimento definida no início da questão, iniciamos com $\neg P(a)$, que é a negação do nosso objetivo, e buscaremos a cláusula vazia.

Obtemos o seguinte:



Como chegamos na cláusula vazia a partir de $\neg P(a)$, então está provado que $P(a)$ é consequência desta base de conhecimento.

3 -

a)

Após ter carregado o programa a resposta do Prolog para a consulta:

$? - result([a, b, c, d, e, f, g], X).$

será:

$X = [b, d, f]$