Semânticas de Linguagens de Programação - CC3033

Exercícios 10

Programação em Lógica

1 Programas simples em Prolog

1. Considera os seguintes predicados em Prolog e cria alguns factos concretos:

   - `pais(N,C,H,L)` se `N` é um país, com capital `C`,habitantes `H` (inteiro) e língua `L`
   - `cidade(A,P)` se a cidade `A` é o do país `P` (sem ser a capital).
   - `comboio(A,B)` existe comboio entre a cidade `A` e a cidade `B`.

   Define predicados correspondentes às seguintes relações:

   (a) `capital(X,Y)` se `X` é capital do país `Y`.

   (b) `maishabitantes(X,Y)` se o país `X` tem mais habitantes que o país s `Y`.

   (c) `mesmalingua(X,Y)` se o país `X` tem a mesma língua que o país `Y`.

   (d) `mesmopais(X,Y)` se as cidades `X` e `Y` são do mesmo país.

   (e) `ircomboio(X,Y)` se se pode ir de comboio da cidade `X` para a cidade `Y`.

   (f) `comboioentre(X,Y)` se se pode ir de comboio de uma cidade do pais `X` para uma cidade do pais `Y`.

2. Define os seguintes predicados relacionados com listas

   (a) `last(X,L)`: `X` é o último elemento da lista `L`

   (b) `substitute(X,Y,L1,L2)`: `L2` é a lista que resulta de substituir as ocorrências de `X` na lista `L1` por `Y`

   (c) `reverse(L1,L2)`: `L2` é o resultado de inverter a ordem dos elementos da lista `L`

   (d) `palindrome(L)`: a lista `L` é um palíndroma, ou seja, se é lida da mesma forma da direita para a esquerda e da esquerda para a direita

   (e) `divide(L,L1,L2)`: `L1` e `L2` são o resultado de dividir a lista `L` em duas partes com o mesmo comprimento. Se o tamanho de `L` for ímpar, o predicado deve falhar

   (f) `nth(X,N,L)`: `X` é o `N`-ésimo elemento da lista `L`, ou seja, é o elemento que está na posição `N` (a partir de 1) da lista

   (g) `count(X,N,L)`: `N` é o número de vezes que o elemento `X` ocorre na lista `L`.

   (h) `deslocaesquerda(L,L1)`: `L1` é a lista que resulta de `L` deslocando uma posição para a esquerda os elementos de `L` e colocando o primeiro elemento de `L` na última posição de `L1`.

3. (a) Supõe que os inteiros não negativos são denotados pelos termos `0`, `s(0)`, `s(s(0))`, `s(s(s(0)))`, ..., ou seja $\underbrace{\texttt{s(s(s(...s(0)...)))}}_{\text{qualquer número} \geq 0}$, os quais passamos a referir como sendo da forma $\texttt{s}^i(0)$, com $i \in dN$.

   1. Define um predicado `soma(X,Y,Soma)` para calcular a soma de `X` com `Y`.
   2. Define um predicado `subtr(X,Y,Soma)` para calcular X-Y, instanciando `Soma` com `0` se `X` for inferior a `Y`.
   3. Escrever um predicado `soma_seg(L,Lf)` para substituir cada um dos elementos de `L` pela soma dos seguintes, sendo os elementos de `L` podem ser da forma `0`, `s(0)`, `s(s(0))`, `s(s(s(0)))`, ...

4. Supõe que numa árvore binária cada nó é representado por um termo `no(V,E,D)`, onde `V` é o valor do nó, `E` (respectivamente, `D`) a sub-árvore esquerda (direita). A árvore vazia é representada por `[]`. Escreve predicados Prolog que:
   (a) Dadas duas árvores binárias suceda se a segunda é uma sub-árvore da primeira.
   (b) `soma_arv(A,S)` que sucede se `A` é uma árvore binária de inteiros e `S` a soma de todos os valores dos nós.
   (c) Dada uma árvore `T` suceda se `T` for uma árvore ordenada de inteiros, isto é, para cada nó o valor é um inteiro e os valores na subárvore esquerda são menores que esse valor e os valores na subárvore direita são maiores que esse valor.
   (d) Determinar se um elemento pertence a uma árvore ordenada.
   (e) Apagar um elemento de uma árvore ordenada.
   (f) Permita inserir um elemento numa árvore binária ordenada `T` resultando uma árvore ordenada `T1`.

5. Escreve um predicado `aceita(Pal,Trans,Finais,Estado,Hist)` para verificar se um dado autómato finito aceita uma dada palavra `Pal`, onde `Trans` é uma lista de termos da forma `t(Est,Simb,Estf)`, representando a relação de transição, `Finais` é a lista dos estados que são finais, `Estado` é o estado em que o autómato se encontra, e `Hist` é a lista das transições efectuadas na análise da palavra. A transição `t(Est,Simb,Estf)` deve ser lida como "se o autómato estiver no estado `Est` e consumir `Simb`, então passa para o estado `Estf`" e `Pal` é a lista de símbolos da palavra. Considera ainda cada um dos casos seguintes.
   (a) O autómato é determinístico.
   (b) O autómato é não-determinístico podendo ter transições ε, as quais são codificadas usando `[ ]` no campo `Simb`.

2 Semântica de Programação em Lógica

1. Escreve as cláusulas abaixo sem ser em notação clausal:

   $p(a) \leftarrow p(X), q(X)$
   $p(f(X)) \leftarrow p(X)$
   $q(b) \leftarrow$
   $q(f(X)) \leftarrow q(X)$
   $\leftarrow q(f(f(a)))$

2. Calcula um umg (unificador mais geral) ou indica que não existe para cada um dos seguintes conjuntos de equações de termos:
   (a) $\{g(a, X, h(g(Z))) = g(Z, h(Y), h(Y))\}$
   (b) $\{X = g(Y), g(U) = g(Z), U = a\}$
   (c) $\{g(X, f(X, W)) = g(g(W), f(g(b), Z))\}$
   (d) $\{g(Z, h(Z), Z) = g(f(V), h(h(X)), X)\}$

3. Para o programa definido $\mathcal{P}$ e objectivo G (da forma $\leftarrow \beta$) descritos abaixo:
   - Calcula uma refutação **SLD** para $\mathcal{P} \cup \{G\}$, indicando as substituições em cada passo e a resposta calculada, no fim. Representa a refutação-SLD por uma árvore de derivação.

- Verifica a execução do programa com o objectivo em Prolog.

(a) $P: \quad Add(X, 0, X) \leftarrow$
$\quad\quad\quad Add(X, s(Y), s(Z)) \leftarrow Add(X, Y, Z)$

$G: \quad \leftarrow Add(s(0), X, s(s(Y))$

(b) $P: \quad Length(nil, 0) \leftarrow$
$\quad\quad\quad Length(cons(X, Y), s(Z)) \leftarrow Length(Y, Z)$

$G: \quad \leftarrow Length(X, s(s(Y)))$

4. Considere o programa $\mathcal{P}$

$$p(f(X)) \leftarrow p(X)$$
$$q(a) \leftarrow p(X)$$

Para $I_1 = B_{\mathcal{P}}$, $I_2 = T_{\mathcal{P}}(I_1)$ e $I_3 = \emptyset$ determine $T_{\mathcal{P}}(I_1)$, $T_{\mathcal{P}}(I_2)$ e $T_{\mathcal{P}}(I_3)$.

5. Seja $\mathcal{P}$ o programa

$$\mathsf{p(f(X), g(Y))} \leftarrow \mathsf{p(X, Y)}$$
$$\mathsf{p(a, a)} \leftarrow$$
$$\mathsf{p(a, b)} \leftarrow$$

, G o objectivo $\leftarrow \mathsf{p(X, Y)}$.

Quais das substituições seguintes são respostas correctas para $\mathcal{P} \cup \{G\}$? Para as substituições θ que são respostas correctas para $\mathcal{P} \cup \{G\}$ determina uma resposta calculada σ para $\mathcal{P} \cup \{G\}$ e uma substituição γ tal que $\theta = \sigma\gamma$. Representa ainda a refutação-SLD que corresponde a σ por uma árvore de derivação.

- $\theta = [a/X, a/Y, b/Z]$;
- $\theta = [a/X]$;
- $\theta = [a/X, X/Y]$;
- $\theta = [a/X, b/Y]$;
- $\theta = [f(X)/X, g(Y)/Y]$;
- $\theta = [f(f(a))/X, g(g(a))/Y]$.

6. Seja $\mathcal{P}$ o programa seguinte:

$$\mathsf{s(h(X)))} \leftarrow \mathsf{r(X), p(h(X))}$$
$$\mathsf{r(a)} \leftarrow$$
$$\mathsf{r(b)} \leftarrow$$
$$\mathsf{p(a)} \leftarrow$$
$$\mathsf{p(h(X))} \leftarrow$$

(a) Escreve cada fórmula do programa $\mathcal{P}$ sem ser na notação clausal.

(b) Mostra que $\mathcal{P} \vdash \exists X \mathsf{s(X)}$.

(c) Determina o conjunto de termos $\{\mathsf{t} \in \mathcal{T}_0 \mid \mathcal{P} \vdash \mathsf{s(t)}\}$.

(d) Determina o modelo mínimo de Herbrand de $\mathcal{P}$.

7. Considera o programa $\mathcal{P}$ seguinte:

$$\begin{array}{l} \mathsf{s(f(X))} \leftarrow \mathsf{s(X), r(Y)} \\ \mathsf{r(b)} \leftarrow \\ \mathsf{r(f(X))} \leftarrow \mathsf{r(X)} \end{array}$$

   (a) Escreve cada fórmula do programa sem ser na *notação clausal* (i.e com os quantificadores e operações lógicas usuais).

   (b) Determina L_P, U_P e B_P.

   (c) Calcula $T_P \uparrow 0$, $T_P \uparrow 1$, $T_P \uparrow 2$, $\ldots$, $T_P \uparrow n$ e $T_P \uparrow \omega$.

8. Considera G o objectivo $\leftarrow \mathsf{p(X,X)}$ e o programa $\mathcal{P}$ seguinte:

$$\begin{array}{l} \mathsf{p(a,b)} \leftarrow \\ \mathsf{p(X,Z)} \leftarrow \mathsf{p(X,Y), p(Y,Z)} \\ \mathsf{p(X,Y)} \leftarrow \mathsf{p(Y,X)} \end{array}$$

   (a) Calcula uma refutação-**SLD** para $\mathcal{P} \cup \{G\}$ e indica a resposta calculada.

   (b) Justifica a veracidade ou a falsidade das seguintes afirmações:

      1. $\mathcal{P} \models \exists \mathsf{X}\, \mathsf{p(X,X)}$
      2. $\mathcal{P} \models \forall \mathsf{X}\, \mathsf{p(X,X)}$

   (c) Determina $M_{\mathcal{P}}$ e o modelo mínimo de Herbrand de $\mathcal{P}$.

9. Para o programa definido $\mathcal{P}$ e objectivo G (da forma $\leftarrow \beta$) descritos abaixo:

   - Escreve cada fórmula do programa e do objectivo sem ser em notação clausal (i.e com quantificadores e operações lógicas)
   - Determina L_P, U_P e B_P.
   - Calcula $T_P \uparrow 0$, $T_P \uparrow 1$, $\ldots T_P \uparrow n$ e $T_P \uparrow \omega$.
   - Determina M_P e o modelo mínimo de Herbrand de $\mathcal{P}$.
   - Calcula uma refutação **SLD** para $P \cup \{G\}$, indicando as substituições em cada passo e a resposta calculada, no fim. Representa a refutação-SLD por uma árvore de derivação.
   - Justifica a veracidade ou falsidade das seguintes afirmações:
      1. $\mathcal{P} \models \exists \beta$
      2. $\mathcal{P} \models \forall \beta$
   - Verifica a execução do programa com o objectivo em Prolog.

   (a) $P:$ $\quad Add(X, 0, X) \leftarrow$
   $\quad Add(X, s(Y), s(Z)) \leftarrow Add(X, Y, Z)$

   $G:$ $\quad \leftarrow Add(s(0), X, s(s(Y))$

   (b) $P:$ $\quad Length(nil, 0) \leftarrow$
   $\quad Length(cons(X, Y), s(Z)) \leftarrow Length(Y, Z)$

   $G:$ $\quad \leftarrow Length(X, s(s(Y)))$

(c)

$$P: \quad Append(cons(X, X_R), X_L, cons(X, X_{RL})) \leftarrow Append(X_R, X_L, X_{RL})$$
$$Append(nil, X_L, X_L) \leftarrow$$

$$G: \quad \leftarrow Append(cons(X, nil), cons(X, nil), Y)$$

(d)

$$P: \quad Member(X, cons(X, Y)) \leftarrow$$
$$Member(X, cons(Y, Z)) \leftarrow Member(X, Z)$$

$$G: \quad \leftarrow Member(X, cons(Y, cons(X, Z)))$$

(e)

$$P: \quad Split(cons(X, Y), cons(X, nil), Y) \leftarrow$$
$$Split(cons(X, Y), cons(X, X_1), Y_1) \leftarrow Split(Y, X_1, Y_1)$$

$$G: \quad \leftarrow Split(cons(Y, cons(X, nil)), Z, W)$$

(f)

$$P: \quad p(s(X), Y) \leftarrow I(Y, Z), p(X, Z)$$
$$p(\mathsf{nil}, Y) \leftarrow Fi(Y)$$
$$I(a, b) \leftarrow$$
$$I(b, c) \leftarrow$$
$$I(c, b) \leftarrow$$
$$Fi(b) \leftarrow$$

$$G: \quad \leftarrow p(s(X), a)$$

(g)

$$P: \quad O(r(X)) \leftarrow p(X)$$
$$p(0) \leftarrow$$
$$p(r(r(X))) \leftarrow p(X)$$

$$G: \quad \leftarrow O(r(r(r(X))))$$

Resolução de exercícios selecionados

Resolução (7.a)

$$\forall X \forall Y \, (s(f(X)) \ \vee \ \neg s(X) \ \vee \ \neg r(Y))$$
$$r(b)$$
$$\forall X \, (r(f(X)) \ \vee \ \neg r(X))$$

Resolução (7.b)

$L_P : \mathcal{F}_0 = \{\mathsf{b}\}$, $\mathcal{F}_1 = \{\mathsf{f}\}$,
$\mathcal{R}_1 = \{\mathsf{s}, \mathsf{r}\}$
$U_P = \{\mathsf{b}, \mathsf{f(b)}, \mathsf{f(f(b))}, \ldots\}$
$B_P = \{\mathsf{s(b)}, \mathsf{r(b)}, \mathsf{s(f(b))}, \mathsf{r(f(b))}, \mathsf{s(f(f(b)))}, \mathsf{r(f(f(b)))}, \ldots\}$

Resolução (7.c)

$$
\begin{array}{rcl}
T_P \uparrow 0 & = & \emptyset \\
T_P \uparrow 1 & = & \{\mathsf{r(b)}\} \\
T_P \uparrow 2 & = & \{\mathsf{r(b)}, \mathsf{r(f(b))}\} \\
T_P \uparrow n & = & \{\mathsf{r(f^k(b))} \mid 0 \leq k \leq n-1\} \\
T_P \uparrow \omega & = & \{\mathsf{r(f^n(b))} \mid n \geq 0\}
\end{array}
$$

Resolução (8.a)

$\leftarrow \mathsf{p(X,X)}$	$\mathsf{p(X_1,z_1)} \leftarrow \mathsf{p(X_1,Y_1)}, \mathsf{p(Y_1,z_1)}$
	$\sigma_1 = [\mathsf{x/X_1}, \mathsf{x/z_1}]$
$\leftarrow \mathsf{p(X,Y_1)}, \mathsf{p(Y_1,X)}$	$\mathsf{p(a,b)} \leftarrow$
	$\sigma_2 = [\mathsf{a/x/}, \mathsf{b/Y_1}]$
$\leftarrow \mathsf{p(b,a)}$	$\mathsf{p(X_2,Y_2)} \leftarrow \mathsf{p(Y_2,X_2)}$
	$\sigma_3 = [\mathsf{b/X_2/}, \mathsf{a/Y_2}]$
$\leftarrow \mathsf{p(a,b)}$	$\mathsf{p(a,b)} \leftarrow$
	$\sigma_4 = \iota$

ϵ

$\sigma_1\sigma_2\sigma_3\sigma_4 = [\mathsf{a/X_1}, \mathsf{a/z_1}, \mathsf{a/X}, \mathsf{b/Y_1}, \mathsf{b/X_2}, \mathsf{a/Y_2}]$

Resposta calculada:$\sigma = [\mathsf{a/x}]$

Resolução (8.b)

1. true Como G é $\neg\exists \mathsf{X p(X,X)}$ e, pela alinea (a) $P \cup \{\neg\exists \mathsf{X p(X,X)}\}$ não é satisfazível, tem-se que $\mathcal{P} \models \exists \mathsf{X p(X,X)}$.

2. false Seja $\mathcal{A} = (\{\mathsf{a,b,c}\}, \cdot^{\mathcal{A}})$ onde $\mathsf{a}^{\mathcal{A}} = \mathsf{a}$, $\mathsf{b}^{\mathcal{A}} = \mathsf{b}$ e $\mathsf{p}^{\mathcal{A}} = \{(\mathsf{a,a}), (\mathsf{a,b}), (\mathsf{b,a}), (\mathsf{b,b})\}$. Temos que $\mathcal{A} \models \mathcal{P}$ (verifica!) e $\mathcal{A} \not\models \forall \mathsf{X}\, \mathsf{p(X,X)}$ (porque $(\mathsf{c,c}) \notin \mathsf{p}^{\mathcal{A}}$).

Resolução (8.c)

$M_{\mathcal{P}} = T_P \uparrow \omega$ e

$$
\begin{array}{rcl}
T_P \uparrow 0 & = & \emptyset \\
T_P \uparrow 1 & = & \{\mathsf{p(a,b)}\} \\
T_P \uparrow 2 & = & \{\mathsf{p(a,b)}, \mathsf{p(b,a)}\} \\
T_P \uparrow 3 & = & \{\mathsf{p(a,b)}, \mathsf{p(b,a)}, \mathsf{p(a,a)}, \mathsf{p(b,b)}\} \\
T_P \uparrow 4 & = & T_P \uparrow 3 \\
T_P \uparrow n & = & T_P \uparrow 3 \\
T_P \uparrow \omega & = & T_P \uparrow 3
\end{array}
$$

O modelo mínimo de $\mathcal{P}$ é a estrutura $\mathcal{A}_P = (\{\mathsf{a,b}\}, \cdot^{\mathcal{A}})$ onde $\mathsf{a}^{\mathcal{A}} = \mathsf{a}$, $\mathsf{b}^{\mathcal{A}} = \mathsf{b}$ e $p^{\mathcal{A}} = \{(\mathsf{a,a}), (\mathsf{a,b}), (\mathsf{b,a}), (\mathsf{b,b})\}$