

GABARITO DA PRIMEIRA PROVA

TEORIA DA COMPUTAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA, UNIVERSIDADE DE BRASÍLIA
 PROF. MAURICIO AYALA-RINCÓN

17 DE OUTUBRO DE 2011

1. (2.0 pontos) Demonstre usando o princípio de resolução no cálculo proposicional que a fórmula

$$\phi = p \wedge (q \rightarrow r)$$

é consequência lógica do conjunto de fórmulas

$$\Phi = \{(p \rightarrow q) \rightarrow r, r \rightarrow p\}$$

Para isto siga o roteiro padrão:

- (a) (1.0 ponto) Construa um conjunto de cláusulas proposicionais para Φ e $\neg\phi$.
- (b) (1.0 ponto) Construa então uma refutação para o conjunto de cláusulas do item precedente.

R/ Conforme o método de resolução no cálculo proposicional, demonstra-se que o conjunto de fórmulas $\{\neg\phi\} \cup \Phi$ é inconsistente.

- (a) Inicialmente as fórmulas em Φ e $\neg\phi$ são transformadas na forma de cláusulas:

$$\begin{array}{ll} \neg\phi & \Phi \\ \equiv & \equiv \\ \neg p \vee (q \wedge \neg r) & \{(p \wedge \neg q) \vee r, \neg r \vee p\} \\ \equiv & \equiv \\ (\neg p \vee q) \wedge (\neg p \vee \neg r) & \{p \vee r, \neg q \vee r, \neg r \vee p\} \end{array}$$

- (b) A seguir demonstra-se que o conjunto de todas as fórmulas é inconsistente.

$$\frac{\frac{\neg p \vee \neg r \quad \neg r \vee p}{\neg r \vee \neg r} (s) \quad (c)}{\neg r} \quad \frac{\frac{\frac{\neg p \vee q \quad \neg q \vee r}{\neg p \vee r} (c) \quad p \vee r}{r \vee r} (s) \quad (c)}{r} (c)}{\square}$$

2. (2.0 pontos) Considere o seguinte programa definido:

$$P := \left\{ \begin{array}{l} p(0) \\ i(s(0)) \\ p(s(s(X))) \leftarrow p(X) \\ i(s(s(X))) \leftarrow i(X) \end{array} \right\}$$

Indicando todos os cálculos necessários, encontre:

- (a) (0.4 pontos) U_P e B_P ;
- (b) (0.6 pontos) $T_P \uparrow \omega$;
- (c) (0.6 pontos) $T_P \downarrow \omega$;
- (d) (0.2 pontos) o mínimo ponto fixo de T_P ($m\mathcal{PF}(T_P)$) e
- (e) (0.2 pontos) o máximo ponto fixo de T_P ($M\mathcal{PF}(T_P)$).

R/

(a) $U_P = \{s^k(0) \mid k \in \mathbb{N}\}$; $B_P = \{p(s^k(0)) \mid k \in \mathbb{N}\} \cup \{i(s^k(0)) \mid k \in \mathbb{N}\}$;

(b)

$T_P \uparrow \omega$:

$$T_P \uparrow 0 = \emptyset;$$

$$T_P \uparrow 1 = \{p(0), i(s(0))\};$$

$$T_P \uparrow 2 = \{p(0), i(s(0)), p(s^2(0)), i(s^3(0))\};$$

\vdots

$$T_P \uparrow \omega = \{p(s^k(0)) \mid k \text{ par em } \mathbb{N}\} \cup \{i(s^k(0)) \mid k \text{ ímpar em } \mathbb{N}\}.$$

(c)

$T_P \downarrow \omega$:

$$T_P \downarrow 0 = B_P;$$

$$T_P \downarrow 1 = B_P \setminus \{i(0), p(s(0))\};$$

$$T_P \downarrow 2 = B_P \setminus \{i(0), p(s(0)), i(s^2(0)), p(s^3(0))\};$$

\vdots

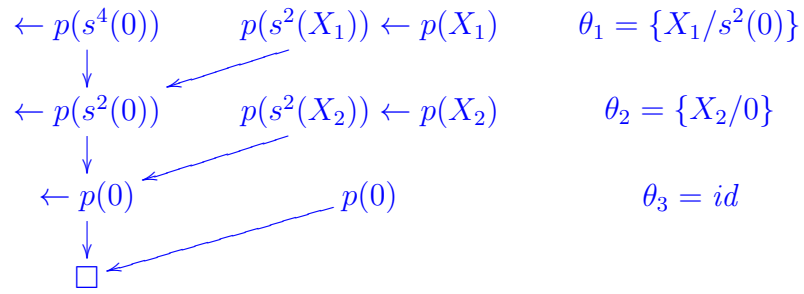
$$T_P \downarrow \omega = \{p(s^k(0)) \mid k \text{ par em } \mathbb{N}\} \cup \{i(s^k(0)) \mid k \text{ ímpar em } \mathbb{N}\}.$$

(d) e (e) $m\mathcal{PF}(T_P) = M\mathcal{PF}(T_P) = T_P \uparrow \omega$.

3. (2.0 pontos) Considere novamente o programa P da questão 2. Indicando todos os cálculos necessários, construa uma refutação para o objetivo $\leftarrow p(s^4(0))$ com o programa P .

Lembre que uma refutação é uma *derivação* cuja cláusula inicial é o objetivo e a final é a cláusula vazia e que consiste de seqüências de objetivos, variantes de cláusulas e substituições, onde em cada passo o novo objetivo é obtido como a resolvente do objetivo anterior com as respectivas cláusula e substituição.

R/



4. (2.0 pontos) Considere o seguinte programa definido:

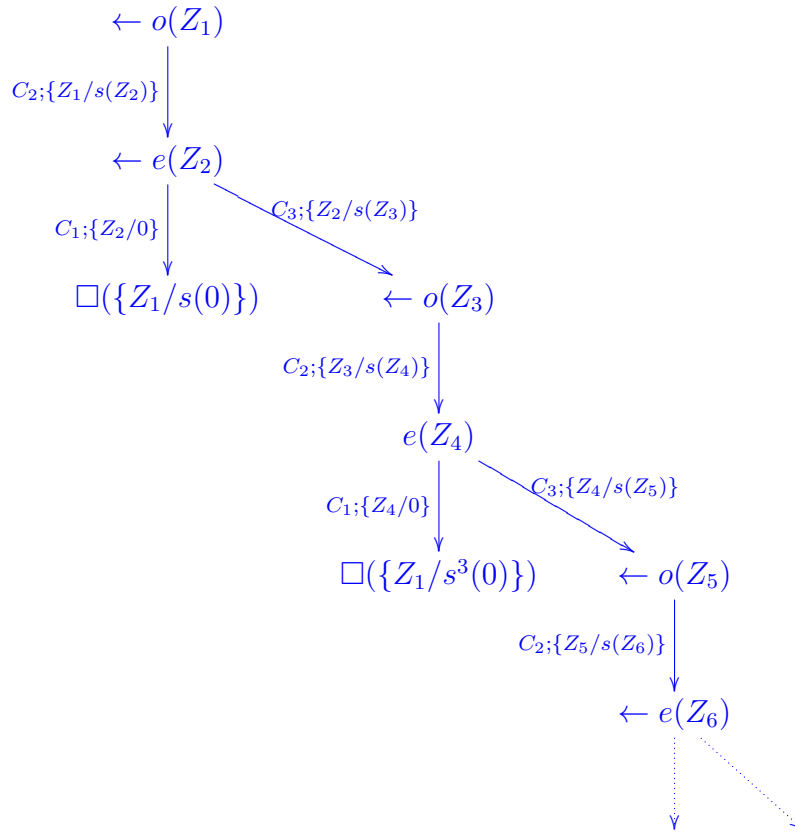
$$Q := \left\{ \begin{array}{l} C_1 : e(0) \\ C_2 : o(s(X)) \leftarrow e(X) \\ C_3 : e(s(X)) \leftarrow o(X) \end{array} \right\}$$

Considerando a regra de computação e estratégia de busca padrão,

- (1.0 pontos) construa a árvore de derivação para o objetivo $\leftarrow o(Z_1)$.
- (0.4 pontos) Indique em cada aresta da árvore a cláusula e substituição utilizadas.
- (0.4 pontos) Em cada folha de ramos de sucesso indique a resposta computada associada.
- (0.2 pontos) Responda quantos ramos de sucesso e quantos de falha tem a árvore.

R/

(a),(b) e (c)



(d) A árvore tem infinitos ramos de sucesso e zero de falha.

5. (2.0 pontos) Que a noção informal de “computável” coincida com noções formais de “computabilidade” formalizadas via modelos específicos de computação, como são as máquinas de Turing, o lambda cálculo, a teoria de reescrita, etc., é o que se conhece como a *Tese de Church-Turing*. O primeiro resultado neste sentido relaciona as funções recursivas parciais e as *computáveis com máquinas de Turing* (1936).

Neste sentido, para provar a *adequação computacional da SLD-resolução* basta, então, demonstrar, p. ex., que as funções recursivas parciais são computáveis via programas definidos. Isto é demonstrado por indução na estrutura destas funções, que podem ser construídas a partir das funções básicas (zero, sucessor e projeções) e operações de composição, recorrência primitiva e minimização.

No passo indutivo desta demonstração, para provar que é possível especificar funções construídas por **minimização**, temos o seguinte *sketch*:

Suponha que f é definida como

$$f(x_1, \dots, x_n) = \mu y (g(x_1, \dots, x_n, y) = 0)$$

sendo g uma função recursiva parcial. $\mu y(g(x_1, \dots, x_n, y) = 0)$ é definida como o mínimo y tal que $g(x_1, \dots, x_n, y) = 0$ e $g(x_1, \dots, x_n, z)$ está definida para todo $z \leq y$ e é indefinida caso contrário. Pela hipótese de indução existe um programa definido P_g e um predicado p_g para g . Defina P_f como P_g juntamente com as seguintes cláusulas:

$$\begin{aligned} p_f(x_1, \dots, x_n, y) &\leftarrow p_g(x_1, \dots, x_n, 0, z_1), r(x_1, \dots, x_n, 0, z_1, y). \\ r(x_1, \dots, x_n, y, 0, y) & \\ r(x_1, \dots, x_n, y, s(z_1), z) &\leftarrow p_g(x_1, \dots, x_n, s(y), u), r(x_1, \dots, x_n, s(y), u, z). \end{aligned}$$

Observe que o predicado r é utilizado para computar em ordem ascendente, utilizando o predicado p_g , imagens de $g(x_1, \dots, x_n, 0), g(x_1, \dots, x_n, s(0)), \dots$, até que seja obtida uma com valor zero. O processo será interrompido caso se atinge uma t pula para a qual g esteja indefinida.

Quest o: Complete a demonstra o desta parte do passo indutivo provando que as respostas computadas para p_f correspondem exatamente  s imagens da fun o f .

R/ Vide exerc cio das notas de aula.