

Métodos Formais em Engenharia de Software

Módulo de Processos e Arquitecturas

Teste de avaliação com consulta autorizada

10 de Julho de 2008, 13:30 h

O teste é composto por 10 alíneas, com idêntica valorização, e tem duração total de 2 horas.

I (fundamentos)

Recorde, das aulas sobre *coindução*, o operador *merge* definido por

$$\begin{array}{ccc} B^\omega & \xrightarrow{\langle \text{hd}, \text{tl} \rangle} & B \times B^\omega \\ \text{merge} \uparrow & & \uparrow \text{id} \times \text{merge} \\ B^\omega \times B^\omega & \xrightarrow{g} & B \times (B^\omega \times B^\omega) \end{array}$$

onde

$$g = \langle \text{hd} \cdot \pi_1, \text{swap} \cdot (\text{tl} \times \text{id}) \rangle$$

Considere, ainda, os seguintes operadores sobre *streams*

$$\text{xpto} = \llbracket \langle \text{hd} \cdot \text{tl}, \text{tl} \cdot \text{tl} \rangle \rrbracket$$

$$\text{otpx} = \llbracket \langle \text{hd}, \text{tl} \cdot \text{tl} \rangle \rrbracket$$

1. Derive uma versão de *xpto* onde a invocação recursiva seja explícita e explique o propósito dos dois operadores.
2. Mostre, por recurso à propriedade universal da extensão coindutiva, que

$$\text{otpx} \cdot \text{merge} = \pi_1$$

3. Repita a prova anterior construindo uma bissimulação adequada.
4. Explique, de forma sucinta mas clara, a razão pela qual os dois métodos de prova usados na resolução das alíneas anteriores são equivalentes.

II (arquitecturas)

Recorde a noção de *estilo arquitectural* estudada neste módulo e, em particular, o popular estilo *pipe & filter*.

1. Caracterize este estilo e enumere algumas das suas aplicações típicas.
2. Considere a seguinte especificação, na linguagem de processos que estudou, da noção de *pipe*

$$U \triangleright V \stackrel{\text{abv}}{\equiv} \mathbf{new} \{c\} (\{c/out\}U \mid \{c/in\}V)$$

assumindo que, em ambos os processos, as acções \overline{out} e in representam, respectivamente, a activação de portas de saída e entrada.

- (a) Considere, agora, os seguintes processos, só parcialmente definidos:

$$U_1 \triangleq \overline{out}.T$$

$$V_1 \triangleq in.R$$

$$U_2 \triangleq \overline{out}.\overline{out}.\overline{out}.T$$

$$V_2 \triangleq in.in.in.R$$

Prove ou refute a igualdade

$$U_2 \triangleright V_2 = U_1 \triangleright V_1$$

- (b) Mostre ou refute a associatividade do operador \triangleright relativamente à igualdade entre processos, *i.e.*, para todo o $P, T, V \in \mathbb{P}$,

$$(U \triangleright V) \triangleright T = U \triangleright (V \triangleright T)$$

- (c) Mostre que $\mathbf{0} \triangleright \mathbf{0} = \mathbf{0}$.

3. Reporte-se, agora, à linguagem ORC que estudou como exemplo de uma linguagem de descrição de arquitecturas de software por coordenação exógena de componentes.
 - (a) Explique a noção de *coordenação exógena* e discuta o modo como ela é implementada em ORC.
 - (b) Esboce na linguagem ORC os elementos fundamentais do estilo *pipe & filter*.
-