$\boxed{\text{Przykład: ZARZĄDCA}}$

Nieformalna specyfikacja

$$Proces_i \stackrel{\mathrm{def}}{=} \overline{a_i}.\ \ldots\ \text{zadanie}\ \ldots\ .\overline{b_i}.\ \ldots\ Proces_i \quad (1 \leq i \leq n)$$

Zarządca powinien udostępniać:

1. zdarzenia a_i cyklicznie, zaczynając od a_1,
2. dla każdego i, zdarzenia a_i i b_i naprzemiennie,
3. **(opcjonalnie)** dostępne powinny być wszystkie sekwencje spełniające 1 i 2.

Formalna specyfikacja

$SZ(i, X)$ oznacza stan t. że:

- kolej na ruch a_i,
- $j \in X \iff$ proces j-ty wykonuje właśnie zadanie.

$$\begin{array}{lcl} SZ(i,X) & \stackrel{\mathrm{def}}{=} & \mathbf{if}\ \ i \in X\ \mathbf{then}\ \ \sum_{j \in X} b_j.SZ(i, X \setminus \{j\}) \\ & & \mathbf{else}\ \ (\ a_i.SZ(i \oplus 1, X \cup \{i\})\ +\ \sum_{j \in X} SZ(i, X \setminus \{j\})\ ) \end{array}$$

Specyfikacja: $SZ(1, \emptyset)$.

Implementacja

$$\begin{array}{lcl} A & \stackrel{\mathrm{def}}{=} & a.c.b.d.A \\ A_i & \stackrel{\mathrm{def}}{=} & A[f_i], \quad f_i = (a_i/a, b_i/b, c_i/c, c_{i \ominus 1}/d) \end{array}$$

$$\begin{array}{lcl} Z & \stackrel{\mathrm{def}}{=} & (A_1 \mid c_1.A_2 \mid\ \ldots\ \mid c_{n-1}.A_n) \setminus \tilde{c}, \quad \tilde{c} = \{c_1, c_2, \ldots, c_n\} \\ & \stackrel{\mathrm{def}}{=} & (A_1 \mid \prod_{j \neq 1} c_{j \ominus 1}.A_j) \setminus \tilde{c} \end{array}$$

Dowód

$$Z(i,X) \stackrel{\text{def}}{=} \begin{array}{lllll} (A_i & \mid \prod_{j \notin X \cup \{i\}} & c_{j\ominus 1}.A_j & \mid \prod_{j \in X} & b_j.c_{j\ominus 1}.A_j \ ) \setminus \tilde{c}, \quad \text{o ile } i \notin X \\ (& \prod_{j \notin X} & c_{j\ominus 1}.A_j & \mid \prod_{j \in X} & b_j.c_{j\ominus 1}.A_j \ ) \setminus \tilde{c}, \quad \text{o ile } i \in X \end{array}$$

W szczególności $Z = Z(1, \emptyset)$. Dla pokazania $Z = SZ(1, \emptyset)$ wystarczy pokazać, że relacja

$$\{(SZ(i,X), Z(i,X) : 1 \leq i \leq n, X \subseteq \{1,2,\ldots,n\}\} \tag{1}$$

jest słabą bisymulacją z dokł. do $\approx$.[1] Niestety tak nie jest! Uważny Czytelnik łatwo odkryje scenariusz, przy którym zachowanie naszej implementacji różni się od zachowania specyfikacji. Trzeba poprawić implementację, a dokładniej trzeba zmodyfikować proces A.

[1] W rzeczywistości trzeba pokazać więcej: że relacja (1) jest pre-bisymulacją z dokładnością do $\lessapprox$.

Poprawka

Jedyny problem to uparte oczekiwanie przez A na b *zanim* A będzie gotowy do odbioru d. Poprawiamy więc tak:

$$A \stackrel{\text{def}}{=} a.c.(b.d.A \;+\; d.b.A)$$

a dla wygody nawet tak:

$$\begin{array}{lclclcl}
A & \stackrel{\text{def}}{=} & a.c.E & \qquad & B & \stackrel{\text{def}}{=} & b.A \\
E & \stackrel{\text{def}}{=} & (b.D \;+\; d.B) & & D & \stackrel{\text{def}}{=} & d.A \\
\\
A_i & \stackrel{\text{def}}{=} & A[f_i] & & B_i & \stackrel{\text{def}}{=} & B[f_i] \\
E_i & \stackrel{\text{def}}{=} & E[f_i] & & D_i & \stackrel{\text{def}}{=} & D[f_i] \\
\\
Z & \stackrel{\text{def}}{=} & (A_1 \mid \prod_{j\neq 1} D_j) \setminus \tilde{c} & & & &
\end{array}$$

Procesy $A_i, E_i, D_i B_i$ odpowiadają następującym stanom:

	proces i gotowy	proces i niegotowy
kolej na proces i	A_i	B_i
kolej na proces $\neq i$	D_i	E_i

Procesy $Z(i,X)$ zdefiniujemy teraz następująco:

$$Z(i,X) \stackrel{\text{def}}{=} \begin{array}{llllll}
(A_i & \mid \prod_{j\notin X\cup\{i\}} & D_j & \mid \prod_{j\in X} & E_j &) \setminus \tilde{c}, \quad \text{o ile } i \notin X \\
(B_i & \mid \prod_{j\notin X} & D_j & \mid \prod_{j\in X\setminus\{i\}} & E_j &) \setminus \tilde{c}, \quad \text{o ile } i \in X
\end{array}$$

Pozostawiamy Czytelnikowi pokazanie, że relacja (1) jest pre-bisymulacją z dokładnością do $\lessapprox$.

Wskazówka

Pomocne może okazać się pokazanie dwóch prostych tożsamości, odpowiadających za wykonanie nieobserwowalnej komunikacji pomiędzy sąsiednimi procesami A_i:

$$(c_i.E_i \mid D_{i\oplus 1}) \setminus c_i \;\approx\; (E_i \mid A_{i\oplus 1}) \setminus c_i$$
$$(c_i.E_i \mid E_{i\oplus 1}) \setminus c_i \;\approx\; (E_i \mid B_{i\oplus 1}) \setminus c_i$$