7 Nierozstrzygalność równoważności procesów

W tym rozdziale ograniczamy nieco język, pozwalając tylko na skończoną sumę; równoważnie mówiąc, mamy tylko binarną sumę $_ + _$. Pozwalamy również tylko na skończoną liczbę stałych procesowych. Dzięki temu, wyrażenia procesowe są obiektami skończonymi. Zakładamy też, że zbiór nazw mogących występować w wyrażeniach procesowym jest skończony. Język tak ograniczony nazwiemy $\mathcal{E}_{\text{fin}}$.

Mimo ograniczenia, równoważność obserwacyjna (wraz z wszystkimi równoważnościami bisymulacyjnymi, które poznaliśmy dotychczas) jest niestety nierozstrzygalna dla $\mathcal{E}_{\text{fin}}$! Pierwsza część tego rozdziału zawiera dowód nierozstrzygalności. Równoważność obserwacyjna staje się rozstrzygalna dla pewnych fragmentów języka $\mathcal{E}_{\text{fin}}$, o czym wspomnimy następnie.

7.1 Dowód nierozstrzygalności

Zaczniemy od wprowadzenia dobrze znanego problemu nierozstrzygalnego, który następnie zredukujemy do problemu równoważności procesów.

Definicja 54 *Maszyną 2-licznikową nazywamy program składający się z m instrukcji, etykietowanych $l_1, \ldots, l_m$:*

$$\begin{aligned} l_1 &: \texttt{instr}_1 \\ &\vdots \\ l_{m-1} &: \texttt{instr}_{m-1} \\ l_m &: \texttt{halt} \end{aligned}$$

Instrukcje $\texttt{instr}_1 \ldots \texttt{instr}_{m-1}$ *operują na dwóch zmiennych (licznikach)* c_1, c_2 *o wartościach naturalnych. Każda z nich należy do jednego z dwóch typów:*

$$\textit{typ } I: \quad c_j := c_j + 1;\ \texttt{goto}\ l_k$$

$$\begin{aligned} \textit{typ } II: \quad & \texttt{if}\ c_j = 0\ \texttt{then goto}\ l_{k_1} \\ & \texttt{else}\ c_j := c_j - 1;\ \texttt{goto}\ l_{k_2} \end{aligned}$$

Działanie programu zaczyna się od instrukcji $l_1 : \texttt{instr}_1$*, przy wartości obydwu zmiennych równych zero:* $c_1 = 0, c_2 = 0$*. Mówimy, że maszyna zatrzymuje się, jeśli sterowanie znajdzie się po pewnej liczbie kroków przy instrukcji* $l_m : \texttt{halt}$*.*

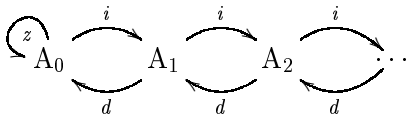
Definicja 55 *Problem stopu dla maszyny 2-licznikowej definiuje się nastepująco: dla danej maszyny 2-licznikowej M stwierdzić, czy maszyna ta się zatrzymuje.*

Twierdzenie 56 *Problem stopu dla maszyny* 2*-licznikowej jest nierozstrzygalny.*

Zajmiemy się teraz redukcją powyższego problemu do następującego: dla danych dwóch procesów P i Q stwierdzić, czy $P = Q$? W tym celu musimy zasymulować wystarczająco wiernie działanie maszyny M w procesie P lub Q lub obydwu. Skończone sterowanie nie nastręcza żadnego kłopotu, natomiast cała trudność jest w symulacji działania licznika. Spójrzmy na narzucające się rozwiązanie (i oznacza inkrementację, d dekrementację a z test na zero):

$$\begin{aligned} \mathrm{A}_0 &\stackrel{\text{def}}{=} i.\mathrm{A}_1 + z.\mathrm{A}_0 \\ \mathrm{A}_1 &\stackrel{\text{def}}{=} i.\mathrm{A}_2 + d.\mathrm{A}_0 \\ &\vdots \\ \mathrm{A}_{n+1} &\stackrel{\text{def}}{=} i.\mathrm{A}_{n+2} + d.\mathrm{A}_n \\ &\vdots \end{aligned}$$

Graf tranzycyjny procesu A_0 wygląda następująco:



(25)

Oczywiście jest ono nieakceptowalne, ponieważ nie należy ono do naszego języka $\mathcal{E}_{\text{fin}}$, skoro wymaga nieskończonej liczby stałych procesowych.

Drogi Czytelniku. Zanim przeczytasz ciąg dalszy, spróbuj rozwiązać następujące bardzo ciekawe i pouczające ćwiczenie:

Ćwiczenie 79 *Znajdź proces symulujący działanie licznika (tzn. proces o grafie tranzycyjnym równoważnym obserwacyjnie grafowi (25)) , zdefiniowany przy pomocy skończonej liczby stałych procesowych. (Wystarczą* 3 *stałe procesowe.) Wskazówka: proces*

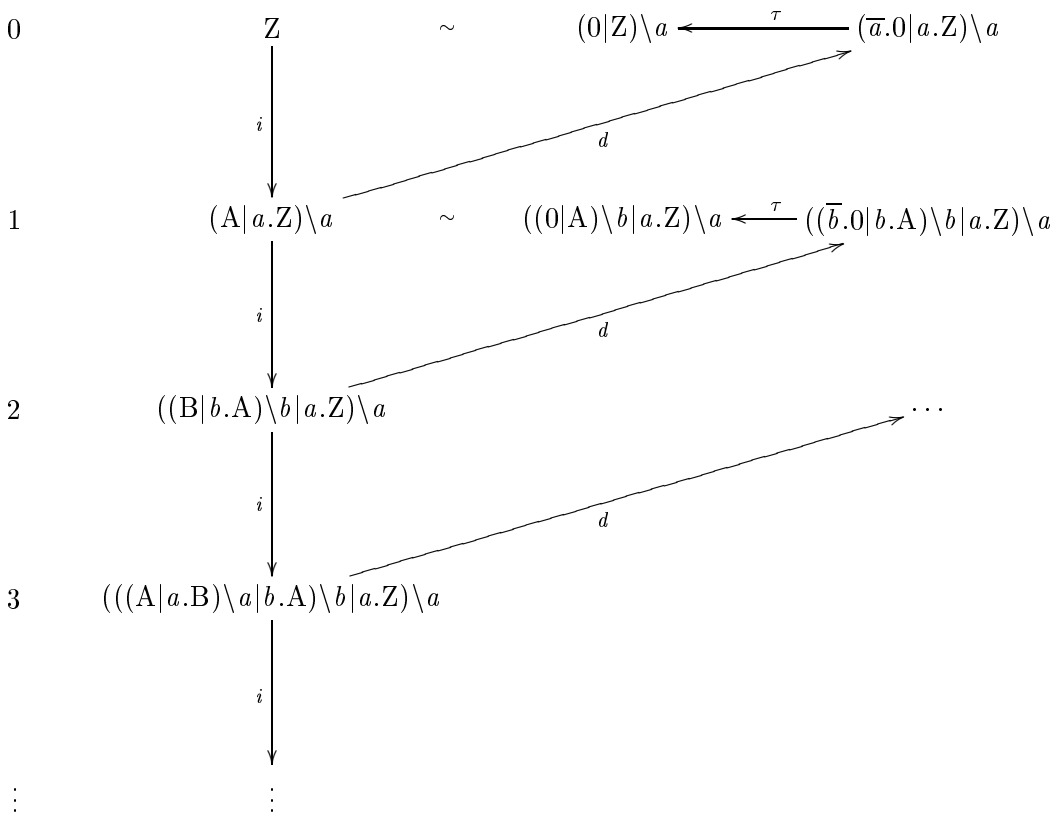
$$\mathrm{B} \stackrel{def}{=} i.(\mathrm{B} | d.0)$$

jest już prawie *poprawnym rozwiązaniem. Brakuje tylko testu na zero ...*

Oto rozwiązanie:

$$\begin{aligned} \mathrm{Z} &\stackrel{\mathrm{def}}{=} z.\mathrm{Z} + i.(\mathrm{A}|a.\mathrm{Z})\backslash a \\ \mathrm{A} &\stackrel{\mathrm{def}}{=} d.\overline{a}.0 + i.(\mathrm{B}|b.\mathrm{A})\backslash b \\ \mathrm{B} &\stackrel{\mathrm{def}}{=} d.\overline{b}.0 + i.(\mathrm{A}|a.\mathrm{B})\backslash a \end{aligned}$$

Graf tranzycyjny procesu Z wygląda następująco:



Wartości licznika równej 0 odpowiada proces Z. Następnie każda inkrementacja polega na zastąpieniu aktywnego procesu Z, A lub B przez większe podwyrażenie. Formalnie, zdefiniujmy wyrażenia procesowe $E_0, E_1, \ldots$ z jedną zmienna wolną X

jak następuje:

$$\begin{aligned} E_0 &\equiv \mathrm{Z} \\ E_1 &\equiv (X|a.\mathrm{Z})\backslash a \\ E_{n+1} &\equiv \begin{cases} E_n\{\ (X|b.\mathrm{A})\backslash b\ /\ X\ \}, & \text{gdy } n \text{ nieparzyste} \\ E_n\{\ (X|a.\mathrm{B})\backslash a\ /\ X\ \}, & \text{gdy } n \text{ parzyste} \end{cases} \end{aligned}$$

Ćwiczenie 80 *Pokaż, że relacja*

$$\{(\mathrm{A}_n, E_n\{\mathrm{B}/X\}) : n \textit{ parzyste}\} \cup \{(\mathrm{A}_n, E_n\{\mathrm{A}/X\}) : n \textit{ nieparzyste}\}$$

jest pre-bisymulacją z dokładnością do $\lessapprox$.

Twierdzenie 57 *Nierozstrzygalny jest następujący problem: dla danych dwóch procesów P i Q stwierdzić, czy $P \sim Q$?*

Dowód: Redukcja problemu stopu dla maszyny 2-licznikowej. Dla danej maszyny 2-licznikowej M zdefiniujemy parę procesów P_M, Q_M taką, że maszyna M zatrzymuje się dokładnie wtedy i tylko wtedy, gdy $P_M \nsim Q_M$. Proces P_M składa się z trzech komponentów:

$$P_M \equiv (\mathrm{Z}[z_1/z, i_1/i, d_1/d] \mid \mathrm{Z}[z_2/z, i_2/i, d_2/d] \mid \mathrm{S}_1)\backslash\{z_j, i_j, d_j : j = 1, 2\}.$$

Pierwsze dwa to dwa liczniki o wartości 0. Komponent S_1 odpowiada za skończone sterowanie: stała procesowa S_i, dla $i = 1, \ldots, m$, będzie oznaczać, że maszyna znajduje się przy instrukcji l_i : `instr`$_i$. Stałe S_i, dla $i = 1, \ldots, m-1$, definiujemy zależnie od tego, do którego typu należy instrukcja `instr`$_i$. Dla typu I:

$$\mathrm{S}_i \stackrel{\text{def}}{=} \overline{i}_j.\mathrm{S}_k$$

a dla typu II:

$$\mathrm{S}_i \stackrel{\text{def}}{=} \overline{z}_j.\mathrm{S}_{k_1} + \overline{d}_j.\mathrm{S}_{k_2}\ .$$

Natomiast dla $i = m$ mamy

$$\mathrm{S}_m \stackrel{\text{def}}{=} 0.$$

Pozostaje tylko wskazać drugi proces Q_M. Ale ponieważ w P_M zawarliśmy już pełną informację o działaniu M, w Q_M musimy tylko uwzględnić fakt, że pytamy o to, czy M się zatrzymuje. Zatem niech

$$Q_M \equiv \mathrm{T}, \quad \text{gdzie } \mathrm{T} \stackrel{\text{def}}{=} \tau.\mathrm{T}.$$

Zauważmy, że obydwa procesy P_M i Q_M są zdeterminowane w następującym sensie: każdy potomek P' każdego z nich ma co najwyżej jedną tranzycję. Ponadto $\mathcal{L}(P_M) = \mathcal{L}(Q_M) = \emptyset$, tzn. wykonują one tylko zdarzenie τ.

Załóżmy, że maszyna M zatrzymuje się po skończonej liczbie kroków. Wtedy dla pewnego N mamy $P_M(\stackrel{\tau}{\longrightarrow})^N P'$ i $P' \stackrel{\tau}{\not\rightarrow}$. Czyli P' nie ma odpowiedzi na $\mathrm{T} \stackrel{\tau}{\longrightarrow} \mathrm{T}$.

Załóżmy przeciwnie, że maszyna M nie zatrzymuje się. Wtedy istnieje silna bisymulacja pomiędzy P_M i Q_M, której znalezienie pozostawiamy Czytelnikowi. □

Pokazaliśmy nierozstrzygalność silnej równoważności. Natychmiastowym wnioskiem jest nierozstrzygalność wszystkich innych poznanych dotychczas równoważności:

Wniosek 58 *Nierozstrzygalne są również następujące dwa problem:*

- *dla danych dwóch procesów P i Q stwierdzić, czy $P \approx Q$?*
- *dla danych dwóch procesów P i Q stwierdzić, czy $P = Q$?*

Ćwiczenie 81 *Udowodnij wniosek 58.*

Podsumujmy naszą wiedzę na temat (nie)rozstrzygalności słabej i silnej równoważności dla $\mathcal{E}_{\mathrm{fin}}$ i pewnych jego fragmentów:

język	$\sim$	$\approx$
$\mathcal{E}_{\mathrm{fin}}$ bez przemianowania $_[f]$	nierozstrzygalna	nierozstrzygalna
$BPP : \mathcal{E}_{\mathrm{fin}}$ bez przemianowania $_[f]$ i zakazu $_\backslash L$	rozstrzygalna PSPACE-trudna	?
procesy skończeniestanowe: 0, $\alpha._$, $_ + _$, rekurencja	P	p

7.2 Kilka ćwiczeń

Ćwiczenie 82 *Pokaż, że $P|\tau.Q = \tau.P|Q$.*

Ćwiczenie 83 *Co by się zmieniło, gdyby definicję słabej równoważności bisymulacyjnej oprzeć na nieznacznie zmodyfikowanej definicji słabej bisymulacji:*

Definicja 59 *Relację binarną R pomiędzy procesami nazywamy* słabą bisymulacją, *jeśli dla każdej pary $(P, Q) \in R$, $\forall \alpha \in \mathrm{Act}$,*

$$(i) \quad P \stackrel{\alpha}{\longrightarrow} P' \implies \exists Q' \quad t.że \quad Q \stackrel{\alpha}{\Longrightarrow} Q' \quad i \quad (P', Q') \in R$$

$$(ii) \quad Q \stackrel{\alpha}{\longrightarrow} Q' \implies \exists P' \quad t.że \quad P \stackrel{\alpha}{\Longrightarrow} P' \quad i \quad (P', Q') \in R.$$

Jedyna różnica to zastąpienie relacji $\stackrel{\widehat{\alpha}}{\Longrightarrow}$ *przez* $\stackrel{\alpha}{\Longrightarrow}$. *Czyli nie pozwalamy na pustą odpowiedź w przypadku* $\alpha = \tau$.

Ćwiczenie 84 *Pokaż, że powyższa modyfikacja pociąga następującą własność:*

$$P \approx Q,\ P \stackrel{\tau}{\longrightarrow} P' \implies \exists Q'\ \ t.że\ \ Q \stackrel{\tau}{\longrightarrow} Q'\ \ i\ \ P' \approx Q',$$

przy założeniu, że P *i* Q *mają skończenie wiele* τ*-potomków, tzn. zbiór* $\{P' : P \stackrel{\tau}{\longrightarrow} P'\}$ *jest skończony i podobnie dla* Q.

Czy zatem konsekwencją definicji 59 jest kolaps $\approx\ =\ \sim$?

Ćwiczenie 85 *Pokaż, że wciąż* $\approx\ \neq\ \sim$.

Rozważmy teraz inną możliwą modyfikację definicji słabej bisymulacji. Jak pamiętamy z rozdziału 5 (por. ćwiczenie 66), definicję 35 można usymetrycznić następująco:

Definicja 60 *Relację binarną* R *pomiędzy procesami nazywamy silną bisymulacją, jeśli dla każdej pary* $(P, Q) \in R$, $\forall t \in \mathcal{L}^*$,

$$(i)\ \ P \stackrel{t}{\Longrightarrow} P' \implies \exists Q'\ \ t.że\ \ Q \stackrel{t}{\Longrightarrow} Q'\ \ i\ \ (P', Q') \in R$$
$$(ii)\ \ Q \stackrel{t}{\Longrightarrow} Q' \implies \exists P'\ \ t.że\ \ P \stackrel{t}{\Longrightarrow} P'\ \ i\ \ (P', Q') \in R.$$

Przypomnijmy, ze $\mathcal{L} = \mathrm{Act} \setminus \{\tau\}$. Definicja 60 definiuje dokładnie te same relacje co definicja 35 (por. ćwiczenie 66).

Zastanówmy się nad modyfikacją, polegającą na rozpatrzeniu tylko niepustych ciągów t.

Definicja 61 *Relację binarną* R *pomiędzy procesami nazywamy silną bisymulacją, jeśli dla każdej pary* $(P, Q) \in R$, $\forall t \in \mathcal{L}^+$,

$$(i)\ \ P \stackrel{t}{\Longrightarrow} P' \implies \exists Q'\ \ t.że\ \ Q \stackrel{t}{\Longrightarrow} Q'\ \ i\ \ (P', Q') \in R$$
$$(ii)\ \ Q \stackrel{t}{\Longrightarrow} Q' \implies \exists P'\ \ t.że\ \ P \stackrel{t}{\Longrightarrow} P'\ \ i\ \ (P', Q') \in R.$$

mamy na przykład: $a.0 + \tau.0 \approx a.0$.

Ćwiczenie 86 *Pokaż, że* $\approx$ *nie jest teraz kongruencją względem prefiksu i względem złożenia równoległego.*

Jak widać, znów konsekwencje zmian są miażdżące. Wniosek: pojęcie słabej bisymulacji nie jest ad hoc, jest starannie wybrane, a niektóre narzucające się alternatywy do niczego się nie nadają.

7.2.1 Symulacja

Definicja 62 *Relację binarną R pomiędzy procesami nazywamy silną symulacją, jeśli dla każdej pary $(P,Q) \in R$, $\forall \alpha \in \mathrm{Act}$,*

$$(i) \quad P \xrightarrow{\alpha} P' \implies \exists Q' \quad t.że \quad Q \xrightarrow{\alpha} Q' \quad i \quad (P',Q') \in R.$$

Niech $\precsim$ oznacza sumę wszystkich silnych symulacji:

$$\precsim \; = \bigcup_{R \;\; silna \; symulacja} R$$

Ćwiczenie 87 *Pokaż, że $\precsim$ jest pre-porządkiem, tzn. jest zwrotna i przechodnia, oraz że jest pre-kongruencją, tzn. jest zachowywana przez wszystkie operatory.*

Niech $\simeq$ oznacza $\precsim \cap \precsim^{-1}$. Relację $\simeq$ nazywamy *silną rownoważnością symulacyjną.*

Ćwiczenie 88 *Pokaż, że $\simeq$ jest kongruencją.*

Ćwiczenie 89 *Pokaż, że $\sim \subsetneq \simeq$. Tzn. wskaż parę procesów, między którymi są dwie symulacje (w dwie strony), ale nie ma bisymulacji.*

Ćwiczenie 90 *Udowodnij, że R jest silną bisymulacją wtw. gdy R oraz R^{-1} są silnymi symulacjami.*

Definicja 63 *Relację binarną R pomiędzy procesami nazywamy słabą symulacją, jeśli dla każdej pary $(P,Q) \in R$, $\forall \alpha \in \mathrm{Act}$,*

$$(i) \quad P \xrightarrow{\alpha} P' \implies \exists Q' \quad t.że \quad Q \stackrel{\widehat{\alpha}}{\Longrightarrow} Q' \quad i \quad (P',Q') \in R.$$

Niech $\precapprox$ oznacza sumę wszystkich słabych symulacji:

$$\precapprox \; = \bigcup_{R \;\; słaba \; symulacja} R$$

Ćwiczenie 91 *Czy własność wspomniana w ćwiczeniu 90 zachodzi też dla słabej bisymulacji i symulacji?*

Relację *słabej rownoważności symulacyjnej* $\approx$ definiujemy jako $\precapprox \cap \precapprox^{-1}$.

Ćwiczenie 92 *Pokaż, że $\precapprox$ jest pre-kongruencją (a zatem $\approx$ jest kongruencją).*

Ćwiczenie 93 *Rozważ następujący przykład:*

$$\begin{aligned}
\mathrm{A} &\stackrel{def}{=} a.b.\mathrm{A} \\
\mathrm{S} &\stackrel{def}{=} \overline{P}.\overline{V}.\mathrm{S} \\
\mathrm{B} &\stackrel{def}{=} a.P.b.V.\mathrm{B} \\
\mathrm{A}_i &\stackrel{def}{=} \mathrm{A}[f_i] && i = 1,2 \\
\mathrm{B}_i &\stackrel{def}{=} \mathrm{B}[f_i] && i = 1,2,
\end{aligned}$$

gdzie $f_i(a) = a_i, f_i(b) = b_i$, *dla* $i = 1, 2$. *Pokaż, że* $\mathrm{A}_1|\mathrm{A}_2 \not\approx (\mathrm{B}_1|\mathrm{B}_2|\mathrm{S})\backslash\{P, V\}$, *ale* $\mathrm{A}_1|\mathrm{A}_2 \cong (\mathrm{B}_1|\mathrm{B}_2|\mathrm{S})\backslash\{P, V\}$.

Zatem $\approx \subsetneq \cong$.

Ćwiczenie 94 *Czy słaba i silna równoważność symulacyjna jest rozstrzygalna?*