Rownoważność systemów nieskończenie stanowych z przejściami epsilonowymi

Piotr Hofman

22 lutego 2014

Autoreferat rozprawy doktorskiej

Spis treści

1 Wstęp

W ostatnich latach obserwuje się nieustanny wzrost zainteresowania formalną analizą systemów komputerowych. Typowe podejście składa się z dwóch etapów. Najpierw budowany jest abstrakcyjny model systemu, a następnie testowane są interesujące nas własności modelu. Dla wydajności tego podejścia kluczowa jest wielkość modelu, a głównym problemem jest efekt eksplozji przestrzeni stanów. Jednym ze sposobów radzenia sobie z tą trudnością jest zamiana modelu na mniejszy i równoważny. To podejście daje motywację do badań nad procedurami sprawdzania równoważności systemów.

Często okazuje się, że mały model skończenie stanowy nie pozwala na uchwycenie istotnych cech systemu. Wynika to z tego, iż wiele zachowań komputerów w swej istocie ma naturę nieskończoną. Przykładami mogą być dowolna głębokość rekursji lub brak ograniczenia na ilość wykonywanych równocześnie wątków programu. Czasami w takich sytuacjach sensowne okazuje się podejście od zupełnie innej strony. Zamiast małego modelu skończenie stanowego decydujemy się na model opisujący nieskończoną przestrzeń stanów. W takim przypadku analizę przeprowadzamy w sposób symboliczny, korzystając ze skończonego opisu przestrzeni stanów. W przypadku analizy symbolicznej jeszcze bardziej istotne staje się zrozumienie symetrii, a więc i równoważności w przestrzeni stanów generowanej przez skończony opis, co znowu motywuje badania równoważności.

W rozprawie są rozważane dwa formalizmy generujące przestrzenie nieskończenie stanowe. Dokładniej: przemienne gramatyki bezkontekstowe i automaty z jednym licznikiem bez testu zera. Pierwszy z formalizmów jest szczególnym przypadkiem sieci Petriego, podczas gdy drugi jest zarówno podklasą sieci, jak i szczególnym przypadkiem automatu ze stosem. W obu formalizmach dopuszczamy ε-przejścia, nazywane też przejściami nieobserwowalnymi. Skupimy się na badaniu równoważności dla systemów opisanych tymi formalizmami, mianowicie równoważności bisymulacyjnej i symulacyjnej. To połączenie jest wynikiem tego, iż przy analizie obu równoważności przydatne są podobne techniki oraz faktu, że inne istotne równoważności zostały wcześniej zbadane.

Dotychczasowe badania nad systemami z ε-przejściami wskazują na to, iż brakuje nam narzędzi i technik, które pozwalałyby na ich analizę. Większość rezultatów dotyczy nierozstrzygalności wybranych problemów w formalizmach dopuszczających ε-przejścia. Znanych jest zaskakująco mało efektywnych algorytmów.

Dwa główne wyniki rozprawy to rozstrzygalność bisymulacji rozgałęziającej (ang. branching bisimulation) dla unormowanych przemiennych gramatyk bezkontekstowych i rozstrzygalność słabej symulacji dla automatów z jednym licznikiem bez testu zera. Ponadto, w rozprawie została przedstawiona analiza możliwości wybranych technik aproksymacji dla problemu słabej bisymulacji dla przemiennych gramatyk bezkontekstowych.

W dalszej części autoreferatu zostaną przedstawione podstawowe pojęcia, najważniejsze wyniki pracy oraz krótkie studium literaturowe.

2 Podstawowe definicje

Niech $\mathbb{A}ct$ będzie skończonym alfabetem etykietującym tranzycje. Dodatkowo tranzycje mogą być etykietowane literą $\varepsilon \notin \mathbb{A}ct$ oznaczająca przejścia nieobserwowalne. Przez $\mathbb{A}ct_\varepsilon$ oznaczamy $\mathbb{A}ct \cup \{\varepsilon\}$.

Definicja 1. Etykietowany system tranzycyjny *(lub* LTS*) nad skończonym alfabetem $\mathbb{A}ct$ jest krotką $L = (U, E)$. Składa się z potencjalnie nieskończonego zbioru wierzchołków U i zbioru skierowanych krawędzi $E \subseteq U \times \mathbb{A}ct_\varepsilon \times U$ etykietowanych elementami $\mathbb{A}ct_\varepsilon$.*

Zauważmy, że powyższa definicja umożliwia istnienie krawędzi równoległych oraz pętli. Będziemy używać notacji $v \xrightarrow{\zeta} v'$ zamiast $(v, \zeta, v') \in E$. Dla wierzchołków LTS używamy też terminu „proces", a krawędzie będziemy nazywali „tranzycjami". Rozważane LTS-y będą indukowane przez przemienne gramatyki bezkontekstowe lub automaty z jednym licznikiem. Dla wygody będziemy zakładać, że we wszystkich rozważnych LTS-ach każdy proces ma dodatkowe ε-przejście będące pętlą, to znaczy $\alpha \xrightarrow{\varepsilon} \alpha$. W szczególności istnieje przejście $\varepsilon \xrightarrow{\varepsilon} \varepsilon$.

Nieobserwowalne tranzycje $\xrightarrow{\varepsilon}$ będą oznaczane przez $\longrightarrow$, a ich domknięcie przechodnie będzie oznaczane $\Longrightarrow$. Zatem $\alpha \Longrightarrow \beta$ zachodzi, jeśli proces β może być osiągnięty z α przy pomocy sekwencji $\xrightarrow{\varepsilon}$-tranzycji.

Przemienne gramatyki bezkontekstowe. Skupiamy się tylko na gramatykach w postaci normalnej Greibach.

Definicja 2. *Przemienna gramatyka bezkontekstowa składa się ze skończonego alfabetu $\mathbb{A}ct$ i skończonego zbioru nieterminali V oraz skończonego zbioru reguł postaci $X \xrightarrow{\zeta} \alpha$, gdzie $X \in V$, $\zeta \in \mathbb{A}ct_\varepsilon$, a α jest skończonym multizbiorem nieterminali.*

Procesem nazwiemy każdy skończony multizbiór nieterminali. Proces zatem może być rozumiany jako złożenie równoległe pewnej liczby nieterminali. Dla ustalenia uwagi pusty proces oznaczamy ε. Przez $\alpha||\beta$ oznaczamy złożenie procesów α i β, rozumiane jako suma multizbiorów.

Gramatyka indukuje LTS, którego wierzchołkami są skończone multizbiory nieterminali. Dla każdej reguły $X \xrightarrow{\zeta} \alpha$ i każdego procesu β, LTS zawiera tranzycję

$$\beta||X \xrightarrow{\zeta} \alpha||\beta.$$

Przykład 1. *Jako przykład rozważmy następującą gramatykę:*

$$\begin{array}{lllllllllll}
P & \longrightarrow & \varepsilon & \quad P & \longrightarrow & P||A & \quad P & \xrightarrow{b} & \varepsilon & \quad P & \longrightarrow & Q \\
Q & \longrightarrow & \varepsilon & \quad Q & \longrightarrow & Q||A & \quad A & \xrightarrow{a} & \varepsilon & \quad A & \longrightarrow & \varepsilon
\end{array}$$

Indukowany LTS zawiera miedzy innymi następujący ciąg tranzycji:

$$Q||P \longrightarrow Q||P||A \longrightarrow Q||P||A||A \longrightarrow P||A||A \xrightarrow{b} A||A.$$

Dodatkowo, jeśli przez A^k rozumiemy k-krotne równoległe złożenie A, to dla każdego k, istnieje ciąg tranzycji $Q||P \Longrightarrow P||A^k \xrightarrow{b} A^k$.

Automaty z jednym licznikiem. W rozprawie rozważamy tylko automaty licznikowe *bez* testu zera.

Definicja 3. *Automat z jednym licznikiem składa się ze skończonego alfabetu $\mathbb{A}ct$, skończonego zbioru stanów Q i skończonego zbioru tranzycji. Każda tranzycja jest postaci $q \xrightarrow{\zeta,d} p$, gdzie $p, q \in Q$, $\zeta \in \mathbb{A}ct_\varepsilon$, a $d \in \{-1, 0, 1\}$ oznacza zmianę wartości licznika.*

Automat indukuje LTS, którego procesy są postaci $qm \in Q \times \mathbb{N}$. Dla każdego przejścia w automacie $q \xrightarrow{\zeta,d} p$ i każdej liczby $m \in \mathbb{N}$ takiej, że $m + d \geqslant 0$, w LTS-ie mamy tranzycję

$$qm \xrightarrow{\zeta} p(m+d).$$

Przykład 2. *Jako przykład weźmy automat nad alfabetem $\mathbb{A}ct = \{a, b\}$, który ma tylko jeden stan i następujące przejścia*

$$p \quad \xrightarrow{a,1} \quad p \qquad\qquad p \quad \xrightarrow{b,-1} \quad p \qquad\qquad p \quad \xrightarrow{-1} \quad p$$

Indukowany LTS zawiera między innymi następującą sekwencję tranzycji:

$$p0 \xrightarrow{a} p1 \xrightarrow{a} p2 \xrightarrow{b} p1 \xrightarrow{a} p2 \longrightarrow p1 \longrightarrow p0.$$

Intuicyjnie, *unormowanie* to zdolność do osiągnięcia procesu pustego (lub wyzerowania wartości licznika).

Definicja 4. *Przemienna gramatyka bezkontekstowa jest unormowana, jeśli dla każdego procesu istnieje sekwencja tranzycji prowadząca do procesu pustego.*[1]

System w przykładzie 1 jest unormowany. Dla automatów z licznikiem nie będziemy potrzebować unormowania.

2.1 Symulacja i bisymulacja

Niech $\alpha \overset{\zeta}{\Longrightarrow} \beta$ oznacza, że $\alpha \Longrightarrow \gamma \xrightarrow{\zeta} \gamma' \Longrightarrow \beta$ dla pewnych procesów γ, γ'. W szczególności, jeśli $\zeta = \varepsilon$, tranzycja $\overset{\varepsilon}{\Longrightarrow}$ oznacza sekwencję ε-tranzycji.

[1] Innymi słowy gramatyka jest unormowana, jeśli każdy proces generuje jakieś słowo. Zauważmy, że unormowanie jako ograniczenie jest nieistotne z punktu widzenia rozpoznawanego języka, ale ma znaczenie dla struktury LTS-u.

Równoważność bisymulacyjna, jak również quasi-porządek symulacyjny, pojawiają się w wielu wariantach. W rozprawie skupiamy się na dwóch najbardziej znanych, mianowicie na słabej i rozgałęziającej (bi)symulacji, które to różnią się od siebie różnym traktowaniem ε-tranzycji. Struktura tych czterech relacji skłania do wydzielenia z ich definicji części zwanej ekspansją.

Definicja 5. *Niech $S \subseteq U \times U$. Para (α, β) procesów spełnia* **ekspansję słabej symulacji** *względem S, jeśli dla każdego $\zeta \in \mathbb{A}ct_\varepsilon$ i każdej tranzycji $\alpha \xrightarrow{\zeta} \alpha'$ istnieje tranzycja $\beta \overset{\zeta}{\Longrightarrow} \beta'$ taka, że $(\alpha', \beta') \in S$.*

Definicja 6. *Niech $B \subseteq U \times U$. Para procesów (α, β) spełnia* **ekspansję słabej bisymulacji** *względem B, jeśli (α, β) spełnia ekspansję słabej symulacji względem B oraz (β, α) spełnia ekspansję słabej symulacji względem B^{-1}.*

Rozwijając powyższą definicję otrzymujemy, że para (α, β) spełnia ekspansję słabej bisymulacji względem B wtw. gdy dla każdego $\zeta \in \mathbb{A}ct_\varepsilon$

- i każdej tranzycji $\alpha \xrightarrow{\zeta} \alpha'$ istnieje $\beta \overset{\zeta}{\Longrightarrow} \beta'$ takie, że $(\alpha', \beta') \in B$,
- i każdej tranzycji $\beta \xrightarrow{\zeta} \beta'$ istnieje $\alpha \overset{\zeta}{\Longrightarrow} \alpha'$ takie, że $(\alpha', \beta') \in B$.

Definicja 7. *$S \subseteq U \times U$ jest* **słabą symulacją***, jeśli dla każdej pary $(\alpha, \beta) \in S$ spełniona jest ekspansja słabej symulacji względem S.* **Słaby quasi-porządek** *to suma teoriomnogościowa wszystkich symulacji.*

Definicja 8. *$B \subseteq U \times U$ jest* **słabą bisymulacją***, jeśli dla każdego $(\alpha, \beta) \in B$ spełniona jest ekspansja słabej bisymulacji względem B.* **Słaba równoważność** *to suma teoriomnogościowa wszystkich relacji słabej bisymulacji.*

Innymi słowy, procesy α i β są w relacji słabego quasi-porządku, jeśli istnieje słaba symulacja zawierająca parę (α, β). Analogicznie procesy α i β są w relacji słabej równoważności, jeśli istnieje słaba bisymulacja zawierająca parę (α, β).

Uwaga 1. *Słaby quasi-porządek jest relacją quasi-porządku, a słaba równoważność jest relacją równoważności.*

W obecności ε-tranzycji, słaba (bi)symulacja nie jest jedynym wyborem. Najbardziej znaną konkurencyjną równoważnością jest relacja bisymulacji rozgałęziającej. Wykorzystując ogólny schemat (bi)symulacji wystarczy, że zdefiniujemy ekspansję dla symulacji rozgałęziającej:

Definicja 9. *Niech $S \subseteq U \times U$. Para procesów (α, β) spełnia ekspansję symulacji rozgałęziającej względem relacji S, jeśli dla każdego $\zeta \in \mathbb{A}ct_\varepsilon$ i każdej tranzycji $\alpha \xrightarrow{\zeta} \alpha'$, istnieje tranzycja $\beta \Longrightarrow \bar{\beta} \xrightarrow{\zeta} \beta'$ taka, że $(\alpha, \bar{\beta}) \in S$ i $(\alpha', \beta') \in S$.*

Podmieniając definicję ekspansji słabej symulacji na definicję ekspansji symulacji rozgałęziającej w definicji 7, otrzymujemy *quasi-porządek*. Co więcej definicja 6 może zostać zaadoptowana dla nowej ekspansji, dając w efekcie *ekspansję bisymulacji rozgałęziającej*.

Definicja 10. *Niech $B \subseteq U \times U$. Para procesów (α, β) spełnia ekspansję bisymulacji rozgałęziającej względem B, jeśli (α, β) spełnia ekspansję symulacji rozgałęziającej względem B oraz (β, α) spełnia ekspansję symulacji rozgałęziającej względem B^{-1}.*

Następnie, *równoważność rozgałęziającą* otrzymujemy przez analogiczną adaptację definicji 8. Co więcej, uwaga 1 zachodzi jeśli zastąpimy słabą (bi)symulację przez rozgałęziający odpowiednik.

Równoważność i quasi-porządek rozgałęziający są co do inkluzji zawarte w swoich słabych odpowiednikach. Możemy zatem relacje inkluzji zobrazować tak jak na poniższym diagramie:

$$\begin{tikzcd}
 & \text{słaby quasi-porządek} & \\
\text{słaba równoważność} \arrow[ur, dotted] & & \text{quasi-porządek rozgałęziający} \arrow[ul, dotted] \\
 & \text{równoważność rozgałęziająca} \arrow[ul, dotted] \arrow[ur, dotted] &
\end{tikzcd}$$

Zauważmy, że dla LTS-ów, w których nie ma ε-tranzycji, relacje słabej i rozgałęziającej (bi)symulacji są odpowiednio tożsame.

Przykład 3. *Jako ilustrację różnic pomiędzy słabą i rozgałęziającą bisymulacją rozważmy następującą gramatykę:*

$$\begin{array}{lllllllll}
A & \longrightarrow & \varepsilon & A & \stackrel{a}{\longrightarrow} & \varepsilon & & & \\
B & \longrightarrow & \varepsilon & B & \longrightarrow & A & B & \stackrel{a}{\longrightarrow} & A
\end{array}$$

oraz dwa procesy B i $A||A$. Jak łatwo zauważyć, oba procesy są w relacji słabej równoważności. Co ciekawe jednak, nie są w relacji równoważności rozgałęziającej. Mianowicie zauważmy, że dla tranzycji

$$B \longrightarrow \varepsilon$$

nie ma pasującej sekwencji zaczynającej się w $A||A$. Jedyny kandydat wydający się być sensownym, a mianowicie $A||A \longrightarrow A \longrightarrow \varepsilon$, nie spełnia warunku ekspansji bisymulacji rozgałęziającej, gdyż proces A nie jest równoważny procesowi B.

2.2 Problemy decyzyjne

Problemy decyzyjne, którymi zajmujemy się w rozprawie, należą do ogólnego schematu, którego parametrem jest klasa C skończenie reprezentowanych etykietowanych systemów tranzycyjnych, oraz pojęcie równoważności $\approx$ (ew. quasi-porządku) pomiędzy procesami:

WEJŚCIE:	LTS z klasy C i dwa procesy α i β.
PYTANIE:	Czy $\alpha \approx \beta$?

Skupiamy się na następujących instancjach powyższego schematu:

- słaba i rozgałęziająca równoważność dla przemiennych gramatyk bezkontekstowych,
- słaby i rozgałęziający quasi-porządek dla automatów z jednym licznikiem bez testu zera.

Rozstrzygalność tych problemów była dotychczas otwarta. Szczegółowe omówienie znanych wyników i ich związek z wynikami tej rozprawy znajduje się w rozdziale 4.

2.3 Aproksymacje

Definicja 11 (Aproksymacje)**.** *Definiujemy rodzinę relacji $R_o \subseteq U \times U$ indeksowaną liczbami porządkowymi o.*

- $R_0 = U \times U$ *zawiera wszystkie pary procesów* (α, β)*;*
- $(\alpha, \beta) \in R_{o+1}$ *wtedy i tylko wtedy gdy* (α, β) *para spełnia ekspansję względem relacji* R_o*;*
- *dla liczby porządkowej granicznej o, definiujemy* $R_o = \bigcap_{n<o} R_n$.

Powyższa definicja nie precyzuje jaka ekspansja powinna być użyta. Jest tak, gdyż definicja ta może być ukonkretniona przy użyciu każdej spośród zdefiniowanych wcześniej ekspansji. W każdym z przypadków aproksymacje tworzą zstępujący łańcuch relacji, który stabilizuje się na odpowiadającej relacji równoważności/quasi-porządku. Jest to bezpośrednia konkluzja z twierdzenia Knastera Tarskiego o punkcie stałym. Formalnie:

Definicja 12. *Dla dowolnej klasy LTS-ów, powiemy, że hierarchia aproksymacji* $(R_o)_o$ **stabilizuje się** *(lub* **zbiega***) na poziomie κ jeśli dla każdego LTS-u z tej klasy zachodzi* $R_\kappa = R_{\kappa+1}$.

W ogólności, hierarchie aproksymacji stabilizują się daleko za pierwszą nieskończoną liczbą porządkową ω. W szczególności, tak właśnie jest dla aproksymacji indukowanych przez ekspansję słabej i rozgałęziającej (bi)symulacji dla LTS-ów indukowanych przez przemienne gramatyki bezkontekstowe, jak i automaty z jednym licznikiem.

Przykład 4. *Dla ilustracji rozważmy raz jeszcze gramatykę z przykładu 3, oraz rodzinę aproksymacji* (R_o) *indukowaną przez ekspansję słabej bisymulacji. Oczywiste jest, że para* (A, ε) *jest w relacji* R_0*, ale nie jest w* R_1*, jako że proces* ε *nie ma wychodzącej* a*-tranzycji. Podobnie para* $(A||A, A)$ *jest w relacji* R_1*, ale nie w relacji* R_2*, jako że jedyną możliwą odpowiedzią na przejście* $A||A \xrightarrow{a} A$ *jest* $A \xrightarrow{a} \varepsilon$*. Ogólnie, jeśli* $n > m$ *wtedy para* (A^n, A^m) *jest w relacji* R_m *ale nie w* R_{m+1}*. Zatem łańcuch nie stabilizuje się na poziomie będącym liczbą naturalną. Co ciekawe, dla tego konkretnego przypadku łańcuch stabilizuje jednak się na poziomie* ω*, to znaczy* R_ω *jest równe słabej równoważności.*

Powodem, dla którego wprowadza się aproksymacje, jest fakt, że często ich obliczanie prowadzi do algorytmu rozstrzygającego aproksymowaną relację. Zwykle oblicza się aproksymacje po kolei, co wiąże się z tym, że jeśli nie stabilizują się na poziomie ω, to nie pozwalają nam one wnioskować o aproksymowanej relacji. Ta obserwacja motywuje próby zdefiniowania aproksymacji zbiegających „szybko".

Aby przedstawić część wyników, potrzebne nam będą dwie nowe hierarchie aproksymacji. Do zdefiniowania jednej z nich musimy najpierw rozszerzyć relację $\stackrel{a}{\Longrightarrow}$ na słowa $w \in \mathbb{A}ct^*$. Niech $\alpha \stackrel{a_1 \ldots a_n}{\Longrightarrow} \beta$, jeśli $\alpha \stackrel{a_1}{\Longrightarrow} \gamma_1 \stackrel{a_2}{\Longrightarrow} \ldots \gamma_{n-1} \stackrel{a_n}{\Longrightarrow} \beta$ dla pewnych procesów $\gamma_1, \ldots, \gamma_{n-1}$.

Definicja 13. *Niech* $S \subseteq U \times U$*. Para procesów* (α, β) *spełnia* **ekspansję długich kroków słabej symulacji** *względem* S*, jeśli dla każdego* $\zeta \in \mathbb{A}ct_\varepsilon$ *i każdego* $\alpha \stackrel{\zeta}{\Longrightarrow} \alpha'$*, istnieje* $\beta \stackrel{\zeta}{\Longrightarrow} \beta'$ *takie, że* $(\alpha', \beta') \in S$*.*

Definicja 14. *Niech* $S \subseteq U \times U$*. Para procesów* (α, β) *spełnia* **słowną ekspansję słabej symulacji** *względem* S*, jeśli dla każdego słowa* $w \in \mathbb{A}ct^*$ *i każdego* $\alpha \stackrel{w}{\Longrightarrow} \alpha'$*, istnieje* $\beta \stackrel{w}{\Longrightarrow} \beta'$ *takie, że* $(\alpha', \beta') \in S$*.*

Jak poprzednio w definicji 6, w naturalny sposób usymetryczniamy powyższe definicje, otrzymując ekspansję długich kroków oraz słowną ekspansję dla słabej bisymulacji. Hierarchie indukowane przez wszystkie cztery nowe ekspansje zbiegają do zdefiniowanych już relacji, konkretnie - dwie pierwsze do słabego quasi-porządku, a dwie kolejne do słabej równoważności.

Jak można się spodziewać, aproksymacje indukowane przez ekspansje słowną zbiegają szybciej, niż te pochodzące od ekspansji długich kroków, które z kolej są szybsze niż aproksymacje indukowane przez ekspansję słabej (bi)symulacji.

Co ciekawe, dla $o = 1$, relacja R_o indukowana przez słowną ekspansję słabej (bi)symulacji jest równoważna odpowiednio relacjom równości i inkluzji języków[2]. Można też zdefiniować analogi powyższych ekspansji dla (bi)symulacji rozgałęziającej.

[2]Przy założeniu, że wszystkie stany (procesy) są akceptujące.

3 Wyniki rozprawy

Rezultaty omówione w rozprawie zostały w większości opisane w pracach [9, 21, 20].

3.1 Aproksymacje

W rozprawie przestudiowane zostały standardowe hierarchie aproksymacji. Pierwsza grupa wyników dotyczy badań nad możliwością aproksymacji słabej równoważności przez hierarchie będące naturalnymi rozwinięciami najprostszej hierarchii aproksymacji, to znaczy aproksymacji indukowanych przez ekspansję słabej bisymulacji. W tej części rozprawy pojawiają się dwa wyniki wskazujące na istotne trudności przy tym podejściu. Mianowicie, udało się nam wykazać, że:

Twierdzenie 1. *Hierarchia aproksymacji indukowanych przez ekspansję długich kroków dla słabej bisymulacji nie stabilizuje się poniżej poziomu ω^2 dla przemiennych gramatyk bezkontekstowych.*

Wynik ten negatywnie rozstrzyga hipotezę sformułowaną przez Hirshfelda i Jančara jakoby ta hierarchia musiała się stabilizować na poziomie $\omega + \omega$. Nie przekłada się to wprawdzie bezpośrednio na pokazanie trudności problemu, ale wskazuje, że długie aproksymacje bardzo nieefektywnie przybliżają słabą równoważność.

Drugi wynik w tej części pracy jest następujący:

Twierdzenie 2. *Hierarchia aproksymacji indukowanych przez słowną ekspansję słabej bisymulacji nie stabilizuje się poniżej poziomu $\omega + \omega$.*

To twierdzenie mówi nam dalece więcej niż poprzedni wynik. Otóż wiele niezmienników bisymulacji, które podejrzewane były o bycie kluczowymi dla rozwiązania problemu, jest rozróżnianych na pewnym skończonym poziomie aproksymacji, a co za tym idzie, nie ma szans aby dowolna ich kombinacja prowadziła do sukcesu.

Z drugiej jednak strony, w czasie badania technik aproksymacji udało nam się zaadoptować je do wszystkich podklas przemiennych gramatyk bezkontekstowych, w których rozstrzygalność została wcześniej udowodniona. Na przykład:

Twierdzenie 3. *Hierarchia aproksymacji indukowanych przez ekspansję długich kroków słabej bisymulacji stabilizuje się na poziomie ω w podklasie rozważanej przez Stirlinga w [38].*

Co za tym idzie, pokazujemy, że tak naprawdę w tym momencie, jeśli chodzi o badania nad słabą bisymulacją dla przemiennych gramatyk bezkontektowych, społeczności naukowej nie udało się wyjść poza standardowe aproksymacje, które wydają się być w ogólności zdecydowanie niewystarczającym narzędziem.

3.2 Rozstrzygalność bisymulacji rozgałęziającej

Najbardziej znaczącym wynikiem pracy jest następujące twierdzenie:

Twierdzenie 4. *Równoważność rozgałęziająca jest rozstrzygalna dla unormowanych przemiennych gramatyk bezkontekstowych.*

Algorytm składa się z dwóch procedur częściowych uruchamianych równolegle: pierwsza – pozytywna, czyli dowodząca, że dwa procesy są równoważne, jest znana od dłuższego czasu; druga, pozwalająca stwierdzać ich nierównoważność, jest jednym z dwóch głównych osiągnięć pracy. Próby rozwiązania problemu przy użyciu naturalnych hierarchii aproksymacji nie przyniosły rezultatów. Jednak analiza kombinatoryczna równoważności rozgałęziającej pozwoliła nam udowodnić, że jeśli procesy α i β są w relacji równoważności rozgałęziającej, to ekspansja spełnia dodatkowo bardzo mocny warunek, nazywany własnością ograniczonej odpowiedzi. Mianowicie, istnieje efektywnie obliczalna stała $c \in \mathbb{N}$ taka, że rozmiar procesu $\overline{\beta}$ w definicji 9 jest co najwyżej c razy większy niż rozmiar procesu α; podobny związek zachodzi również pomiędzy β' i α'.

Wynik ten prowadzi do nowej hierarchii aproksymacji indukowanych przez nową, niestandardową ekspansję. Otóż okazuje się, że taka hierarchia stabilizuje się na poziomie ω, a kolejne aproksymacje są obliczalne. Oba te fakty razem prowadzą do rozstrzygalności. Zdecydowanie najtrudniejszym fragmentem dowodu poprawności tego algorytmu jest oszacowanie wielkości stałej c.

Próby powtórzenia tej ścieżki rozumowania dla słabej bisymulacji prowadzą do dosyć zaskakującego rezultatu, który mówi, że słaba bisymulacja też ma własność ograniczonej odpowiedzi. Niestety nie umiemy w tym przypadku oszacować stałej c, a rozstrzygalność słabej bisymulacji pozostaje otwarta.

3.3 Rozstrzygalności słabej symulacji

Trzecim rezultatem pracy jest:

Twierdzenie 5. *Słaby i rozgałęziający quasi-porządek jest rozstrzygalny dla automatów z jednym licznikiem bez testu zera.*

Algorytm używa kolejnej, nieznanej wcześniej, techniki aproksymacji. Co ciekawe, o ile aproksymacje należące do wcześniej zdefiniowanych hierarchii były łatwe do obliczenia w świetle znanej teorii, o tyle hierarchia użyta tutaj nie ma tej własności. Aby obliczać kolejne elementy łańcucha, jako podprocedury używamy wysoce nietrywialnego algorytmu Jančara pozwalającego obliczać semiliniowy opis relacji symulacji dla automatów z jednym licznikiem bez testu zera i bez ε-tranzycji.

Co więcej, drugi element niezbędny do użycia techniki aproksymacji, a mianowicie stabilizacja na poziomie ω, też nie jest oczywisty. Tutaj, zaskakująco, udaje się pokazać, iż dla każdego automatu istnieje skończona liczba k taka, że hierarchia stabilizuje się na poziomie k, a co za tym idzie, dostajemy wynik silniejszy niż stabilizacja na poziomie ω.

Pokazujemy też, że słaby i rozgałęziający quasi-porządek mogą być efektywnie opisane jako zbiory semiliniowe.

Rozstrzygalność słabego quasi-porządku w tym modelu jest dodatkowo ciekawa ze względu na ogólnie przyjęty w środowisku pogląd, jakoby obliczalność symulacji była trudniejsza niż obliczalność bisymulacji. Twierdzenie 5 przeczy tej regule, gdyż słaba równoważność jest nierozstrzygalna dla automatów z jednym licznikiem bez testu zera.

4 Studium literatury

Rozstrzygalność wielu problemów dla systemów nieskończenie stanowych, takich jak sieci Petriego, automaty ze stosem czy algebry procesów, była badana już od lat 70-tych. Prawdziwy rozkwit możemy jednak notować pod koniec lat 80-tych, kiedy powstały prace Beatena, Bergstry i Klopa [2, 3], w których pokazują oni, że równoważność bisymulacyjna jest rozstrzygalna dla unormowanych gramatyk bezkontekstowych. Wyczerpujące omówienie wyników powstałych od tego czasu dla przemiennych i nieprzemiennych gramatyk bezkontekstowych, a także dla innych modeli, można znaleźć w przeglądowej pracy [37].

4.1 Gramatyki bezkontekstowe

Okazuje się, że dla przemiennych jak i zwykłych nieprzemiennych gramatyk bezkontekstowych, nawet przy założeniu ich unormowania, niemal wszystkie relacje zdefiniowane przez van Glabekka w pracach [11] i [12] są nierozstrzygalne [23, 15, 13, 24]. Jedynym wyjątkiem jest równoważność bisymulacyjna.

Gramatyki bezkontekstowe. Klasa ta często funkcjonuje pod nazwą *Basic Proces Algebra*, np. w [33]. Pierwszy dowód rozstrzygalności równoważności bisymulacyjnej dla gramatyk bez ε-tranzycji pochodzi od Christensena, Hüttela i Stirlinga i jest opisany w pracy [7]. Najlepszy znany obecnie algorytm zaproponowany przez Burkarta, Caucala i Steffena w pracy [4] działa w czasie 2-EXPTIME. Z drugiej strony w pracy [30] Kiefer pokazał, że problem ten jest EXPTIME trudny.

Jeśli dodatkowo założymy unormowanie to, zaskakująco, złożoność spada do wielomianowej. Ciąg prac w tej tematyce [17, 18, 31, 10, 8] wieńczy doktorat Wojciecha Czerwińskiego zawierający najlepszy znany algorytm, o złożoności $O(n^4 polylog(n))$.

Z drugiej strony, znacząco mniej wiadomo o gramatykach z ε-tranzycjami. Jedyne znane rezultaty, dotyczące podklasy gramatyk całkowicie unormowanych (ang. totally normed), można znaleźć w [16, 22]. Rozstrzygalność słabej i rozgałęziającej równoważności pozostają zatem intrygującymi problemami otwartymi.

Przemienne gramatyki bezkontekstowe. Ta klasa często funkcjonuje pod nazwą *Basic Parallel Processes*, np. w [5]. Dla gramatyk bez ε-tranzycji pierwsze wyniki można znaleźć w pracach [6, 14, 25]. Dzięki pracom [36] (dolna granica) i [27] (górna granica) wiemy, że równoważność bisymulacyjna jest PSPACE-zupełna.

Jeśli dodatkowo założymy unormowanie, to dzięki pracy [19] wiemy, że złożoność jest wielomianowa. Najszybszy znany algorytm, działający w czasie $O(n^3)$, można znaleźć pracy [28] autorstwa Jančara i Kota.

Natomiast jeśli dodamy ε-tranzycje, to znowu wkraczamy na słabo zbadane tereny. Podklasa procesów całkowicie unormowanych została rozwiązana w [16]. Kolejnym ciekawym wynikiem jest praca Stirlinga [38], w której dowodzi on rozstrzygalności problemu słabej równoważności dla gramatyk, w których nakłada się pewne ograniczenia na zachowanie zmiennych nazywanych generatorami. W klasie tej, w przeciwieństwie do podklasy całkowicie unormowanej, aproksymacje indukowane przez ekspansję słabej bisymulacji nie stabilizują się na poziomie ω. Prostszy dowód dla tej klasy został podany w pracy [21] i wchodzi w skład niniejszej rozprawy. O ile słaba równoważność pozostaje trudnym problemem otwartym, o tyle w pracy [9] została pokazana rozstrzygalność bisymulacji rozgałęziającej dla gramatyk unormowanych. Dowód wchodzi w skład niniejszej rozprawy.

4.2 Automaty z jednym licznikiem

Automaty z jednym licznikiem to podklasa automatów ze stosem, które używają jednoliterowego alfabetu stosowego. Jeśli automat dodatkowo nie testuje zera, czyli nie może sprawdzać czy stos jest pusty (jak zakładamy w rozprawie), to otrzymujemy model, który jest również podklasą etykietowanych sieci Petriego.

W modelu z testem zera, tak jak poprzednio, większość relacji ze spektrum van Glabeka jest nierozstrzygalna nawet bez ε-przejść. Jedyny pozytywny wynik to rozstrzygalność bisymulacji dla automatów z jednym licznikiem, pokazana przez Jančara w pracy [26]. Wynik ten został niedługo potem rozszerzony przez Senizeurgesa w sławnych pracach [34, 35].

Jak pokazał Jančar, jeśli dodatkowo założymy brak testów zera, to problem równoważności bisymulacyjnej staje się PSPACE-zupełny. Co więcej, jak pokazali Abdulla i Cerans w pracy [1], problem symulacji dla automatów bez ε-tranzycji jest rozstrzygalny. Niedługo potem ich wynik został w bardziej elegancki sposób powtórzony przez Jančara i Mollera w pracy [29]. Dokładna złożoność problemu pozostaje nieznana. Problem równości języków jest nierozstrzygalny, jak pokazano w pracy [20].

Szczególnie interesujące dla nas pytania pojawiają się, jeśli rozszerzymy automaty z jednym licznikiem o ε-tranzycje. Tutaj należy wspomnieć o dwóch pracach. Mayr w pracy [32] pokazał, że słaba równoważność jest nierozstrzygalna. Z drugiej strony, w pracy [20] zaprezentowany jest algorytm rozstrzygający słaby quasi-porządek. Wynik ten wchodzi w skład niniejszej rozprawy doktorskiej.

Literatura

[1] Parosh Aziz Abdulla and Karlis Cerans. Simulation is decidable for one-counter nets (extended abstract). In *CONCUR*, pages 253–268, 1998.

[2] Jos C. M. Baeten and Jan Willem Bergstra, Jan A.and Klop. Decidability of bisimulation equivalence for processes generating context-free languages. In *PARLE (2)*, pages 94–111, 1987.

[3] Jos C. M. Baeten and Jan Willem Bergstra, Jan A.and Klop. Decidability of bisimulation equivalence for processes generating context-free languages. *J. ACM*, 40(3):653–682, 1993.

[4] Olaf Burkart, Didier Caucal, and Bernhard Steffen. An elementary bisimulation decision procedure for arbitrary context-free processes. In *MFCS*, pages 423–433, 1995.

[5] Søren Christensen. *Decidability and Decomposition in process algebras*. PhD thesis, Dept. of Computer Science, University of Edinburgh, UK, 1993.

[6] Søren Christensen, Yoram Hirshfeld, and Faron Moller. Bisimulation equivalence is decidable for Basic Parallel Processes. In *CONCUR*, pages 143–157, 1993.

[7] Søren Christensen, Hans Hüttel, and Colin Stirling. Bisimulation equivalence is decidable for all context-free processes. *Inf. Comput.*, 121(2):143–148, 1995.

[8] Wojciech Czerwiński. *Partially-commutative context-free graphs*. PhD thesis, University of Warsaw, 2012.

[9] Wojciech Czerwiński, Piotr Hofman, and Sławomir Lasota. Decidability of branching bisimulation on normed commutative context-free processes. In *CONCUR*, pages 528–542, 2011.

[10] Wojciech Czerwiński and Sławomir Lasota. Fast equivalence-checking for normed context-free processes. In *FSTTCS*, pages 260–271, 2010.

[11] Rob J. van Glabbeek. The linear time-branching time spectrum (extended abstract). In *CONCUR*, pages 278–297, 1990.

[12] Rob J. van Glabbeek. The linear time - branching time spectrum II. In *CONCUR*, pages 66–81, 1993.

[13] Jan Friso Groote and Hans Hüttel. Undecidable equivalences for basic process algebra. *Inf. Comput.*, 115(2):354–371, 1994.

[14] Yoram Hirshfeld. Congruences in commutative semigroups. Technical report, University of Edinburgh, LFCS report ECS-LFCS-94-291, 1994.

[15] Yoram Hirshfeld. Petri nets and the equivalence problem. In *CSL*, pages 165–174. 1994.

[16] Yoram Hirshfeld. Bisimulation trees and the decidability of weak bisimulations. *Electr. Notes Theor. Comput. Sci.*, 5:2–13, 1996.

[17] Yoram Hirshfeld, Mark Jerrum, and Faron Moller. A polynomial-time algorithm for deciding equivalence of normed context-free processes. In *FOCS*, pages 623–631, 1994.

[18] Yoram Hirshfeld, Mark Jerrum, and Faron Moller. A polynomial algorithm for deciding bisimilarity of normed context-free processes. *Theor. Comput. Sci.*, 158(1&2):143–159, 1996.

[19] Yoram Hirshfeld, Mark Jerrum, and Faron Moller. A polynomial-time algorithm for deciding bisimulation equivalence of normed Basic Parallel Processes. *Mathematical Structures in Computer Science*, 6(3):251–259, 1996.

[20] Piotr Hofman, Richard Mayr, and Patrick Totzke. Decidability of weak simulation on one-counter nets. In *LICS*, 2013. Accepted.

[21] Piotr Hofman and Patrick Totzke. Approximating weak bisimilarity of basic parallel processes. In *DCM*, pages 99–113, 2012.

[22] Hans Hüttel. Silence is golden: Branching bisimilarity is decidable for context-free processes. In *CAV*, pages 2–12, 1991.

[23] Hans Hüttel. Undecidable equivalences for Basic Parallel Processes. In *TACS*, pages 454–464, 1994.

[24] Dung T. Huynh and Lu Tian. On deciding readiness and failure equivalences for processes. *Inf. Comput.*, 117(2):193–205, 1995.

[25] Petr Jančar. Undecidability of bisimilarity for petri nets and some related problems. *Theor. Comput. Sci.*, 148(2):281–301, 1995.

[26] Petr Jančar. Bisimulation equivalence is decidable for one-counter processes. In *ICALP*, pages 549–559, 1997.

[27] Petr Jančar. Strong bisimilarity on Basic Parallel Processes is PSPACE-complete. In *LICS*, pages 218–227, 2003.

[28] Petr Jančar and Martin Kot. Bisimilarity on normed Basic Parallel Processes can be decided in time $O(n^3)$. In *AVIS*, 2004.

[29] Petr Jančar and Faron Moller. Simulation of one-counter nets via colouring. Technical report, Uppsala Computing Science, February 1999.

[30] Stefan Kiefer. BPA bisimilarity is EXPTIME-hard. *Inf. Process. Lett.*, 113(4):101–106, 2013.

[31] Sławomir Lasota and Wojciech Rytter. Faster algorithm for bisimulation equivalence of normed context-free processes. In *MFCS*, pages 646–657, 2006.

[32] Richard Mayr. Undecidability of weak bisimulation equivalence for 1-counter processes. In *ICALP*, pages 570–583, 2003.

[33] A. Ponse and Scott A. Smolka, editors. *Handbook of Process Algebra*. Elsevier Science Inc., 2001.

[34] Géraud Sénizergues. The equivalence problem for deterministic pushdown automata is decidable. In *ICALP*, pages 671–681, 1997.

[35] Géraud Sénizergues. Complete formal systems for equivalence problems. *Theor. Comput. Sci.*, 231(2):309–334, 2000.

[36] Jirí Srba. Strong bisimilarity and regularity of Basic Parallel Processes is PSPACE-hard. In *STACS*, pages 535–546, 2002.

[37] Jirí Srba. *Roadmap of Infinite results*, volume Vol 2: Formal Models and Semantics. World Scientific Publishing Co., 2004.

[38] Colin Stirling. Decidability of weak bisimilarity for a subset of Basic Parallel Processes. In *FoSSaCS*, pages 379–393, 2001.