

Częściowo przemienne grafy bezkontekstowe

Wojciech Czerwiński

autoreferat rozprawy doktorskiej

Tematem rozprawy jest klasa częściowo przemiennych grafów bezkontekstowych. Jest to model obliczeń odzwierciedlający zarówno rekurencyjne jak i współbieżne zachowania programów. W dziedzinie automatycznej weryfikacji i analizy programów istnieje zapotrzebowanie na systemy, które potrafią modelować różne aspekty programów a przy okazji posiadają dobre własności algorytmiczne. Wydaje się, że jesteśmy wciąż na początku drogi ku efektywnej automatycznej weryfikacji, nie mniej jednak niektóre techniki stosowane są już w przemyśle.

Rezultaty przedstawione w tej pracy są podzielone na dwie główne części. Pierwsza z nich skupia się na wyrażalności badanego modelu, druga przedstawia metody jego analizy. Część algorytmiczna dotyczy głównie problemów osiągalności oraz sprawdzania równoważności.

Grafy bezkontekstowe. Jednym z powszechnie znanych formalizmów w teorii języków i automatów są gramatyki bezkontekstowe. Najwygodniejsze z punktu widzenia weryfikacji i analizy systemów są gramatyki w postaci Greibach. Gramatyka w postaci Greibach składa się z nieterminali, liter alfabetu oraz produkcji postaci:

$$X \xrightarrow{a} X_1 \dots X_k, \quad (1)$$

gdzie X, X_1, \dots, X_k są nieterminalami, a zaś jest literą alfabetu. Dozwolone są produkcje postaci $X \xrightarrow{a} \varepsilon$.

Dla gramatyki bezkontekstowej możemy w sposób naturalny zdefiniować graf konfiguracji. Jest to graf etykietowany¹, którego wierzchołkami są wszystkie konfiguracje (czyli w tym wypadku słowa złożone z nieterminali) a krawędzie odpowiadają produkcjom. Taki graf nazywamy grafem bezkontekstowym, a klasę grafów grafami bezkontekstowymi.

Przykład 1 Jako ilustrację rozważmy następującą gramatykę bezkontekstową w postaci Greibach z dwoma nieterminalami: S i B , trzema literami: a , b i s

¹W rzeczywistości jest multigraf, gdyż może istnieć wiele krawędzi, etykietowanych różnymi literami, pomiędzy parą tych samych wierzchołków. Nie mniej jednak w tym tekście używamy terminu graf.

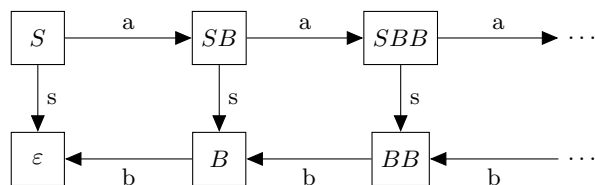
oraz trzema produkcjami

$$S \xrightarrow{a} SB \quad S \xrightarrow{s} \varepsilon \quad B \xrightarrow{b} \varepsilon.$$

Przykładowy wywód słowa a^2sb^2 to

$$S \xrightarrow{a} SB \xrightarrow{a} SBB \xrightarrow{s} BB \xrightarrow{b} B \xrightarrow{b} \varepsilon,$$

a język generowany przez tę gramatykę dla nieterminala początkowego S to zbiór słów postaci $a^n sb^n$ dla $n \geq 0$. Graf konfiguracji jest następujący:



Grafy bezkontekstowe są uważane za podstawowy model rekursji w programach. Produkcja $X \xrightarrow{a} YZ$ jest interpretowana jako wywołanie rekurencyjnego procesu potomnego w stanie Y z procesu w stanie X . Po zakończeniu procesu potomnego proces pierwotny rozpocznie dalsze obliczenie w stanie Z . Produkcja ta może być zaobserwowana z zewnątrz systemu jako zdarzenie typu a .

Przemienne grafy bezkontekstowe. Modelem bardzo podobnym do gramatyk bezkontekstowych są mniej znane przemienne gramatyki bezkontekstowe. One także składają się z nieterminali, liter oraz produkcji postaci (1). Różnicą jest interpretacja produkcji: pomijamy ograniczenie dotyczące lewostronności wywodu. Intuicyjnie porządek nieterminali w produkcji jest tu nieistotny, tak jakby były one zupełnie przemienne. Języki definiowalne przez takie gramatyki nazywamy przemiennymi językami bezkontekstowymi.

W tym wypadku konfiguracją jest multizbiór nieterminali zamiast słowa złożonego z nieterminali, gdyż kolejność nie gra roli. Dlatego wierzchołki grafu konfiguracji to multizbiory nieterminali, a krawędzie jak poprzednio odpowiadają produkcjom. Tak określony zbiór grafów nazywamy przemiennymi grafami bezkontekstowymi.

Przemienne grafy bezkontekstowe służą jako podstawowy model współbieżności. Produkcja $X \xrightarrow{a} YZ$ jest interpretowana jako instrukcja wywołana przez proces w stanie X . Skutkuje ona przejściem do stanu Y oraz powołaniem procesu potomnego rozpoczynającego obliczenie w stanie Z . Poza systemem widoczne jest tylko, że nastąpiło zdarzenie typu a

Opólniejsze klasy grafów. Opisane powyżej grafy bezkontekstowe (dalej zwane sekwencyjnymi) oraz przemienne grafy bezkontekstowe są szczególnymi przypadkami klasy systemów zwanych PRS (Process Rewrite Systems) zaproponowanej przez R. Mayra w 1997 roku [14]. Przedstawił on w jednolity sposób

szereg znanych klas systemów nieskończenie stanowych, między innymi: sekwencyjne i przemienne grafy bezkontekstowe, grafy automatów ze stosem, sieci Petriego itd. Jedną z podklas PRS są grafy PA, bardzo naturalne uogólnienie zarówno sekwencyjnych jak i przemiennych grafów bezkontekstowych.

Gramatyka PA składa się ze skończonych zbiorów nieterminali, liter i produkcji postaci

$$X \xrightarrow{a} t,$$

gdzie t jest termem zawierającym nieterminale w liściach (jako symboli zeroargumentowych) oraz dwie operacje dwuargumentowe: złożenia sekwencyjnego ';' i złożenia równoległego '||'. Konfiguracjami w tym systemie są termy. Złożenie sekwencyjne $t_1; t_2$ interpretowane jest jako wykonanie najpierw operacji dotyczących termu t_1 , a potem termu t_2 , a złożenie równoległe $t_1 || t_2$ jako równoczesne przetwarzanie obu termów. Wierzchołki termu, które w danym momencie można przetwarzać to te liście, których żaden przodek nie jest drugim argumentem złożenia sekwencyjnego. Innymi słowy intuicyjnie nie istnieje operacja, która koniecznie musi być wykonana wcześniej. Aplikacja produkcji do takiego liścia polega na zastąpieniu liścia prawą stroną produkcji. Graf konfiguracji określamy analogicznie jak w poprzednich rozdziałach, wierzchołki to termy, a krawędzie odpowiadają produkcjom.

Grafy PA odzwierciedlają zarówno mechanizmy rekurencyjne jak i współbieżne, w związku z czym dobrze nadają się jako model do takich obliczeń. Problemem są jednak ich własności algorytmiczne. Wiele pytań ma w tym wypadku złożoność obliczeniową wyższą niż dla sekwencyjnych lub przemiennych grafów bezkontekstowych.

Między innymi ten fakt motywuje poszukiwanie innej naturalnej wspólnej nadklasy obu klas grafów bezkontekstowych.

Częściowo przemienne grafy bezkontekstowe. Jak wspominaliśmy powyżej produkcje zarówno w sekwencyjnych jak i przemiennych grafach bezkontekstowych mają tę samą postać (1):

$$X \xrightarrow{a} X_1 \dots X_k,$$

ale inną interpretację. W pierwszy przypadku nieterminale nie mogą się w ogóle zamieniać miejscami, w drugim wszystkie nieterminale mogą się zamieniać miejscami (więc istotny jest tylko multizbiór nieterminali).

Ta obserwacja pozwala na ogólniejszą definicję pozwalającą niektórym parom nieterminali zamieniać się ze sobą, a innym nie. Pomysł ten jest zaczerpnięty z teorii śladów [15, 16, 7], która bada własności języków słów w sytuacji, gdy pozwolimy niektórym literom (terminalom) zamieniać się ze sobą. Warto jednak podkreślić, że te dwa podejścia różnią się między sobą: teoria śladów bada częściową przemienność na poziomie liter, a prezentowany pomysł bada ją na poziomie nieterminali.

Częściowo przemienne gramatyka bezkontekstowa składa się ze skończonego zbioru nieterminali V , skończonego zbioru liter alfabetu Σ , skończenie wielu produkcji postaci (1) oraz dodatkowo symetrycznej, przeciwzwrotnej *relacji*

niezależności $I \subseteq V \times V$. Powiemy, że dwa nieterminale X i Y są *niezależne* gdy $(X, Y) \in I$. Dla ułatwienia definiujemy także *relację zależności* $D \subseteq V \times V$ jako $D = (V \times V) \setminus I$ i mówimy, że X i Y są *zależne* gdy $(X, Y) \in D$. Relacja D jest zatem symetryczna i zwrotna.

Dwa słowa złożone z nieterminali nazwiemy *równoważnymi* gdy jedno może być przekształcone na drugie poprzez ciąg operacji zamian par sąsiednich, niezależnych nieterminali. Innymi słowy relacja ta, \sim_I , jest zdefiniowana jako najmniejsza relacja równoważności zawierająca pary

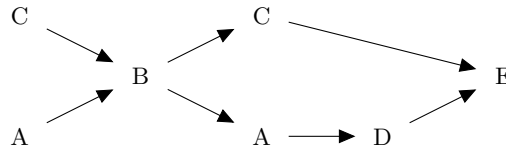
$$(\alpha X Y \beta, \alpha Y X \beta)$$

dla wszystkich par nieterminali $(X, Y) \in I$ oraz słów $\alpha, \beta \in V^*$.

Konfiguracją w tym systemie jak klasa abstrakcji relacji \sim_I . Zauważmy, że ta definicja odpowiada słowu dla sekwencyjnych grafów bezkontekstowych i multizbiorowi dla przemiennych grafów bezkontekstowych.

Naturalną reprezentacją konfiguracji jest porządek częściowy, w którym nieterminal X jest większy od nieterminala Y gdy w każdym słowie z rozważanej klasy abstrakcji ten konkretny X jest wcześniej w słowie niż ten konkretny Y . Aby tak się zdarzyło albo nieterminale X i Y muszą być zależne albo musi istnieć sekwencja nieterminali Z_1, \dots, Z_k taka, że X jest zależny z Z_1 , Z_i jest zależny z Z_{i+1} dla $i \in \{1, \dots, k-1\}$ oraz Z_k jest zależny z Y .

Przykład 2 Dla zilustrowania powyższego pojęcia rozważmy słowo $w = CABACDE$ oraz relację D zawierającą pary (A, B) , (B, C) , (C, D) , (D, E) i symetryczne oraz pary identycznościowe. Wówczas graficzna ilustracja konfiguracji $[w]_{\sim_I}$ wygląda następująco



Rysunek 1: Konfiguracja $[w]_{\sim_I}$ dla $w = CABACDE$

Graf konfiguracji jest zdefiniowany naturalnie. Wierzchołki to konfiguracje, a krawędzie są postaci

$$[X \cdot \alpha]_{\sim_I} \xrightarrow{a} [\gamma \cdot \alpha]_{\sim_I}$$

dla pewnej produkcji $X \xrightarrow{a} \gamma$ należącej do zbioru Δ . Odpowiadające języki nazywamy częściowo przemiennymi językami bezkontekstowymi, a grafy częściowo przemiennymi grafami bezkontekstowymi.

Przechodnia relacja zależności. Zakładając przechodniość relacji zależności otrzymujemy model, który wydaje się mieć ciekawsze nawet własności niż

poprzedni. Zauważmy bowiem, że dla częściowo przemiennych grafów bezkontekstowych relacja zależności zawsze jest zwrotna i symetryczna, więc przy powyższym założeniu staje się ona relacją równoważności. Klasę abstrakcji tej relacji nazwiemy *wątkiem*. Każdy nieterminal jest wówczas zależny z nieterminalami z tego samego wątku, niezależny zaś od nieterminali z innych wątków. Zauważmy, że w tej sytuacji diagram z Rysunku 2 staje się zbiorem niezależnych liniowych porządków. Konfiguracja jest zatem krotką słów, po jednym słowie dla każdego wątku.

Powyższą klasę na potrzeby tego autoreferatu będziemy nazywać w skrócie przechodnimi grafami (lub językami) bezkontekstowymi.

Przykład 3 Dla ilustracji rozważmy następującą gramatykę

$$\begin{array}{l} S \xrightarrow{a} SP \quad P \xrightarrow{c} B \quad B \xrightarrow{b} \varepsilon \\ S \xrightarrow{\bar{a}} \varepsilon \quad P \xrightarrow{b} C \quad C \xrightarrow{c} \varepsilon \end{array}$$

z początkowym nieterminalem S oraz wątkami $\{S, P\}$, $\{B\}$ and $\{C\}$. Przykładowym wywodem dla słowa $aa\bar{a}bbcc$ jest

$$S \xrightarrow{a} SP \xrightarrow{a} SPP \xrightarrow{\bar{a}} PP \xrightarrow{b} CP \xrightarrow{b} PC \xrightarrow{b} CC \xrightarrow{c} C \xrightarrow{c} \varepsilon,$$

gdzie dla uproszczenia zamiast klas abstrakcji wypisujemy ich reprezentantów, konkretne słowa. Zauważmy, że każdy wywód ma prefiks postaci

$$S \xrightarrow{a} \dots \xrightarrow{a} SP^n \xrightarrow{\bar{a}} P^n$$

oraz z P^n może być wyprowadzony dowolny przeplot słów b^n i c^n . Zatem język dla powyższej gramatyki to

$$\bigcup_{n \geq 0} a^n \bar{a} (b^n \parallel c^n),$$

gdzie przez $u \parallel w$ oznaczamy zbiór przeplotów słów u i w .

Przechodnie grafy bezkontekstowe są równoważne bardzo naturalnemu modelowi automatu. *Bezstanowy automat wielostosowy* jest bardzo podobny do zwykłego automatu ze stosem, z tą tylko różnicą, że ma wiele stosów i tylko jeden stan. Ponieważ od stanu nic nie zależy, to ignorujemy go w dalszych rozważaniach. Automat w pojedynczym kroku czyta jedną literę z wejścia i wybiera niedeterministycznie pewną produkcję. W wyniku produkcji pewien symbol stosowy jest zdejmowany a kilka innych (być może z różnych stosów) jest wkładane na odpowiednie stosy.

Odpowiedniość między przechodnimi grafami bezkontekstowymi a bezstanowymi automatami wielostosowymi jest naturalna. W automacie stosy odpowiadają wątkom a symbole stosowe nieterminalom.

Rezultaty Rozprawa doktorska skupia się na badaniu własności częściowo przemiennej grafów bezkontekstowych oraz ich ważnej podklasy przechodnich grafów bezkontekstowych. Wyniki przedstawione w pracy podzielone są na trzy części. Pierwsza część porównuje wyrażalność opisywanych dwóch klas oraz innych modeli posiadających zarówno współbieżne jak i sekwencyjne własności. Druga część dotyczy problemu osiągalności. Ostatnia bada rozstrzygalność problemu równoważności dwóch konfiguracji w częściowo przemiennej grafie bezkontekstowym.

Wyrażalność. Aby pokazać, że pewna klasa struktur \mathcal{K}_1 ma wyrażalność nie mniejszą niż druga klasa struktur \mathcal{K}_2 standardowo szuka się struktury w \mathcal{K}_1 takiej, która nie ma sobie równoważnej w \mathcal{K}_2 . W zależności od tego jaką badamy równoważność możemy otrzymać różne odpowiedzi. W przypadku grafów etykietowanych naturalnym wyborem może być dobrze znana równoważność językowa, ale także inne rodzaje równoważności semantycznych. Jedną z najważniejszych jest równoważność bisymulacyjna [17, 18], która jako relacja jest zawarta w równoważności językowej. Badanych było również wiele innych, zawartych pomiędzy równoważnością językową a bisymulacyjną, znanych pod wspólną nazwą spektrum van Glabbeeka [8].

W pracy doktorskiej opisane jest porównanie wyrażalności częściowo przemiennej grafów bezkontekstowych, ich podklasy przechodnich grafów bezkontekstowych i innych, znanych wcześniej modeli łączących sekwencyjność ze współbieżnością. Większość klas okazuje się być nieporównywalnymi, a takie rezultaty są najmocniejsze w wypadku największych relacji. Między innymi dlatego podczas badania wyrażalności badana jest wyrażalność językowa opisanych klas.

Rozprawa pokazuje, że badane klasy grafów bezkontekstowych są nieporównywalne z klasą języków grafów PA oraz z klasą języków bezkontekstowych domkniętych ze względu na relację przemienności na literach. Formalnie, ostatnia klasa zawiera wszystkie języki postaci:

$$\{w : \exists_{u \in L} w \sim_I u\},$$

gdzie L jest pewnym językiem bezkontekstowym nad alfabetem Σ , $I \subseteq \Sigma \times \Sigma$ symetryczną relacją niezależności, a \sim_I najmniejszą relacją równoważności zawierającą wszystkie pary słów

$$(u a b w, u b a w)$$

dla $u, w \in \Sigma^*$, $(a, b) \in I$.

Wyniki dotyczące wyrażalności są najtrudniejszą technicznie częścią opisywanego rozdziału. Część z nich udało się uzyskać za pomocą lematów o pompowaniu opracowanych specjalnie dla badanych klas. Inne jednak wydają się wymagać bardziej subtelnej analizy.

W rozdziale dotyczącym wyrażalności rozważane są także inne aspekty klas częściowo przemiennej języków bezkontekstowych oraz przechodnich języków bezkontekstowych.

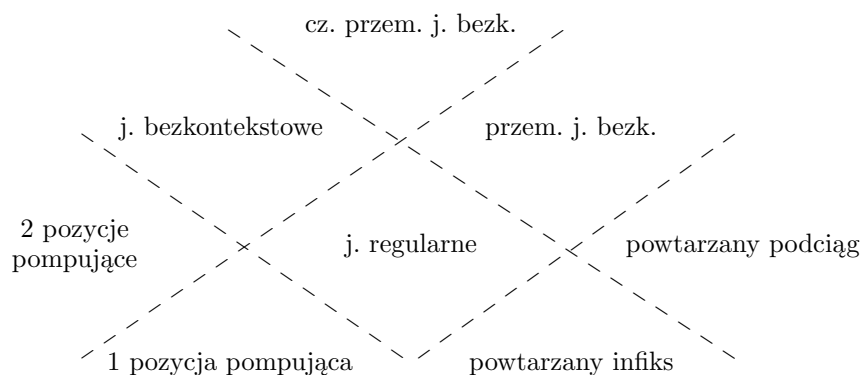
Zbadane są własności domknięcia. Obie klasy są zamknięte na operacje sumy i przepłotu. Pierwsza z nich jest też zamknięta na konkatencję, obrazy homomorficzne oraz substytucję. Druga nie jest zamknięta na konkatencję, w przypadku obrazów homomorficznych i substytucji nie znamy odpowiedzi, jednak przypuszczamy, że również nie posiada tej własności. Obie klasy nie posiadają własności zamknięcia na odwrotne obrazy homomorficzne i przecięcia z językami regularnymi. Powyższe fakty nie wymagają skomplikowanych metod, dosyć standardowe techniki dostarczają odpowiedzi.

Ostatnim wynikiem w tym rozdziale jest opracowanie wspomnianych wyżej lematów o pompowaniu. Dla częściowo przemiennych języków bezkontekstowych brzmi on:

Lemat 1 *Dla dowolnego częściowo przemiennego języka bezkontekstowego L istnieje stała $N \in \mathbb{N}$ taka, że jeśli $w \in L$, gdzie $|w| > N$ to istnieją słowa x, y, z, s, t takie, że*

1. $w \in x((s(y|t))|z)$,
2. $1 \leq |syt| \leq N$ oraz
3. $\forall m \geq 0, x s^m y t^m z \in L$.

Lemat ten przypomina analogiczny dla bezkontekstowych, jedyną różnicą jest to, że zamiast równości $w = xsyztz$ mamy $w \in x((s(y|t))|z)$, czyli zamiast infiksów otrzymujemy specyficznie wybrane podciągi słowa w . Powyższy lemat oraz lematy dla klas języków regularnych, bezkontekstowych i przemiennych bezkontekstowych dają się przedstawić w ogólnym jednolitym schemacie. Obecność sekwencyjności w języku powoduje wystąpienie dwóch miejsc pompowania, obecność równoległości zaś owocuje wybieraniem podciągów zamiast infiksów. Sytuację ilustruje rysunek



Problem osiągalności. Powyżej wspomnieliśmy, że przechodnie grafy bezkontekstowe posiadają naturalny odpowiadający model automatu, bezstanowy

automat wielostosowy. W rozprawie rozważamy problem osiągalności dla ogólnych automatów wielostosowych. W szczególności rozwiązujemy problem dla przypadku bezstanowego, w którym złożoność obliczeniowa jest lepsza niż w ogólności.

Automaty wielostosowe są zupełne w sensie Turinga, co skutkuje nierozstrzygalnością problemu osiągalności. Okazuje się jednak, że po niewielkich ograniczeniach ich mocy problem staje się rozstrzygalny, a w wielu przypadkach nawet NP-zupełny. Rozważaliśmy dwa rodzaje ograniczeń. Pierwsze z nich to *unormowanie*: symbol stosowy jest unormowany jeżeli jego język jest niepusty, automat jest unormowany jeśli każdy symbol stosowy jest unormowany. Drugi rodzaj to ograniczenia na strukturę stanów. Automat jest *słaby*, jeżeli istnieje liniowy porządek na stanach taki, że dla każdego przejścia automatu $q \rightarrow q'$ stan q' jest mniejszy lub równy stanowi q w rozważanym porządku. Innymi słowy nie ma nietrywialnych pętli w strukturze stanów.

Najprostszym wariantem osiągalności jest pytanie o możliwość osiągnięcia konfiguracji końcowej t z konfiguracji początkowej s . My rozważaliśmy problem w nieco większej ogólności, dla początkowego zbioru konfiguracji S oraz końcowego zbioru konfiguracji T . Przy założeniu, że S i T są zbiorami regularnymi wiele problemów nadal miało dobrą złożoność.

Okazało się, że dopuszczenie regularnego zbioru początkowego zazwyczaj nie pogarsza złożoności w stosunku do pojedynczego stanu początkowego. Dlatego przedstawiamy wyniki przy założeniu, że zbiór początkowy jest dowolnym zbiorem regularnym. Główne rezultaty są skrótowo przedstawione w poniższej tabeli.

Reg \rightsquigarrow singleton Reg \rightsquigarrow Reg	unormowany	nieunormowany
bezstanowy	NP-zupełność	NP-zupełność nierozstrzygalność
słaby	rozstrzygalność nierozstrzygalność	rozstrzygalność nierozstrzygalność

Równoważność bisymulacyjna. Równoważność bisymulacyjna jest jedną z fundamentalnych równoważności semantycznych [17, 18]. Jest najbardziej odróżniającą spośród wszystkich równoważności w spektrum van Glabbeeka, a równocześnie często jedyną rozstrzygalną (grafy bezkontekstowe [9], przemienne grafy bezkontekstowe [13]). Równoważność bisymulacyjna może być widzialna jako ta właściwa dla systemów niedeterministycznych, tak jak równoważność językowa jest właściwa dla systemów deterministycznych.

Rozważmy graf o zbiorze wierzchołków V i krawędziach etykietowanych alfabetem Σ . Relacja $R \subseteq V \times V$ jest *bisymulacją* jeżeli dla wszystkich par wierzchołków $(p, q) \in R$ następujące warunki są spełnione:

- dla każdej litery $a \in \Sigma$ i dla każdej krawędzi $p \xrightarrow{a} p'$ istnieje krawędź $q \xrightarrow{a} q'$ taka, że $(p', q') \in R$

- dla każdej litery $a \in \Sigma$ i dla każdej krawędzi $q \xrightarrow{a} q'$ istnieje krawędź $p \xrightarrow{a} p'$ taka, że $(p', q') \in R$

Suma bisymulacji jest również bisymulacją, więc dla danego grafu istnieje największa relacja bisymulacji, ona właśnie nazywana jest równoważnością bisymulacyjną i oznaczana $p \sim q$.

Była ona badana intensywnie dla różnych klas Process Rewrite Systems [19]. Najczęstszym pytaniem jest: dany graf G i dwa jego wierzchołki p i q ; rozstrzygnij, czy $p \sim q$. Dla unormowanych grafów bezkontekstowych oraz unormowanych przemianych grafów bezkontekstowych problem ten jest rozstrzygalny w czasie wielomianowym [11, 12]. Dla grafów PA najlepszy znany algorytm jest podwójnie wykładniczy [10].

Głównym wynikiem w rozprawie dotyczącym bisymulacji jest algorytm wielomianowy rozstrzygający równoważność bisymulacyjną w podklasie przechodnich grafów bezkontekstowych. Klasę tę nazywamy *rozłączną*, jest ona nadklasą zarówno sekwencyjnych jak i przemianych grafów bezkontekstowych. Zaletą algorytmu jest jednolite przedstawienie algorytmów dla sekwencyjnych i przemianych grafów bezkontekstowych. Poprzednio istniejące algorytmy dla tych klas istotnie się różniły.

Przedstawiamy też nowy algorytm dla grafów bezkontekstowych, uważną adaptację powyższego dla tego przypadku. Jest on jest najszybszym znanym do tej pory, działa w czasie $\mathcal{O}(n^4 \text{polylog}(n))^2$. Dodatkowo, w podklasie gramatyk prostych, czyli deterministycznych gramatyk bezkontekstowych, algorytm ma złożoność $\mathcal{O}(n^3 \text{polylog}(n))$, która jest również najlepszą do tej pory znaną.

Idea algorytmu opiera się na obliczaniu relacji równoważności bisymulacyjnej poprzez przybliżanie się do niej z góry, podobnie zresztą jak w niektórych poprzednich pracach. Jako pierwszy krok znajdowana jest relacja \sim_0 będąca ograniczeniem górnym szukanej. Następnie w każdej iteracji z \sim_k wyliczana jest pewna poprawka \sim_{k+1} , nie większa niż poprzednia relacja, ale wciąż nie mniejsza niż równoważności bisymulacyjnej. Algorytm zaprojektowany jest w ten sposób, by zagwarantować, że gdy punkt stały obliczeń zostanie osiągnięty, czyli $\sim_k = \sim_{k+1}$ to będzie nim dokładnie szukana równoważność, $\sim_k = \sim$.

Zarówno początkowa relacja, jak i wszystkie pośrednie są zdefiniowane na potencjalnie nieskończonym zbiorze V . Aby w trakcie algorytmu utrzymywać je w pamięci potrzebna jest pewna ich skończona reprezentacja. Istotną techniczną trudnością było wykazanie, że w przypadku klasy traktowalnej każda pośrednia relacja daje się reprezentować w taki sposób. Zostało to osiągnięte przez pokazanie tzw. własności jednoznacznego rozkładu.

Aby zapewnić dobrą złożoność obliczeniową konieczne było zastosowanie pewnych optymalizacji. Obiekty musiały być trzymane w skompresowanej formie, a operacje dokonywane na nich nie mogły wymagać rozkompresowania. W tym celu użyliśmy algorytmu, który pozwala przechowywania rodziny słów oraz szybkie wykonywanie operacji na nich [1]. Dozwolone operacje, m.in. test

²Złożoność ta jest liczona przy powszechnie stosowanym założeniu, że operacje atomowe nawet na dużych liczbach wykonywane są w czasie stałym.

na równość oraz konkatenacja dwóch znanych słów wykonywane są w czasie $\mathcal{O}(n \text{ polylog}(n))$.

Problem pozostaje otwarty dla klasy unormowanych grafów częściowo przemennych. Nie jest znany żaden rezultat dotyczący rozstrzygalności. nierozstrzygalny jest jedynie problem bisymulacji dla nieunormowanych częściowo przemennych grafów bezkontekstowych z ε -przejściami. Rezultat ten może być prosto uzyskany przez dostosowanie konstrukcji z pracy [20].

Prezentacja wyników. Wyniki opisane w rozprawie doktorskiej zostały opublikowane w kilku pracach. Rezultaty opisywane w rozdziale dotyczącym wyrażalności znajdują się w [6]. Badania dotyczące problemu osiągalności przedstawione zostały w [4]. Algorytm dotyczący rozstrzygnięcia równoważności bisymulacyjnej w klasie traktowalnych przechodnich grafów bezkontekstowych jest opisany w [2, 3]. Poprzednia wersja algorytmu dla klasy grafów bezkontekstowych, działająca w czasie $\mathcal{O}(n^5)$ została przedstawiona w [5]. Aktualna wersja, działająca w czasie $\mathcal{O}(n^4 \text{ polylog}(n))$ nie jest jeszcze opublikowana.

Literatura

- [1] Stephen Alstrup, Gerth Stølting Brodal, and Theis Rauhe. Pattern matching in dynamic texts. In *SODA*, pages 819–828, 2000.
- [2] Wojciech Czerwinski, Sibylle B. Fröschle, and Sławomir Lasota. Partially-commutative context-free processes. In *CONCUR*, pages 259–273, 2009.
- [3] Wojciech Czerwinski, Sibylle B. Fröschle, and Sławomir Lasota. Partially-commutative context-free processes: Expressibility and tractability. *Inf. Comput.*, 209(5):782–798, 2011.
- [4] Wojciech Czerwiński, Piotr Hofman, and Sławomir Lasota. Reachability problem for weak multi-pushdown automata. Accepted to CONCUR 2012., 2012.
- [5] Wojciech Czerwinski and Sławomir Lasota. Fast equivalence-checking for normed context-free processes. In *FSTTCS*, pages 260–271, 2010.
- [6] Wojciech Czerwiński and Sławomir Lasota. Partially-commutative context-free languages. Almost submitted to EXPRESS 2012, 2012.
- [7] Volker Diekert and Grzegorz Rozenberg, editors. *Book of Traces*. World Scientific, Singapore, 1995.
- [8] R.J. van Glabbeek. The linear time – branching time spectrum I; the semantics of concrete, sequential processes. In J.A. Bergstra, A. Ponse, and S.A. Smolka, editors, *Handbook of Process Algebra*, chapter 1, pages 3–99. Elsevier, 2001.

- [9] Jan Friso Groote and Hans Hüttel. Undecidable equivalences for basic process algebra. *Inf. Comput.*, 115(2):354–371, 1994.
- [10] Yoram Hirshfeld and Mark Jerrum. Bisimulation equivalence is decidable for normed process algebra. In *ICALP*, pages 412–421, 1999.
- [11] 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.
- [12] 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.
- [13] Hans Hüttel. Undecidable equivalences for basic parallel processes. In *TACS*, pages 454–464, 1994.
- [14] Richard Mayr. Process rewrite systems. *Electr. Notes Theor. Comput. Sci.*, 7:185–205, 1997.
- [15] Antoni W. Mazurkiewicz. Trace theory. In *Advances in Petri Nets*, pages 279–324, 1986.
- [16] Antoni W. Mazurkiewicz. Basic notions of trace theory. In *REX Workshop*, pages 285–363, 1988.
- [17] Robin Milner. *A Calculus of Communicating Systems*, volume 92 of *Lecture Notes in Computer Science*. Springer, 1980.
- [18] David Michael Ritchie Park. Concurrency and automata on infinite sequences. In *Theoretical Computer Science*, pages 167–183, 1981.
- [19] Jirí Srba. Roadmap of infinite results. *Bulletin of the EATCS*, 78:163–175, 2002. see also an updated online version on <http://www.brics.dk/srba/roadmap/>.
- [20] Jirí Srba. Undecidability of weak bisimilarity for PA-processes. In *Developments in Language Theory*, pages 197–208, 2002.