Autoreferat

Marzec 2019

1 Imię i nazwisko

Lorenzo Clemente

2 Dyplomy i stopnie

- Stopień doktora nauk informatycznych uzyskany w 2012 r. na Uniwersytecie Edinburgskim. Praca doktorska: *Uogólnione relacje symulacji z zastosowaniami w teorii automatów* (ang. *Generalized Simulation Relations with Applications in Automata Theory*).
- Tytuł magistra inżynierii uzyskany w 2008 r. na Universytecie Rzymskim "Tor Vergata". Praca magisterska: Algorytmy i narzędzia dla weryfikacji formalnej w systemach współbierznych (ang. *Algorithms and Tools for the Formal Verification of Concurrent Systems*).

3 Zatrudnienie w jednostkach naukowych

- Uniwersytet w Edinburgu (Wielka Brytania), 2008–2011 (3 lata), asystent dydaktyczny.
- Université de Bordeaux (Francja), 2011–2013 (2 lata), staż podoktorski.
- Université Libre de Bruxelles (Belgia), 2013–2014 (7 miesięcy), staż podoktorski.
- Uniwersytet Warszawski (Polska), 2014–2016 (2 lata), staż podoktorski.
- Uniwersytet Warszawski (Polska), od 2016, adiunkt (aktualne zatrudnienie).

4 Wskazanie osiągnięcia wynikającego z art. 16 ust. 2 ustawy z dnia 14 marca 2003 r. o stopniach naukowych i tytule naukowym oraz o stopniach i tytule naukowym w zakresie sztuki

4.1 Tytuł

Wzbogacenie o czas systemów nieskończenie-stanowych

4.2 Prace wchodzące w skład osiągnięcia naukowego

[A] Lorenzo Clemente, Sławomir Lasota. *Binary reachability of timed pushdown automata via quantifier elimination.* In Proc. of ICALP'18, pages 118:1–118:14.

[B] Lorenzo Clemente, Sławomir Lasota, Ranko Lazić, and Filip Mazowiecki. *Timed pushdown automata and branching vector addition systems.* In Proc. of LICS'17, pages 1–12.

[C] Lorenzo Clemente, Sławomir Lasota. *Timed Pushdown Automata Revisited.* In Proc. of LICS'15, pages 738–749.

[D] Lorenzo Clemente and Sławomir Lasota. *Reachability Analysis of First-order Definable Pushdown Systems.* In Proc. of CSL'15, pages 244–259.

[E] Lorenzo Clemente, Frédéric Herbreteau, and Grégoire Sutre. *Decidable Topologies for Communicating Automata with FIFO and Bag Channels.* In Proc. of CONCUR'14, pages 281–296.

[F] Lorenzo Clemente, Frédéric Herbreteau, Amelie Stainer, and Grégoire Sutre. *Reachability of Communicating Timed Processes.* In Proc. of FOSSACS'13, pages 81–96.

4.3 Omówienie wyników

4.3.1 Wstęp

W naszym codziennym życiu polegamy na prawidłowym funkcjonowaniu systemów ICT (ang. *Information and Communication Technology*) w coraz większym stopniu. Przykładami systemów ICT są systemy wbudowane (ang. *embedded systems*), kryptowaluty (n.p. Bit Coin), protokoły komunikacyjne, oprogramowanie komputerowe, itd. Przez długi okres paradygmatycznym przykładem ważności prawidłowego funkcjonowania systemów ICT była rakieta Ariane-5, która wybuchła w roku 1996 z powodu błędu w procedurze numerycznej, powodując utratę około 7 miliardów dolarów. W dzisiejszych czasach kryptowaluty są jeszcze bardziej aktualnymi przykładami, jak pokazał *atak DAO* (ang. *DAO attack*) na kryptowalutę Ethereum [**?**], w wyniku czego hakerowi udało się ukraść około 60 milionów dolarów.

Dziedzina *metod formalnych* (ang. *formal methods*) podchodzi do problemu stwierdzania, czy systemy ICT nie zawierają błędów w sposób *formalny*, tj. *matematyczny*. W podejściu *model checking*, system reprezentuje się matematycznie, a jego własność jako formułę pewnej logiki. Brak błędów systemu sprawdza się przez jego całkowitą eksplorację [**?**]. Najczęściej, model checking stosowano do analizy systemów skończenie-stanowych, ponieważ ten warunek gwarantuje, że procedura eksploracji zawsze się skończy. Z drugiej strony, w podejściu *theorem proving*, użytkownik musi konstruować ręcznie dowód poprawności systemu, który następnie jest weryfikowany przez algorytm. Dzięki temu, theorem proving można aplikować do systemów skończenie-stanowych, jak również do nieskończenie-stanowych.

Od lat 90. obserwuje się tendencję do poszerzania zakresu techniki model checkingu z systemów skończonych na nieskończone [**?**]. Systemy te modelowane są jako (nieskończone) *systemy*

tranzycyjne (ang. *transition systems*), to jest grafy $G = (S, \to)$, gdzie S to zbiór stanów a $\to \subseteq S \times S$ to relacja przejścia (ang. *transition relation*). W tym przeglądzie, skupimy się na następującym najbardziej podstawowym problemie algorytmicznym analizy systemów tranzycyjnych.

PROBLEM OSIĄGALNOŚCI(ang. *reachability problem*)
Wyjście: Skończona reprezentacja (potencjalnie nieskończonego) grafu $G = (S, \to)$ i dwa wierzchołki $s, t \in S$.
Wejście: Rozstrzygnij, czy $s \to^* t$?

(Powyżej, "$\to^*$" oznacza refleksyjne i przechodnie domknięcie relacji "$\to$", tak zwaną *relację osiągalności* (ang. *reachability relation*).) Problem osiągalności jest ważny dlatego, że przy jego zastosowaniu można udowodnić, że system nigdy nie osiągnie błędnej konfiguracji. Najważniejszym pytaniem jest, czy problem ten jest *rozstrzygalny*; innymi słowy, czy istnieje algorytm, który zawsze się kończy i odpowiada TAK wtedy, i tylko wtedy, gdy $s \to^* t$.

Rozstrzygalność: Czy problem osiągalności jest rozstrzygalny?

W ogólności, problem osiągalności jest nierozstrzygalny, na przykład dla maszyn Turinga [**?**]. Pomimo tego, istnieją nietrywialne klasy systemów nieskończenie-stanowych z rozstrzygalną osiągalnością. Systemy te są zbudowane z dyskretnych struktur danych, takich jak

1. jeden stos, jak w *automatach ze stosem* (ang. *pushdown automata*) [**?**, **?**];
2. nieujemne całkowite liczniki, jak w *sieciach Petriego* (ang. *Petri nets*) [**?**, **?**], które używane są do modelowania systemów współbieżnych;
3. kolejki FIFO (ang. *first-in first-out*), jak w *automatach komunikujących się* (ang. *communicating automata*) [**?**, **?**], które używane są w modelowaniu protokołów.

Rozstrzygalność jest tylko pierwszym etapem w zrozumieniu systemów nieskończenie-stanowych. Drugim najważniejszym pytaniem jest dokładna złożoność obliczeniowa problemu osiągalności, to znaczy, pytanie w której klasie złożoności znajduje się dany problem (np. LOGSPACE, PTIME, NPTIME, itd.)

Złożoność: Jaka jest złożoność obliczeniowa problemu osiągalności?

Znajomość dokładnej teoretycznej złożoności obliczeniowej niekoniecznie ma bezpośrednie implikacje dotyczące rozwiązywania instancji problemu, które występują w praktyce. Powstaje pytanie po co w takim razie badać złożoność teoretyczną. Udzielamy na nie trzech odpowiedzi: 1) jest to precyzyjnie zadane pytanie, więc posiada precyzyjną matematyczną odpowiedź; 2) zrozumienie teoretycznej złożoności może być traktowane jako równoważne ze zrozumieniem samego problemu; 3) określenie dokładnej złożoności obliczeniowej wymaga nowych matematycznych spostrzeżeń, które mogą być interesujące same w sobie i mieć praktyczne konsekwencje.

Wszystkie wyżej wymienione systemy są *dyskretne* w tym sensie, że przejście $s \to t$ nie może być dalej dekomponowane na mniejsze jednostki. Ten poziom abstrakcji nie jest dość precyzyjny, jeżeli chce się modelować bardziej ilościowe aspekty obliczeń, takie jak czas wymagany do wykonania przejścia. Najbardziej udanym modelem aspektów czasowych obliczeń są *automaty czasowe* (ang. *timed automata*; TA) [**?**]. Mimo wyższej siły ekspresji modelu TA, problem osiągalność jest dla nich rozstrzygalny, z optymalną złożonością PSPACE.

Te dwa kierunki badań (rozszerzanie skończonych automatów o dyskretne struktury danych, albo o ciągły czas) zostało połączone w wielu pracach w ciągu ostatnich dwóch dekad, na przykład w następujących pracach o *czasowych sieciach Petriego* (ang. *timed Petri nets*) [**?**, **?**], *systemach komunikujących się przez stratne kanały* (ang. *timed lossy channel systems*) [**?**], *automatach czasowych ze stosem* (ang. *timed pushdown automata*) [**?**] i *komunikujących się automatach czasowych* (ang. *timed communicating automata*) [**?**].

Plan. W tym rozdziale skupimy się na naszych nowych wynikach dotyczących rozstrzygalności i złożoności obliczeniowej czasowych automatów ze stosem i czasowych automatów komunikujących się. Dokładniej, w pracy [F] i [E] badamy rozstrzygalność i złożoność obliczeniową problemu osiągalności czasowych automatów komunikujących się dla *dowolnych topologii grafu komunikacji*. W pracy [C] i [D] proponujemy model który nazywamy *czasowymi automatami rejestrowymi ze stosem* (ang. *timed-register pushdown automata*; trPDA), jako fundamentalny model opisujący jednocześnie czas jak i rekursję pierwszego rzędu. Ogólnie mówiąc, osiągalność dla trPDA jest nierozstrzygalnym problemem, a jednak pokazujemy, że problem staje się rozstrzygalny dla pewnej podklasy trPDA (tzw. orbitowo-skończone, ang. *orbit-finiteness*) o dużej sile ekspresji. W pracy [B], pokazujemy, że w rzeczywistości możemy opuścić powyższe założenia techniczne, zachowując rozstrzygalność; osiągamy to poprzez odkrycie interesującego połączenia z modelem zwanym *rozgałęziającymi się systemami dodawania wektorowymi* (ang. *branching vector addition systems*; BVAS). W końcu, w pracy [A] rozszerzamy model *automatów czasowych ze stosem* (ang. *timed pushdown automata*; TPDA) [**?**] z wyrażalnymi ograniczeniami logicznymi i pokazujemy, że dla tego bardziej ogólnego modelu możemy rozwiązać problem osiągalności, a w rzeczywistości nawet bardziej ogólny problem osiągalności binarnej (por. rozdz. **??**). W dalszej części tego rozdziału bardziej szczegółowo opisujemy powyższe wyniki.

4.3.2 Czasowe automaty komunikacyjne [**F**]

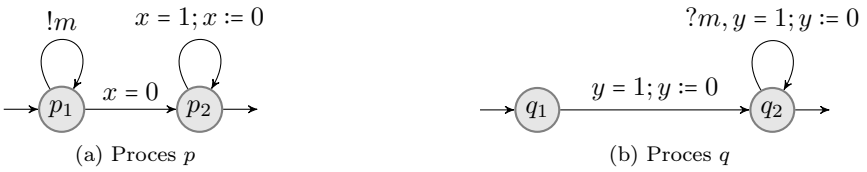
Automaty komunikujące się (ang. *communicating automata*; CA) to podstawowy model do badania równoczesnych procesów wymieniających wiadomości przez nieograniczone kanały [**?**, **?**]. Jednak model ten jest Turing-zupełny, a więc nawet podstawowe pytania weryfikacyjne, takie jak osiągalność, są nierozstrzygalne. Aby uzyskać rozstrzygalność, rozważono różne ograniczenia, w tym umożliwienie wysyłania tylko jednego rodzaju wiadomości [**?**], stratne kanały (ang. *lossy channels*) [**?**, **?**, **?**], lub ograniczenie do komunikacji półdupleksowej [**?**] (później uogólnione na mutex [**?**]). Rozstrzygalność można również uzyskać, gdy nakłada się dodatkowe warunki na obliczenie, o którego istnienie pytamy, takie jak ograniczenie na kontekstowe przełączanie (ang. *bounded context-switching*) [**?**], lub ograniczenie na wielkość kanałów. Wreszcie, i to jest ograniczenie, które nas interesuje, rozstrzygalność uzyskuje się przez ograniczenie topologii komunikacji. Mówimy, że topologia komunikacji to *polyforest*, jeśli nie zawiera cyklu nieskierowanego. Powszechnie wiadomo, że osiągalność jest rozstrzygalna (a właściwie PSpace-zupełna), wtedy i tylko wtedy, gdy topologia komunikacji to polyforest [**?**, **?**].

Pewne rozszerzenie automatów komunikujących, tj. automaty komunikujące się z ograniczeniami czasowymi (w postaci zegarów) zostało zbadane w [**?**], gdzie autorzy pokazują, że osiągalność jest rozstrzygalna dla dwu-procesowych topologii $p \to q$, a nierozstrzygalna dla trzy-procesowych topologii $p \to q \to r$. Powyższy wynik nierozstrzygalności zależy przede wszystkim od *pilnej semantyki* (ang. *urgent semantics*) dla odbiorów wiadomości, przy czym jeśli jakaś wiadomość może zostać odebrana przez proces, wszystkie inne akcje wewnętrzne są wyłączone. Daje to kolejną możliwość synchronizacji, która w połączeniu z niejawną synchronizacją przez upływ czasu, powoduje nierozstrzygalność.

Badamy *czasowe automaty komunikujące się* (ang. *timed communicating automata*; TCA), uogólniając je [**?**] do dowolnych topologii komunikacji. Pokazujemy, że pilna semantyka dla odbiorów wiadomości jest równoważna *testom pustości* (ang. *emptiness tests*) kanałów, pozwalając nam wyrazić nasze wyniki na dowolnych topologiach komunikacyjnych w jednolity sposób. Pierwszym głównym wynikiem jest pełna charakteryzacja rozstrzygalnych topologii komunikacyjnych przy założeniu czasu dyskretnego (por. [Twierdzenie 3, F]).

Theorem 1 (TCA rozstrzygalność [F]) *Problem osiągalności dla* TCA *w dyskretnym czasie jest rozstrzygalny, wtedy i tylko wtedy, gdy topologia komunikacji to polyforest i, dodatkowo, w każdej jego silnie spójnej składowej istnieje co najwyżej jeden kanał, dla którego można testować pustość.*

W bardziej ogólnym czasie gęstym, pokazujemy rozstrzygalność w przypadku bez testu na pu-



(a) Proces p (b) Proces q

Figure 1: Kanały czasowych automatów komunikacyjnych nie mogą być ograniczone.

stość (por. [Twierdzenie 5, F])[1]). Wynik ten rozszerza poprzednią pracę [**?**] odnośnie zarówno rozstrzygalnych, jak i nierozstrzygalnych topologii. Po pierwsze, nasza charakteryzacja powyżej znacznie rozszerza klasę rozstrzygalnych topologii: Nie tylko $p \to q$ jest rozstrzygalne, ale także, na przykład, $p \to q \to r$ (i $p \to q \leftarrow r$) pod warunkiem, że co najwyżej jeden z dwóch kanałów ma testy na pustość (równoważnie, pilną semantykę). Po drugie, nasza charakteryzacja rozszerza również klasę nierozstrzygalnych topologii: Na przykład, topologia czterech procesów $p \to q \to r \to s$ jest nierozstrzygalna, gdy dwa dowolne kanały dopuszczają testy na pustość (równoważnie, pilną semantykę).

Aby przekazać intuicję, które trudności techniczne należy pokonać, aby uzyskać rozstrzygalność, rozważmy prostą (rozstrzygalną) topologię komunikacji $p \to q$. W kontekście bezczasowym, rozstrzygalność problemu osiągalności wynika z prostej obserwacji, że możemy ograniczyć naszą uwagę do harmonogramów natychmiast dopasowujących każdą transmisję $!m$ procesu p z odpowiednim odbiorem $?m$ procesu q. W ten sposób długość kanału jest ograniczona. W wyniku tej obserwacji, otrzymamy skończony system stanowy równoważny pierwotnemu pod względem osiągalności. W ustawieniu (dyskretnym) czasowym ta technika nie jest już poprawna. Dla ilustracji, rozważmy dwa procesy p i q; por. Fig. **??**. Dwa zegary x, y to początkowo 0. Proces p rozpoczyna się w stanie p_1 i stamtąd może wysłać nieograniczoną liczbę wiadomości m do procesu q. W pewnym momencie, proces p może przejść do stanu p_2, *bez upływu czasu* zgodnie z warunkiem $x = 0$. Proces q rozpoczyna się w stanie q_1, czeka jedną jednostkę czasu i przechodzi do stanu q_2 (zgodnie z warunkiem $y = 1$), a zegar y jest zresetowany ($y := 0$). Ze stanu q_2, proces q może odczytywać jedną wiadomość $?m$ co jednostkę czasu. Najważniejsze jest to, że proces q musi czekać jedną jednostkę czasu, zanim będzie mógł odebrać jakąkolwiek wiadomość; proces p przeciwnie, może wysłać nieograniczoną liczbę wiadomości w pierwszej jednostce czasu. W związku z tym, nie możemy ograniczyć rozmiaru kanałów. Jest to zasadnicza trudność w analizie systemów zawierających kanały i ograniczenia czasowe.

Jeśli zabronimy testów na pustość, to nasza charakteryzacja z Twierdzenia **??** jest taka sama, jak w ustawieniu bezczasowym. Jednak w ustawieniu czasowym złożoność obliczeniowa problemu osiągalności się pogarsza, już w czasie dyskretnym. To jest nasz drugi główny wynik (por. [Wniosek 1, F]).

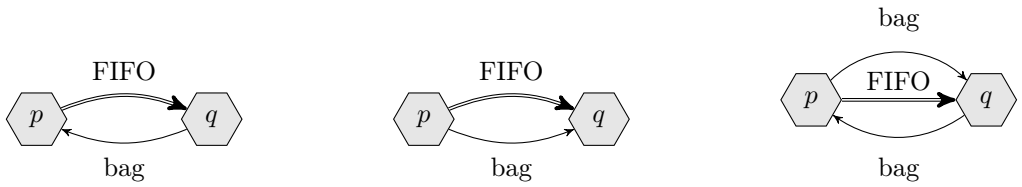
Theorem 2 (TCA złożoność [F]) *Złożoność obliczeniowa problemu osiągalności* TCA *jest nieelementarna, już w czasie dyskretnym.*

Powyższa granica dolna jest pokazana za pomocą połączonych topologii symulujących sieci Petriego/maszyny licznikowe i korzysta z ostatniego wyniku [**?**]. Chociaż powyższa złożoność może wydawać się zniechęcająca, to jeśli zabronimy testów na pustość i zignorujemy wartość zegarów na końcu wykonania, problem jest w ExpSpace (poprzez redukcję do problemu pokrywalności sieci Petriego, który jest w ExpSpace [**?**]).

4.3.3 Automaty komunikacyjne z kanałami FIFO i bag [E]

W pracy [E] badamy pewne uogólnienie TCA na dyskretny czas. Naszy dowód Twierdzenia **??** [F]

[1] W jeszcze nieopublikowanej pracy udowodniliśmy, że Twierdzenie **??** jest prawdziwe nawet w gęstym czasie [**?**].



(a) Przeciwległe kanały (rozstrzygalny) (b) Równoległe kanały (rozstrzygalny) (c) Oba rodzaje kanałów (nierozstrzygalny)

pokazuje, że możemy symulować TCA w dyskretnym czasie z bezczasowymi CA, wprowadzając dodatkowe kanały komunikacji: Dokładniej, każdy TCA kanał $p \to q$ może być symulowany przez dwa CA (bezczasowe) kanały, jeden FIFO $p \to q$ i drugi "bag" $q \to p$ w przeciwnym kierunku. *Bag* to taki kanał, w którym wiadomości można odbierać w dowolnej kolejności, a zatem jest słabszy niż kanał FIFO. Pokazujemy, że wynikowa topologia, pokazana na Rys. **??**, ma rozstrzygalny problem osiągalności. Z drugiej strony, jeśli odwrotny kanał $q \to p$ jest FIFO, to uzyskujemy nierozstrzygalną topologię, ponieważ dwa przeciwnie skierowane kanały tworzą nieskierowany cykl FIFO (ang. *undirected FIFO cycle*) (tj. nie polyforest [**?**, **?**]).

To oczywiście rodzi następne pytanie: Które topologie komunikacyjne z kanałami FIFO i typu bag są rozstrzygalne[2]? Motywacja studiowania takich mieszanych topologii nie ogranicza się do symulacji dyskretnego czasu. Kanały typu bag mogą modelować asynchroniczne wywołania procedur, które mogą być dowolnie porządkowane [**?**, **?**, **?**]. Ponadto, kanały typu bag oferują nietrywialną górną aproksymację kanałów FIFO, przekształcając nierozstrzygalne topologie FIFO w rozstrzygalne topologie bag. Na przykład, udowodnimy, że Rys.**??** i **??** są rozstrzygalne; jeżeli oba kanały byłyby FIFO, to topologie te stałyby się nierozstrzygalne. Mamy również nietrywialne przykłady nierozstrzygalnych topologii z kanałami FIFO/bag: Poprzednie dwa przykłady są maksymalne, ponieważ pokazujemy, że ich połączenie daje nierozstrzygalną topologię; por. Rys. **??**.

Naszym głównym wynikiem jest pełna charakteryzacja topologii $\mathcal{T}$ mających rozstrzygalną osiągalność. (W poniższym stwierdzeniu, P jest rozstrzygalną właściwością topologii obejmujących określone cykle kanałów FIFO i bag.)

Theorem 3 (Charakteryzacja rozstrzygalnych topologii [E]) *Problem osiągalności* CA *na topologii $\mathcal{T}$ kanałów FIFO i bag jest rozstrzygalny, wtedy i tylko wtedy, gdy $P(\mathcal{T})$.*

Zauważymy, że topologia zawierająca tylko kanały typu bag jest efektywnie równoważna sieciom Petriego, a zatem osiągalność jest rozstrzygalna. W ten sposób zawsze można aproksymować z góry CA FIFO zmieniając wszystkie kanały FIFO w typ bag. Dzięki naszej charakteryzacji można uzyskać znacznie dokładniejszą analizę przez selektywne górne przybliżanie tylko niektórych kanałów FIFO (przy zachowaniu rozstrzygalności).

4.3.4 Czasowe automaty rejestrowe ze stosem (ang. *timed-register pushdown automata*) [B], [C], [D]

W pracach [B], [C] i [D] badamy problem wprowadzenia ograniczeń czasowych do automatów ze stosem. Wcześniejsza próba w literaturze przedmiotu to tak zwane *automaty czasowe ze stosem* (ang. *pushdown timed automata*; PDTA) [**?**], które rozszerzają klasyczne automaty ze stosem zastępując skończony mechanizm sterowania automatem czasowym. Stos w PDTA jest więc po prostu klasycznym stosem (bez czasu), co poważnie ogranicza moc wyrażania modelu. Na przykład, PDTA nie może rozpoznać naturalnych czasowych języków bezkontekstowych takich jak *czasowe palindromy* (ang. *timed palindromes*) (poniżej, w^R jest odwróceniem w):

$$L = \{ww^R \mid w \in (\Sigma \times \mathbb{R})^*\}. \tag{1}$$

[2]Podobny problem topologii połączenia kanałów FIFO i stratnych został zbadany w [**?**].

Przez jakiś czas wydawało się, że tę sytuację mogą poprawić tak zwane *automaty ze stosem czasu gęstego* (ang. *dense-timed pushdown automata*; DTPDA) [**?**], które rozszerzają PDTA o mechanizm zegarów stosowych, które można kłaść i zdejmować ze stosu zgodnie z niediagonalnymi ograniczeniami przedziałowymi (ang. *non-diagonal interval constraints*). Można powiedzieć, że DTPDA są „nieskończone w dwóch wymiarach", ponieważ stos jest nieograniczony, a każdy symbol stosu zawiera zegar, który również jest nieograniczony. Nasza praca rozpoczęła się od obserwacji, że DTPDA redukuje się semantycznie do PDTA, w tym sensie, że oba modele są efektywnie równoważne co do rozpoznawanej klasy języków czasowych (por. [Twierdzenie II.1, C]).

Theorem 4 (DTPDA=PDTA [C]) *Dla każdego* DTPDA *można efektywnie skonstruować* PDTA *rozpoznający ten sam język czasowy.*

Ten nieco zaskakujący wynik jest konsekwencją nieoczekiwanego wzajemnego oddziaływania między monotonicznością czasu a zagnieżdżoną (ang. *well-nested*) dyscypliną stosu. W istocie pokazujemy, że to samo zachodzi nawet wtedy, gdy zezwala się na ograniczenia zegarów bardziej liberalne niż w DTPDA. Rodzi się pytanie, czy istnieje sensowne czasowe rozszerzenie automatów ze stosem o mocy wyrażania większej niż DTPDA. To jest punkt wyjścia naszego badania.

4.3.5 Zbiory z atomamy, część I: Oligomorficzne i homogeniczne atomy [D]

Jedną z odpowiedzi na powyższy problem można znaleźć w podejściu opartym na *zbiorach z atomami* (ang. *sets with atoms*), które stanowi jednolity sposób rozszerzenia teorii automatów (i, ogólniej, matematyki) przez uogólnienie pojęcia skończoności do *orbitowej skończoności* (ang. *orbit-finiteness*; [**?**, **?**]). To podejście jest sparametryzowane strukturą relacyjną *atomów* $\mathbb{A}$. Najprostszym przykładem atomów są *atomy równościowe* $(\mathbb{N}, =)$ (ang. *equality atoms*), gdzie atomy stanowią dowolny przeliczalny zbiór, a jedynym symbolem relacyjnym w sygnaturze jest relacja równości. Bogatszą strukturę ma tak zwany *(gęsty) porządek liniowy* $(\mathbb{Q}, \leq)$, który można wykorzystać do modelowania jakościowego pojęcia gęstego czasu. W dalszej części rozdziału użyjemy atomów porządku liniowego jako ilustracyjnego przykładu.

Kluczową intuicją jest to, że dwa zbiory z atomami są nierozróżnialne jeśli istnieje *automorfizm* struktury atomów[3] mapujący jeden zbiór na drugi; mówimy wtedy, że zbiory te są w tej samej *orbicie* (ang. *orbit*). Na przykład, automorfizmy atomów porządku liniowego są dokładnie monotonicznymi bijekcjami struktury $\mathbb{Q}$; dwie krotki $\bar{a} = (2, 1.1, 2)$ i $\bar{b} = (3, 2.9, 3)$ są nierozróżnialne ponieważ istnieje automorfizm $\alpha : \mathbb{Q} \to \mathbb{Q}$ rozszerzający $[2 \mapsto 3, 1.1 \mapsto 2.9]$ o tej własności, że $\alpha(\bar{a}) = \bar{b}$; w tym przypadku, orbitą $\bar{a}$ jest dokładnie $\{(a, b, c) \in \mathbb{Q}^3 \mid a = c, b < a\}$.

Kluczowym pojęciem w teorii zbiorów z atomami jest orbitowa skończoność, która jest uogólnieniem skończoności: Zbiór z atomami jest *orbitowo skończony* (ang. *orbit finite*) jeśli można go podzielić na skończenie wiele orbit. Innymi słowy, zbiór orbitowo skończony jest skończony z dokładnością do automorfizmów atomów. Na przykład, $\mathbb{Q}$ i $\{(a, b) \in \mathbb{Q}^2 \mid a \neq b\}$ są orbitowo skończone: Ten pierwszy ma dokładnie jedną orbitę, a drugi ma dwie orbity, $\{(a, b) \in \mathbb{Q}^2 \mid a < b\}$ i $\{(a, b) \in \mathbb{Q}^2 \mid a > b\}$.

W naszym kontekście, interesuje nas uogólnienie automatów ze stosem do teorii zbiorów z atomami. Można to osiągnąć poprzez uogólnienie wszystkich komponentów zwykłej podręcznikowej definicji automatów ze stosem (tj. alfabetu wejściowego i stosowego, stanów i relacji przejścia) ze skończonych zbiorów do zbiorów z atomami definiowalnych w logice pierwszego rzędu. Powstały w ten sposób model nazywamy *definiowalnymi automatami ze stosem* (ang. *definable pushdown automata*; DPDA). Każdy wybór struktury atomów $\mathbb{A}$ daje inne pojęcie automatu DPDA. Na przykład, dla atomów równościowych $\mathbb{A} = (\mathbb{N}, =)$ uzyskujemy model automatów rejestrowych ze stosem zbadanych w [**?**]. Inne możliwości wyboru atomów zostaną opisane poniżej. Przeanalizowaliśmy problem osiągalności dla definiowalnych automatów ze stosem dla dobrze zachowujących się klas atomów. Mówimy, że struktura relacyjna $\mathbb{A}$ jest *oligomorficzna* (ang. *oligomorphic*) jeśli zbiór $\mathbb{A}^k$ jest orbitowo-skończony dla dowolnego k [**?**]; ta sama struktura jest *rozstrzygalna w*

[3] Automorfizmem struktury relacyjnej $\mathbb{A}$ nazywamy bijekcję na jej dziedzinie, która zachowuje i odbija interpretację wszystkich jej symboli relacyjnych.

pierwszym rzędzie (ang. *(first-order) decidable*) jeśli problem spełnialności dla teorii pierwszego rzędu struktury $\mathbb{A}$ jest rozstrzygalny. Na przykład, atomy równościowe i atomy porządku liniowego są oligomorficzne i rozstrzygalne; później podamy przykład struktury nieoligomorficznej. Naszym pierwszym wynikiem jest rozstrzygalność problemu osiągalności dla DPDA z oligomorficznymi, rozstrzygalnymi atomami (por. [Twierdzenie 4, D]).

Theorem 5 (Oligomorficzne DPDA [D]) *Niech $\mathbb{A}$ będzie rozstrzygalną oligomorficzną strukturą relacyjną. Problem osiągalności jest rozstrzygalny dla definiowalnych automatów ze stosem z atomami $\mathbb{A}$.*

(W istocie udowodniliśmy o wiele ogólniejszy wynik, mianowicie, że przy założeniach z treści twierdzenia DPDA dopuszczają korzystanie z klasycznej procedury saturacji opartej na automatach orbitowo-skończonych [**?**].) Powyższy wynik jest bardzo ogólny, ponieważ wymaga jedynie, aby problem spełnialności teorii pierwszego rzędu był rozstrzygalny dla atomów, bez założenia o jego złożoności. W szczególności, nie można podać górnej granicy złożoności bez dalszych założeń.

Mówimy, że struktura relacyjna $\mathbb{A}$ jest *homogeniczna* (ang. *homogeneous*) jeśli każdy częściowy izomorfizm podstruktur skończonych rozszerza się na automorfizm całej struktury [**?**]. Homogeniczność ma tę przyjemną konsekwencję, że liczba orbit zbioru $\mathbb{A}^k$ jest wykładniczo ograniczona[4]. W szczególności, homogeniczne struktury są oligomorficzne. (Istnieją oligimorficzne struktury relacyjne, które nie są homogeniczne, takie jak atomy wektorów bitowych [**?**].) Naszym drugim wynikiem jest to, że problem osiągalności automatów DPDA jest w klasie FPT (ang. *fixed-parameter tractable*) nad homogeniczymi strukturami, co znaczy, że jego czas działania ma postać $O(f(k) \cdot \mathsf{poly}(n))$, gdzie n jest rozmiarem automatu DPDA a k jest jego wymiarem (por. [Twierdzenie 7, Wniosek 8, D]).

Theorem 6 (Homogeniczne DPDA [D]) *Niech $\mathbb{A}$ będzie homogeniczną strukturą relacyjną. Problem osiągalności dla* DPDA *z atomami z struktury $\mathbb{A}$ jest należy do klasy* FPT*, z wymiarem automatu jako parametrem.*

Naszym trzecim wynikiem jest to, że w przypadku powszechnie występujących struktur można osiągnąć jeszcze więcej. Niech $\mathbb{A}$ i $\mathbb{B}$ będą dwiema strukturami relacyjnymi. Mówimy, że $\mathbb{A}$ jest *podstrukturą* (ang. *substructure*) struktury $\mathbb{B}$, jeśli mają tę samą sygnaturę, a dziedzina struktury $\mathbb{A}$ jest zawarta w dziedzinie struktury $\mathbb{B}$; jeśli dodatkowo interpretacja wszystkich symboli relacyjnych zgadza się na dziedzinie struktury $\mathbb{B}$, to mówimy, że $\mathbb{A}$ jest *indukowaną podstrukturą* (ang. *induced substructure*) struktury $\mathbb{B}$. Problem indukowanej podstruktury polega na znalezieniu odpowiedzi na pytanie, czy struktura $\mathbb{A}$ jest indukowaną podstrukturą struktury $\mathbb{B}$. Nasza obserwacja jest taka, że problem indukowanej podstruktury jest w klasie PTime, tj. jest rozwiązywalny w czasie wielomianowym, dla następujących powszechnie występujących homogenicznych struktur relacyjnych ([Rozdział 6, D]; por. [**?**]):

- Atomy równościowe $(\mathbb{N}, =)$: Można ich użyć do modelowania wartości danych z nieskończonego zbioru, które można porównać tylko na równość.
- Atomy liniowego porządku $(\mathbb{Q}, \leq)$: Jak wspomniano powyżej, mogą być użyte do modelowania zdarzeń czasowych.
- Atomy z relacją pomiędzy $(\mathbb{Q}, B)$, gdzie $(x, y, z) \in B$ wtedy i tylko wtedy, gdy $y < x < z$ lub $z < x < y$: Mogą być użyte do modelowania czasowych zdarzeń gdzie można obserwować to, czy zdarzenie x występuje pomiędzy dwoma innymi zdarzeniami y i z, ale nie ma dalszych informacji o kolejności samych y i z.
- Atomy porządku cyklicznego $(\mathbb{Q}, K)$, gdzie $(x, y, z) \in K$ wtedy i tylko wtedy, gdy $x < y < z$, lub $y < z < x$ lub $z < x < y$: Można ich użyć do modelowania kolejności ułamkowych wartości

[4]Dwie krotki $\bar{a}, \bar{b} \in \mathbb{A}^k$ są izomorficzne dokładnie wtedy, gdy spełniają te same relacje z struktury $\mathbb{A}$; zatem są $O(2^{\mathsf{poly}(k)})$ równoważności klas krotek. Ponieważ każdy częściowy izomorfizm rozszerza się na automorfizm struktury $\mathbb{A}$, to taka sama liczba ogranicza z góry liczbę orbit zbioru $\mathbb{A}^k$.

znaczników czasu (ang. *timestamps*) zdarzeń. Jest to naturalny wybór, ponieważ relacja K jest niezmienna w czasie: $(x, y, z) \in K$ wtedy i tylko wtedy, gdy $(x \oplus \delta, y \oplus \delta, z \oplus \delta)$ dla dowolnego $x, y, z \in [0, 1)$ i $\delta \in \mathbb{R}$, gdzie $a \oplus b$ jest ułamkową częścią wartości $a + b$.

- Atomy porządku częściowego: Mogą być używane do opisywania zdarzeń, które mogą wystąpić niezależnie od siebie, ale nadal muszą spełniać pewne ograniczenia przyczynowości.
- Atomy porządku drzewiastego: Mogą być używane do modelowania dynamicznego tworzenia procesów, które nie będą dalej oddziaływać w przyszłości.
- Inne przykłady to: atomy równoważności, preporządku, grafu i turnieju.

Poniższy wynik dotyczy każdego z tych przykładów (por. [Wniosek 10, D]).

Theorem 7 (Efektywne homogeniczne DPDA [D]) *Niech $\mathbb{A}$ będzie homogeniczną strukturą relacyjną, dla której problem indukowanej podstruktury jest w klasie* PTime. *Problem osiągalności dla definiowalnych automatów ze stosem z atomami z struktury $\mathbb{A}$ jest w klasie* ExpTime. *Co więcej, ten problem jest* ExpTime*-trudny już dla atomów równościowych $(\mathbb{N}, =)$.*

Powyższa lista struktur atomów pokazuje szerokie zastosowanie Twierdzenia **??**[5]. Głównym ograniczeniem wszystkich powyższych struktur jest to, że są zasadniczo *jakościowe*, a zatem nie są w stanie modelować bardziej ilościowych aspektów zdarzeń, takie jak ich dokładna długość. Następny rozdział jest poświęcony metodom poprawienia tej sytuacji.

4.3.6 Zbiory z atomami, część II: Atomy czasowe [C]

Aby modelować bardziej ilościowe cechy systemów czasu rzeczywistego (ang. *real-time systems*), rozważamy bogatsze struktury, takie jak (w rosnącej kolejności mocy wyrażania)

- atomy *czasu dyskretnego* $(\mathbb{Z}, +1, \leq)$ (ang. *discrete time atoms*);
- atomy *czasu gęstego* $(\mathbb{Q}, +1, \leq)$ (ang. *dense time atoms*);
- atomy *czasowe hybrydowe* $\mathbb{H} = (\mathbb{Z}, +1, \leq) \uplus (\mathbb{Q}, \leq)$ (ang. *hybrid time atoms*), dwusortowa (ang. *two-sorted*) struktura z dyskretnym składnikiem (dla modelowania całkowitych wartości znaczników czasu) i z gęstym składnikiem (dla modelowania części ułamkowych).

Łącznie nazywamy powyższe struktury *atomami czasowymi* (ang. *timed atoms*), a definiowalne automaty ze stosem na czasowych atomach nazywamy *nieograniczonymi czasowymi automatami rejestrowymi ze stosem* (nieograniczone trPDA). Żadna ze struktur atomów czasowych nie jest oligomorficzna (ani homogeniczna)[6], a nasza ogólna technika z Rozdz. **??** nie może być zastosowana do trPDA. Co gorsza, problem osiągalności dla definiowalnych automatów skończonych nad czasowymi atomami jest w istocie nierozstrzygalny już nad atomami czasu dyskretnego w wymiarze 3: Zbiór stanów jest podzbiorem $\mathbb{Z}^3$, a składniki (x, y, z) symulują dwa liczniki maszyny Minsky'ego (która ma nierozstrzygalny problem osiągalności) [**?**]): Zwiększanie/zmniejszanie liczników jest symulowane definiowalnymi przejściami $(x, y, z) \mapsto (x \pm 1, y, z)$ dla pierwszego licznika i $(x, y, z) \mapsto (x, y \pm 1, z)$ dla drugiego; test na zero odpowiednio przez $x = z$ lub $y = z$.

Źródłem nierozstrzygalności jest to, że można zapamiętać znaczniki czasu, które mogą być dowolnie oddalone od siebie (takie jak x i z powyżej). Dla definiowalnych automatów ten problem jest natychmiast rozwiązywany przez wymaganie, aby zbiór stanów był orbitowo skończony. Powstający w ten sposób model automatów orbitowo skończonych nad atomami czasowymi ma rozstrzygalny problem osiągalności [**?**]. Nie jest to jednak wystarczające dla nieograniczonych

[5] Ograniczenie z góry złożoności problemu indukowanej substruktury jest koniecznym założeniem, ponieważ można skonstruować homogeniczne struktury, dla których problem indukowanej podstruktury może symulować problem należenia (ang. *membership problem*) dla dowolnych podzbiorów zbioru $\mathbb{N}$ [Twierdzenie 9, D].

[6] Na przykład, już $\mathbb{Z}^2$ zawiera nieskończenie wiele orbit: dla każdej liczby całkowitej k, zbiór $\{(x, x + k) \mid x \in \mathbb{Z}\}$ jest odrębną orbitą zbioru $\mathbb{Z}^2$. Wynika to z faktu, że automorfizmy struktury $\mathbb{Z}$ muszą zachować odległość między punktami, a zatem nie istnieje wystarczająco wiele automorfizmów.

trPDA, ponieważ wciąż można zakodować dwa liczniki Minsky'ego, nawet dla orbitowo skończonych stanów: Pierwszy licznik jest zakodowany klasycznie jako wysokość stosu, a drugi jako różnica między wartością rejestru kontrolnego a wartością na szczycie stosu (por. [Twierdzenie IV.1, C]).

Theorem 8 (trPDA [C]) *Problem osiągalności dla nieograniczonych* trPDA *jest nierozstrzygalny.*

Powyższa symulacja jest możliwa dzięki temu, że operacje wkładania na stos mogą widzieć wartość na szczycie stosu. Jako rozwiązanie, zabraniamy operacji wkładania na stos z jednoczesnym odczytaniem wartość na szczycie stosu. Uzyskany model poniżej nazywamy trPDA. O ekspresywności trPDA świadczy fakt, że rozpoznają one palindromy czasowe (**??**), co można zrobić w następujący sposób. Automat działa w dwóch fazach. W pierwszej fazie, automat odczytuje symbol wejściowy postaci $(a, x) \in \Sigma \times \mathbb{R}$ i wkłada go na stos. Następnie, automat niedeterministycznie odgaduje środek słowa, i przechodzi do drugiej fazy, w której odczytuje (a, x) i jednocześnie zdejmuje (a, y) ze stosu, pod warunkiem że $x = y$. Moc wyrażania trPDA wykracza nawet poza języki bezkontekstowe: rozważmy język bezczasowych palindromów nad alfabetem $\Sigma = \{a, b\}$ zawierających tę samą liczbę symbolów a i b:

$$M = \{ww^R \mid w \in \Sigma^*, \#_a(w) = \#_b(w)\}. \tag{2}$$

Język M jest rozpoznywany przez trPDA w następujący sposób: Początkowo, automat wkłada $(\perp, x)$ na stos i pamięta x w stanie, gdzie $x \in \mathbb{Z}$ jest zgadywaną liczbą całkowitą, która zostanie użyta na końcu biegu automatu. Pozostała część jest podobna do palindromów czasowych; dodatkowo, lokalny rejestr x jest zwiększany/zmniejszany o 1 po odczytaniu a/b. Pod koniec biegu, aby sprawdzić, czy słowo zawiera tę samą liczbę symboli a i b, automat wyrzuca $(\perp, x)$ ze stosu tylko wtedy, gdy x jest taki sam jak lokalny rejestr.

Pomimo tego, że języki trPDA wykraczają klasę języków bezkontekstowych, naszym głównym wynikiem jest to, że trPDA są rozstrzygalne (por. [Twierdzenie 1, B]).

Theorem 9 (trPDA [B]) *Problem osiągalności dla* trPDA *jest rozstrzygalny, ze złożonością* 2ExpTime.

Powyższy wynik uzyskuje się poprzez wskazanie ścisłego połączenia między trPDA a tak zwanymi *systemami dodawania wektorów rozgałęziających* (ang. *branching vector addition system*) z dodawaniem i odejmowaniem ($\mathbb{Z}$-BVASS) [**?**]. Ten ostatni model jest ekspresyjnym uogólnieniem sieci Petriego / automatów z licznikami, który w swoim podstawowym wariancie z dodawaniem (bez odejmowania) jest rozstrzygalny w PTime dla wymiaru 1 [**?**]; dla wyższych wymiarów, jest to od dawna otwarty problem [**?**]. Nasz główny wynik techniczny prowadzący do Twierdzenia **??** jest taki, że $\mathbb{Z}$-BVASS są rozstrzygalne, w rzeczywistości elementarne (por. [Twierdzenie 6, B]).

Theorem 10 ($\mathbb{Z}$-BVASS [B]) *Problem osiągalności modelu* $\mathbb{Z}$-BVASS *dla wymiaru* 1 *jest rozstrzygalny, ze złożonością* ExpTime.

Pokazujemy, że złożoność poprawia się dla następujących stopniowo coraz mniej ekspresywnych podklas modelu trPDA. W tak zwanym *orbitowo skończonym* trPDA (ang. *orbit-finite* trPDA) wymagamy, aby połączony zbiór lokalizacji kontrolnych i symboli na szczycie stosu jest orbitowo skończony. W ramach tego ograniczenia, udowodniliśmy poprawione ograniczenie górne na złożoność problemu ościągalności (por. [**?**] Twierdzenie IV.8).

Theorem 11 (orbitowo-skończone trPDA [C]) *Problem osiągalności modelu orbitowo skończonego jest rozstrzygalny w* NExpTime *i* ExpTime*-trudny.*

Co więcej, jeśli całkowicie wyeliminujemy informacje o czasie ze stosu, uzyskujemy jeszcze lepszą złożoność (por. [Twierdzenie IV.5, C]). To jest szczególnie interesujące, ponieważ trPDA z bezczasowym stosem wystarczają do symulowania modeli dtPDA and PDTA [Wniosek IV.10, C], i w tym przypadku złożoność obliczeniowa jest optymalna, ponieważ te ostatnie modele są już ExpTime-trudne[**?**].

Theorem 12 (trPDA z bezczasowym stosem [C]) *Problem osiągalności modelu* trPDA *z bezczasowym stosem jest* ExpTime*-zupełny.*

Wreszcie, i może to być najciekawsza podklasa modelu trPDA, model trPDA z *czasem monotonicznym* jest również ExpTime-zupełny [**?**, Wniosek 6.8].

Uważamy, że trPDA stanowią ekspresywną, a zarazem rozstrzygalną klasę systemów czasowych o nieskończenie wielu stanach. Otrzymano ją odchodząc od standardowego podejścia opartego na zegarach i zastępując je modelem opartym na atomach. Pozostaje pytanie, czy uda nam się znaleźć ekspresywny model automatów czasowych ze stosem w dziedzinie klasycznych zegarów czasu rzeczywistego. Tym zajmujemy się w następnym rozdziale.

4.3.7 Automaty czasowe ze stosem [A]

Twierdzenie **??** pokazuje, że tradycyjne ograniczenia zegarowe (ang. *clock constraints*) nie są wystarczające do wykorzystania siły wyrazu stosu czasowego. Rozważmy dla przykładu następujący język palindromów czasowych o nieparzystej długości nad alfabetem $\Sigma = \{a, b\}$, gdzie odległość między odpowiadającymi sobie symbolami jest całkowita:

$$N = \{(a_0, x_0)\cdots(a_{2n}, x_{2n}) \in (\Sigma \times \mathbb{R})^* \mid \forall (0 \le i \le n) \cdot a_i = a_{2n-i} \wedge x_i - x_{2n-i} \in \mathbb{N}\}. \tag{3}$$

Jest oczywiste, że potrzebujemy stosu czasowego, aby rozpoznać N, ponieważ musimy zapamiętać część ułamkową wartości x_i, którą wkładamy na stos w pierwszej części biegu, w celu sprawdzenia, czy jest ona równa części ułamkowej wartości x_{2n-i} w drugiej części biegu. W Rozdz. **??**, zajęliśmy się tym problemem, wprowadzając automaty czasowo-rejestrowe ze stosem trPDA. Powstaje jednak pytanie, czy klasyczny model oparty na zegarach może rozpoznać prawdziwie czasowe języki bezkontekstowe, takie jak N.

Pokazujemy, że tak właśnie jest. Wprowadzamy tak zwane *automaty czasowe ze stosem* (ang. *timed pushdown automata*; TPDA), model rozszerzający klasyczne (bezczasowe) automaty ze stosem o zegary, które można odłożyć na stos i zdjąć ze stosu. Ponieważ stos jest nieograniczony, TPDA są „nieskończone w dwóch wymiarach". W miarę upływu czasu wszystkie zegary zwiększają swoje wartości w tym samym tempie. Zegary można porównywać używając kombinacji logicznych ograniczeń klasycznych i ułamkowych[7]:

	(klasyczne)	(ułamkowe)
(nie-przekątniowe)	$x \le k, \quad k \in \mathbb{Z}$	$\{x\} = 0,$
(przekątniowe)	$x - y \le k, \quad k \in \mathbb{Z}$	$\{x\} \le \{y\}.$

Lokalne zegary można zerować i porównywać zgodnie z powyższymi ograniczeniami. W czasie operacji odkładania na stos tworzone są nowe zegary, których wartości są inicjalizowane (być może niedeterministycznie) tak, aby spełniały zadane ograniczenia pomiędzy zegarami na szczycie stosu i zegarami lokalnymi; podobnie operacja zdjęcia ze stosu wymaga, aby zegary na szczycie stosu, które zostaną zdjęte, spełniały pewne ograniczenia analogicznej postaci.

Nie można usunąć informacji o czasie ze stosu czasowego modelu TPDA (w przeciwieństwie do dtPDA; zob. podobną obserwację w [**?**]), ponieważ możemy skonstruować TPDA rozpoznający język czasowy N zdefiniowany powyżej: Jest jeden zegar lokalny x, który początkowo ma wartość 0 i nigdy nie jest zerowany. W pierwszej fazie automat odczytuje symbole wejściowe $c \in \Sigma$ i odkłada je na stos czasowy wraz z nowotworzonym zegarem stosowym y spełniającym ograniczenie przekątniowe $x = y$. W drugiej fazie automat odczytuje symbole $c \in \Sigma$ i zdejmuje (c, y) ze stosu (gdzie y jest zegarem na stosie powiązanym z symbolem stosowym c) pod warunkiem, że y ma tę samą wartość ułamkową co zegar lokalny x, tj. $\{x\} = \{y\}$.

W kontekście modelu TPDA, naturalne jest ograniczenie relacji osiągalności do konfiguracji z pustym stosem. Niech X będzie zbiorem zegarów, niech L będzie skończonym zbiorem stanów

[7] Oznaczamy jako $\{x\}$ część ułamkową wartości zegara x.

kontrolnych, i niech $\mathbb{R}^X_{\geq 0}$ będzie zbiorem wartościowań zegarów (ang. *clock valuations*). W pozostałej części tego rozdziału *relacją osiągalności* będziemy nazywać rodzinę relacji $\{\rightsquigarrow_{pq}\}_{p,q\in L}$ zdefiniowanych w następujący sposób: Dla dowolnych $\mu, \nu \in \mathbb{R}^X_{\geq 0}$ i $p, q \in L$,

$$\mu \rightsquigarrow_{pq} \nu$$

wtedy i tylko wtedy, gdy istnieje bieg zaczynający się od stanu kontrolnego p, zegarów lokalnych o wartościach żądanych przez μ i pustego stosu, a kończący się w stanie kontrolnym q, zegarach lokalnych o wartościach zadanych przez ν i pustym stosie. Dla ustalonych stanów kontrolnych p, q, relacja osiągalności $\rightsquigarrow_{pq}$ jest podzbiorem $\mathbb{R}^X_{\geq 0} \times \mathbb{R}^X_{\geq 0}$ i dlatego może być opisana formułą logiczną nad liczbami rzeczywistymi. Rozważamy tak zwany *problem osiągalności binarnej* (ang. *binary reachability problem*) dla TPDA, który jest bardziej ogólny niż zwykła osiągalność: Zamiast sprawdzać, czy z danej konfiguracji źródłowej (p, μ) można osiągnąć daną konfigurację docelową (q, ν), tj. czy $\mu \rightsquigarrow_{pq} \nu$, jesteśmy zainteresowani wyrażeniem całej relacji osiągalności $\rightsquigarrow_{pq}$ za pomocą formuły φ_{pq} zapisanej w rozstrzygalnej logice. Mówimy, że formuła φ_{pq} *wyraża* relację $\rightsquigarrow_{pq}$ wtedy, gdy dla dowolnych $\mu, \nu \in \mathbb{R}^X_{\geq 0}$,

$$\mu \rightsquigarrow_{pq} \nu \quad \text{wtw} \quad \mu, \nu \vDash \varphi_{pq}.$$

Osiągalność redukuje się do osiągalności binarnej, ponieważ mając daną formułę φ_{pq} wyrażającą relację $\rightsquigarrow_{pq}$, możemy rozstrzygać, czy $\mu \rightsquigarrow_{pq} \nu$ po prostu obliczając φ na μ, ν.

Naszym głównym wynikiem jest twierdzenie mówiące, że relacje osiągalności modelu TPDA można wyrazić w efektywnie rozstrzygalnej logice. Niech $\mathcal{A}_{\mathbb{Z}} = (\mathbb{Z}, \leq, (\equiv_m)_{m\in\mathbb{N}}, +, (k)_{k\in\mathbb{Z}})$ i $\mathcal{A}_{\mathbb{Q}} = (\mathbb{Q}, \leq, +, (k)_{k\in\mathbb{Z}})$ z naturalną interpretacją symboli. Językami pierwszego rzędu tych struktur są dobrze znane *arytmetyka Presburgera* i *arytmetyka wymierna*. Rozważamy bardziej ogólną dwusortową strukturę $\mathcal{A} = \mathcal{A}_{\mathbb{Z}} \uplus \mathcal{A}_{\mathbb{Q}}$, której język pierwszego rzędu nazywamy *arytmetyką liniową* (ang. *linear arithmetic*). Intuicyjnie, używamy pierwszego sortu do opisania części całkowitych wartości zegarów, a drugiego do części ułamkowych. Oto nasz główny wynik dotyczący algorytmicznej analizy modelu TPDA (por. [Twierdzenie 5, A], pierwsza część).

Theorem 13 (TPDA [A]) *Relacja osiągalności modelu* TPDA $\rightsquigarrow$ *może być wyrażona za pomocą rodziny formuł egzystencjalnych arytmetyki liniowej, obliczalnych w* 2ExpTime.

Bez stosu, tj. dla zwykłych automatów czasowych (ang. *timed automata*; TA) [**?**], i, bardziej ogólnie, dla TPDA z bezczasowym stosem, tj. dla PDTA [**?**], złożoność poprawia się wykładniczo (por. [Twierdzenie 5, A], druga część).

Theorem 14 (TPDA z bezczasowym sostem i TA [A]) *Relacja osiągalności* $\rightsquigarrow$ *dla modelu* TPDA *z bezczasowym stosem i dla* TA *może być wyrażona za pomocą rodziny formuł egzystencjalnych arytmetyki liniowej, obliczalnych w* ExpTime.

Nasze powyższe wyniki poprawiają podobne wyniki z literatury na temat modelu TA [**?**, **?**, **?**, **?**] i PDTA [**?**]. Przy okazji udowadniamy wynik interesujący sam w sobie: pewna podlogika arytmetyki liniowej, odpowiadająca ograniczeniom zegara, ma efektywną eliminację kwantyfikatorów [Wniosek 3, A]. Eliminacja kwantyfikatorów pozwala nam ograniczyć się do pewnej podklasy modelu TPDA, gdzie wszystkie ograniczenia są ułamkowe. Ponadto redukujemy ułamkowe TPDA do DPDA z atomami porządku cyklicznego (por. rozdz. **??**), do których stosujemy nasze poprzednie wyniki [D]. Jest to ciekawe, ponieważ pokazuje możliwość zastosowania automatów z rejestrami, takich jak DPDA, do analizy modelu z zegarami, takiego jak TPDA.

4.3.8 Conclusions

Badaliśmy problem osiągalności dla wyrażalnych klas systemów nieskończenie stanowych, łączących nieskończone dyskretne struktury danych z wyrażalnymi ograniczeniami czasowymi. W szczególności, w pracach [E] i [F] rozważaliśmy kanały komunikacyjne FIFO, a stosy w [A], [B], [C] i [D].

Ogólny obraz wyłaniający się z naszych badań można podsumować stwierdzeniem, że problem osiągalności pozostaje rozstrzygalny, aczkolwiek złożoność obliczeniowa czasami się pogarsza. Na przykład w przypadku automatów komunikający się (CA; pracy [E] i [F]) wiadomo było, że osiągalność jest rozstrzygalna, jeśli topologia sieci komunikacji jest typu polyforest. Z naszych badań wynika, że jeśli dodamy czas do tego modelu (TCA), to zbiór rozstrzygalnych topologii nie zmienia się (bez testów pustości; por. Twierdzenie **??**). Złożoność jest jednak ExpSpace-zupełna, o ile pozwolimy procesom czekać dowolną ilość czasu na koniec biegu, a nieelementarna w ogólnym przypadku (por. Twierdzenie **??**). Kontrastuje to z faktem, że w przypadku bezczasowym problem osiągalności jest PSpace-zupełny.

Podobnie w przypadku automatów czasowych ze stosem (TPDA) problem osiągalności jest nadal rozstrzygalny, jednak złożoność pogarsza się z PTime w przypadku bezczasowym do ExpTime-zupełnej w przypadku czasowym. Z pozytywnej strony możemy jeszcze powiedzieć, że ograniczenie górne ExpTime dotyczy DPDA nad bardzo dużą klasą „danych jakościowych" (atomy homogeniczne; por. Twierdzenie **??** [D]), trPDA i TPDA z bezczasowym stosem (por. Twierdzenie **??** [C] i, odp., **??** [A]) oraz trPDA nad czasem monotonicznym [**?**, Wniosek 6.8]. Dla jeszcze bardziej wyrażalnych klas trPDA i TPDA (ze stosem czasowym), pokazaliśmy, że złożoność jest 2ExpTime (por. Twierdzenie **??** [B] i, odp., **??** [A]). Kwestię, czy złożoność 2ExpTime jest w tych przypadkach optymalna, pozostawiamy do zbadania w przyszłości.

5 Inne publikacje

5.1 Prace o schematach rekurencyjnych wyższego rzędu

Następujące dwie prace dotyczą algorytmicznej analizy *schematów rekurencyjnych wyższego rzędu* (ang. *higher-order recursive schemes*, HORS), bardzo wyrażalnego formalizmu używanego do modelowania rekurencji wyższego rzędu.

- Lorenzo Clemente, Paweł Parys, Sylvain Salvati, Igor Walukiewicz. *Ordered Tree-Pushdown Systems.* In Proc. of FSTTCS'15, pages 163–177 [**?**].

  W tej pracy proponujemy alternatywne podejście do weryfikacji HORS oparte na automatach ze stosem drzewiastym, uogólniających zwyczajne automaty ze stosem liniarnym. Udowadniamy ogólny wynik o *zachowaniu regularności*: przeciwobraz relacji osiągalności regularnego zbioru drzew T to też regularny zbiór drzew. Wynik ten jest efektywny w tym sensie, że z podanego automatu rozpoznającego T można skonstruować automat dla jego przeciwobrazu.

- Lorenzo Clemente, Paweł Parys, Sylvain Salvati, Igor Walukiewicz. *The Diagonal Problem for Higher-Order Recursion Schemes is Decidable.* In Proc. of LICS'16, pages 96–105 [**?**].

  Głownym wynikiem tej pracy jest udowodnienie rozstrzygalności następującej nieregularnej własności dotyczącej HORS: Czy dla każdego ograniczenia $n \in \mathbb{N}$ dany HORS rozpoznaje drzewo skończone, w którym każda litera ze skończonego alfabetu występuje co najmniej n razy? Ponieważ własność ta nie jest regularna, jej rozstrzygalność nie wynika z [**?**]. Z naszego twierdzenia wynika szereg interesujących konsekwencji, takich jak obliczalność domknięcia w dół języków słów skończonych [**?**], rozstrzygalność problemu separowalności przez języki fragmentarycznie testowalne (ang. *piecewise testable languages*) [**?**] i rozstrzygalność problemu osiągalności pewnej klasy systemów współbieżnych [**?**].

5.2 Prace o problemach separowalności

Zbiór S separuje zbiory A, B wtedy, gdy $S \subseteq A$ i $S \cap B = \varnothing$. Dla ustalonych klas zbiorów $\mathcal{A}$ i $\mathcal{S}$ problem separowalności zdefiniowany jest następująco: Mając dane dwa zbiory $A, B \in \mathcal{A}$ rozstrzygnąć, czy istnieje zbiór $S \in \mathcal{S}$ separujący A, B. Problem separowalności jest jednym z głownych tematów w informatyce teoretycznej (por. teoria obliczeń i teoria języków formalnych) jak również w matematyce (np. w topologii). Następujące dwie prace zajmują się problemem separowalności.

- Lorenzo Clemente, Wojciech Czerwiński, Sławomir Lasota, and Charles Paperman. *Separability of Reachability Sets of Vector Addition Systems.* In Proc. of STACS'17, pages 24:1–24:14 [**?**].

  W tej pracy badamy problem separowalności dla zbiorów osiągalności VAS (ang. *vector addition systems*) i ich uogólnień. Głownym wynikiem jest rozstrzygalność problemu separowalności. To pierwszy krok w kierunku bardziej ambitnego celu: rozstrzygalności regularnej separowalności dla języków rozpoznanych przez VAS.

- Lorenzo Clemente, Wojciech Czerwiński, Sławomir Lasota, and Charles Paperman. *Regular Separability of Parikh Automata.* In Proc. of ICALP'17, pages 117:1–117:13 [**?**].

  W tej pracy badamy problem regularnej separowalności dla języków rozpoznanych przez automaty Parikha. Głownym wynikiem jest rozstrzygalność problemu regularnej separowalności. To trochę zaskakujący wynik, ponieważ problem regularności automatów Parikha jest nierozstrzygalny. Udowadniamy dodatkowo, że separowalność dla zbiorów semilinearnych przez zbiory unarne i modularne także jest rozstrzygalna.

5.3 Prace na temat gier stochastycznych

Następujące dwie prace dotyczą gier stochastycznych dla grafów skończonych i nieskończonych.

- Lorenzo Clemente and Jean-François Raskin. *Multidimensional beyond Worst-Case and Almost-Sure Problems for Mean-Payoff Objectives.* In Proc. of LICS'15, pages 257–268 [?] (por. [?]).

  Tak znany problem BWC (ang. *beyond worst-case*) uogólnia gry deterministyczne i stochastyczne [?]: dokładniej, w problemie tym szukamy strategii wygrającej z każdym deterministycznym przeciwnikiem; dodatkowo ta sama strategia musi optymalizować oczekiwaną wartość względem ustalonego modelu stochastycznego przeciwnika. Głównym wynikiem jest coNPTime-zupełność wielowymiarowego problemu BWC dla skończonych grafów; wynik ten dotyczy zarówno strategii skończenie pamięciowych, jak i dowolnych strategii; dodatkowo udowadniamy, że złożoność problemu spada do PTime jeśli interesuje nas prawie pewna wygrana (ang. *almost-sure winning*). Nasze wyniki uogólniają jednowymiarowy problem BWC [?].

- Parosh Abdulla, Lorenzo Clemente, Richard Mayr, and Sven Sandberg. *Stochastic Parity Games on Lossy Channel Systems.* Logical Methods in Computer Science, 2014, volume 10, number 4 [?] (specjalna wersja czasopismowa artykułu konferencyjnego [?]).

  W tej pracy badamy klasę gier stochastycznych na nieskończonych grafach konfiguracji LCS (ang. *lossy channel systems*). Rozważamy ω-regularne warunki wygranej oraz prawie pewną wygraną (ang. *almost-sure winning*). Udowadniamy, że gry te są rozstrzygalne pod warunkiem, że obaj gracze mają skończoną pamięć. Założenie skończonej pamięci jest konieczne, gdyż dla dowolnych strategii gry te są nierozstrzygalne już dla warunków wygranej typu *co-Büchi* [?].

5.4 Prace na temat inkluzji języków i upraszczania automatów

Następujące prace zajmują się problemami inkluzji języków i upraszczania automatów. Inkluzja języków jest istotnym problem w teorii automatów, z ważnymi zastosowaniami jak na przykład *weryfikacja modelowa* (ang. *model-checking*) [?] i tak znana zasada *size-change termination* [?]. Upraszczanie automatów jest istotym problemem w rozstrzyganiu teorii logicznych z użyciem algorytmów opartych na automatach [?], w analizie systemów nieskończenie stanowych takich jak sieci Petriego (ang. *Petri nets*) [?] i w weryfikacji modeli regularnych (ang *regular model checking*) [?]. Oba problemy są PSpace-zupełne, więc efektywne algorytmy są istotne w praktyce. Skupimy się na językach słów nieskończonych rozpoznawanych przez niedeterministyczne i alternujące automaty Büchiego.

- Lorenzo Clemente and Richard Mayr. *Multipebble simulations for alternating automata.* In Proc. of CONCUR'10, pages 297–312 [?].

  Dobrym podejściem do zmniejszenia ilości stanów automatu jest branie ilorazu. Podejście brania ilorazu opiera się na zidentyfikowaniu odpowiedniej relacji równoważności, którą można szybko obliczyć. Badamy tak zwane relacje symulacji z wieloma kamykami (ang *multipebble simulation relations*) dla automatów alternujących, uogólniąc wcześniejsze wyniki [?, ?, ?]. Udowiadniamy, że relacje symulacji z wieloma kamykami implikują zawieranie się języków, można ich więc używać do brania ilorazu; można je też obliczać w PTime dla stałej liczby kamyków (ang *pebbles*).

- Parosh Abdulla, Yu-Fang Chen, Lorenzo Clemente, Lukáš Holík, Chih-Duo Hong, Richard Mayr, and Tomáš Vojnar. *Simulation subsumption in Ramsey-based Büchi automata universality and inclusion testing.* In Proc. of CAV'10, pages 132–147 [?].

  Proponujemy efektywny algorytm rozwiązujący problemy uniwersalności i inkluzji dla niedeterministycznych automatów Büchiego. Ulepszamy poprzedne podejście do problemu uniwersalności [?] stosując relacje symulacyjne i uogólniemy nasze metody do problemu inkluzji.

Nasze eksperymenty wykazują znaczącą poprawę wydajności w stosunku do poprzedniego podejścia.

- Parosh Abdulla, Yu-Fang Chen, Lorenzo Clemente, Lukáš Holík, Chih-Duo Hong, Richard Mayr, and Tomáš Vojnar. *Advanced Ramsey-based Büchi automata inclusion testing.* In Proc. of CONCUR'11, pages 187–202 [**?**].

  Wykorzystujemy relacje symulacji w przód i w tył (ang. *forward and backward simulations*), aby jeszcze bardziej poprawić algorytmy rozwiązujące problemy uniwersalności i inkluzji automatów.

- Lorenzo Clemente. *Büchi automata can have smaller quotients.* In Proc. of ICALP'11, pages 258–270 [**?**].

  Badamy różne uogólnienia relacji symulacji dla automatów Büchiego umożliwiające branie ilorazu. Udowadniamy, że relacja zwana *fixed-word delayed simulation* jest maksymalna w tym kontekście.

- Lorenzo Clemente and Richard Mayr. *Efficient reduction of nondeterministic automata with application to language inclusion testing.* Logical Methods in Computer Science, volume 15, issue 1, 2019 [**?**] (wersja czasopismowa pracy [**?**]).

  Prezentujemy bardzo efektywne algorytmy zmniejszenia rozmiaru automatów Büchiego. Łączymy podejście brania ilorazu z nowatorskimi technikami usunięcia i dodawania przejść. Proponujemy tak znane *symulacje z wyprzedzeniem* (ang. *lookahead simulations*) jako efektywnie obliczalne relacje symulacji. Przeprowadziliśmy rozległe eksperymenty wykazując, że nasz algorytm zmniejsza rozmiar automatu znacznie lepiej niż wcześniejsze techniki.