Statement of scientific achievements
Autoreferat

March 2019
Marzec 2019

1 Name
Imię i nazwisko

Lorenzo Clemente

2 Degrees
Dyplomy i stopnie

- PhD in theoretical computer science, University of Edinburgh, 2012.
  Stopień doktora nauk informatycznych uzyskany w 2012 r. na Uniwersytecie Edinburgskim.
  Thesis: *Generalized Simulation Relations with Applications in Automata Theory.*
  Praca doktorska: *Uogólnione relacje symulacji z zastosowaniami w teorii automatów* (ang. *Generalized Simulation Relations with Applications in Automata Theory*).

- MSc in computer engineering, University of Rome "Tor Vergata", 2008.
  Tytuł magistra inżynierii uzyskany w 2008 r. na Universytecie Rzymskim "Tor Vergata".
  Thesis: *Algorithms and Tools for the Formal Verification of Concurrent Systems.*
  Praca magisterska: Algorytmy i narzędzia dla weryfikacji formalnej w systemach współbierznych (ang. *Algorithms and Tools for the Formal Verification of Concurrent Systems*).

3 Professional career
Zatrudnienie w jednostkach naukowych

- University of Edinburgh (UK), 2008–2011 (3 years), teaching assistant.
  Uniwersytet w Edinburgu (Wielka Brytania), 2008–2011 (3 lata), asystent dydaktyczny.

- Université de Bordeaux (France), 2011–2013 (2 years), post-doc.
  Université de Bordeaux (Francja), 2011–2013 (2 lata), staż podoktorski.

- Université Libre de Bruxelles (Belgium), 2013–2014 (7 months), post-doc.
  Université Libre de Bruxelles (Belgia), 2013–2014 (7 miesięcy), staż podoktorski.

- University of Warsaw (Poland), 2014–2016 (2 years), post-doc.
  Uniwersytet Warszawski (Polska), 2014–2016 (2 lata), staż podoktorski.

- University of Warsaw (Poland), 2016–present, associate professor.
  Uniwersytet Warszawski (Polska), od 2016, adiunkt (aktualne zatrudnienie).

4 Scientific achievements

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 Title

Tytuł

Adding time to infinite state systems

Wzbogacenie o czas systemów nieskończenie-stanowych

4.2 Publications included in the scientific achievements

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 Description of the results

Omówienie wyników

4.3.1 Introduction

Wstęp

Our every day life increasingly depends on the correct functioning of ICT systems (Information and Communication Technology), which come in a wide variety of forms, such as embedded systems, cryptocurrencies (e.g., Bit Coin), communication protocols, and computer software. Malfunctioning ICT systems can cause loss of money at best, and loss of human life in critical applications. For a long time the paradigmatic example of the importance of the correct functioning of ICT systems was that of the Ariane-5 rocket, whose explosion in 1996 due to a flaw in a numeric procedure caused the loss of the ~7 billion USD spent during a decade of research. Nowadays, cryptocurrencies provide an even more striking example, such as the infamous *DAO attack* on Ethereum [**?**], whereby an attacker exploited a vulnerability of the corresponding smart contract and managed to put the equivalent of ~60 million USD under her control.

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.

The area of *formal methods* addresses the problem of showing that ICT systems do not have bugs *formally* (i.e., *mathematically*). This is achieved with a variety of techniques, such as theorem proving and model checking. In the *model checking* approach, one starts by building a mathematical model of the system. Given a specification in some logical formalism, one verifies that the model of the system meets the specification, i.e., the absence of bugs, through the exhaustive exploration of all its reachable states [**?**]. Classically, the model checking approach has been applied to systems composed of finitely many states. This guarantees that the exploration procedure terminates after a finite amount of time, and thus model checking is a completely automated approach. This contrasts with theorem proving, which relies on the user to manually provide a formal proof of the correctness of the system, which then the theorem prover can verify. On the one hand, this means that theorem proving is not a completely automated procedure (at least in its classical incarnation). On the other hand, theorem proving can be applied to the wider class of systems with *infinitely many states*.

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.

There has been a trend starting in the 90's aiming at extending the scope of model checking from finite to infinite systems (but of course finitely represented) [**?**]. Infinite state systems are modelled as (infinite) *transition systems*, i.e., graphs, $G = (S, \to)$, where S is a set of *states* and $\to \subseteq S \times S$ is the *transition relation*, whereby we write $s \to t$ whenever the system can go from state $s \in S$ to state $t \in S$ in one discrete step. In this overview, we will concentrate on the following algorithmic question, which is arguably the most fundamental algorithmic problem in formal verification.

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.

THE REACHABILITY PROBLEM PROBLEM OSIĄGALNOŚCI (ang. *reachability problem*)
Input: A finite presentation of a (potentially infinite) graph $G = (S, \to)$ and two vertices $s, t \in S$.
Wyjście: Skończona reprezentacja (potencjalnie nieskończonego) grafu $G = (S, \to)$ i dwa wierzchołki $s, t \in S$.
Output: Is it the case that $s \to^* t$?
Wejście: Rozstrzygnij, czy $s \to^* t$?

(In the box above, "$\to^*$" is the *reachability relation*, i.e., the reflexive and transitive closure of the transition relation "$\to$".) The importance of the reachability problem stems from the fact that it can be used to prove that a system will never reach an erroneous configuration. Once we have identified a decision problem, such as reachability, the first question is whether this problem is

decidable, i.e., whether there exists an algorithm which, given an instance of the problem, always terminates, and answers YES precisely when $s \to^* t$ holds.
(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 $s \to^* t$.

Decidability: Is the reachability problem decidable?
Rozstrzygalność: Czy problem osiągalności jest rozstrzygalny?

Fundamental negative results in computability theory tell us that the reachability problem is undecidable in general, i.e., for Turing machines [**?**]. Nonetheless, there are non-trivial classes of infinite state systems with a decidable reachability problem, which are obtained by extending finite systems with discrete data structures, such as
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. a single pushdown stack, yielding *pushdown automata* [**?**, **?**] which are used to model programs with recursive procedures;
   jeden stos, jak w *automatach ze stosem* (ang. *pushdown automata*) [?, ?];
2. non-negative integer counters, yielding *Petri nets* (a.k.a. vector addition systems) [**?**, **?**] which are used to model concurrent systems comprised of unboundedly many processes;
   nieujemne całkowite liczniki, jak w *sieciach Petriego* (ang. *Petri nets*) [**?**, **?**], które używane są do modelowania systemów współbieżnych;
3. FIFO communication channels, yielding *communicating automata* [**?**, **?**], and their lossy variant [**?**, **?**] which are used to model communicating protocols.
   kolejki FIFO (ang. *first-in first-out*), jak w *automatach komunikujących się* (ang. *communicating automata*) [?, ?], które używane są w modelowaniu protokołów.

Establishing that reachability is decidable for a class of infinite state systems (as the ones above) is just the first step towards a more complete understanding thereof. The second natural question is determining the precise computational complexity of the reachability problem, i.e., whether the problem belongs to complexity classes such as LogSpace, PTime, NPTime, etc.
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.)

Complexity: What is the computational complexity of the reachability problem?
Złożoność: Jaka jest złożoność obliczeniowa problemu osiągalności?

It can be argued that knowing the exact *theoretical* computational complexity of a decision problem does not necessarily have an implication on the *empirical* complexity in solving instances of the problem that occur in practice (i.e., the SAT problem is NPTime-complete, but modern SAT solvers are blazingly fast on real-life SAT instances), and therefore this raises the question of why this is an important question to settle. We will give three answers. Determining the exact complexity of a decision problem 1) is a precise mathematical question, which can be given a precise mathematical answer—this is not the case with the empirical complexity; 2) can be taken as a synonym of having understood the problem, at least as a first approximation; 3) requires new mathematical insights interesting on their own, and those insights may have practical consequences.

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.

All the systems mentioned above are *discrete* in the sense that a single transition $s \to t$ cannot be further decomposed into smaller units. This level of abstraction is too coarse if one is interested in explicitly modelling more quantitative aspects of the computation, such as the amount of time necessary to execute a transition. This is important, since eliminating the timing information from a real-time system may result in the introduction of spurious bugs in the model, which cannot otherwise appear in the real system, thus leading to false positives when looking for bugs. The most successful model addressing this issue is that of *timed automata* (TA) [**?**], which extend finite transition systems with *clocks*, i.e., real-valued variables that can be manipulated in order to impose timing constraints on the behaviour of the system. Notwithstanding the increased expressive power of timed automata, their reachability problem is decidable, with optimal PSpace complexity.

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.

These two lines of research, that of extending finite systems with either unbounded discrete data structures, or with timing information, have happily been remarried in many works from the last two decades. Examples include *timed Petri nets* [**?**, **?**], *timed lossy channel systems* [**?**], *timed pushdown automata* [**?**], and *timed communicating automata* [**?**].

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*) [**?**].

Outline.

Plan. In this overview, we will concentrate on our new decidability and complexity results about the last two models, namely, timed pushdown automata and timed communicating automata. More precisely, our works [F] and [E] investigate decidability and complexity of the reachability problem for timed communicating automata in the setting of *arbitrary communication topologies*, thus extending [**?**] which addresses fixed topologies with one or two processes only. In the work [C] and [D] we investigate *timed-register pushdown automata* (trPDA), an alternative approach of introducing timing constraints for pushdown automata, namely, by using *timed-registers* (instead of clocks, as it is most commonly done). While trPDA have in general an undecidable reachability problem, under a technical assumption (called orbit-finiteness) we show that trPDA retain decidability of the reachability problem; moreover, we identify a subclass of orbit-finite trPDA simulating with optimal complexity timed pushdown automata from [**?**]. In the work [B] we show that we can in fact lift the technical assumption above while preserving decidability; we achieve this by uncovering an interesting connection with an expressive model called *branching vector addition systems* (BVAS). Finally, in the work [A] we extend *timed pushdown automata* (TPDA) from [**?**] with expressive integral and fractional clock constraints, and we show that, for this more general model, we can solve the reachability problem (and in fact the more general binary reachability problem; cf. Sec. **??**). In the rest of this section we explain the results above

in more detail.
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 Timed communicating automata
Czasowe automaty komunikacyjne [F]

Communicating automata (CA) are a fundamental model for studying concurrent processes exchanging messages over unbounded channels [?, ?]. However, the model is Turing-powerful, and even basic verification questions, like reachability, are undecidable. To obtain decidability, various restrictions have been considered, including allowing only one kind of messages to be sent [?], making channels unreliable [?, ?, ?], or restricting to half-duplex communication [?] (later generalized to mutex [?]). Decidability can also be obtained when restricting to executions satisfying additional restrictions, such as bounded context-switching [?], or bounded channels. Finally, and this is the restriction that we are interested in, decidability is obtained by constraining the communication topology. We say that a communication topology is a *polyforest* if it does not contain an undirected cycle. It is well-known that reachability is decidable (and in fact, PSpace-complete) if, and only if, the topology is a s.t. [?, ?].
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 [?, ?].

An extension of communicating automata with timing constraints (in the form of clocks) was studied in [?], where the authors show that reachability is decidable for the two process topology $p \to q$ and undecidable for the three process topology $p \to q \to r$. The undecidability result above crucially relies on the so-called *urgent semantics* for receptions, whereby if a message can be received by a process, then all other internal actions are disabled. This gives a further synchronisation facility, which, coupled with the implicit synchronisation through the elapse of time, yields the undecidabiliy result above.
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.

We study *timed communicating automata* (TCA) generalising [**?**] over arbitrary communication topologies. We show that the urgent semantics is equivalent to channel *emptiness tests*, allowing us to express our results over general communication topologies in a uniform way. Our first main result provides a complete characterisation of decidable communication topologies in the setting of discrete time; in the statement below, a *polytree* is a connected component in a polyforest (cf. [Theorem 3, F]).

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 decidability rozstrzygalność [F]) *The reachability problem for discrete time* TCA *is decidable if, and only if, the communication topology is a polyforest and in every polytree there exists at most one channel which can be tested for emptiness.*

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.*

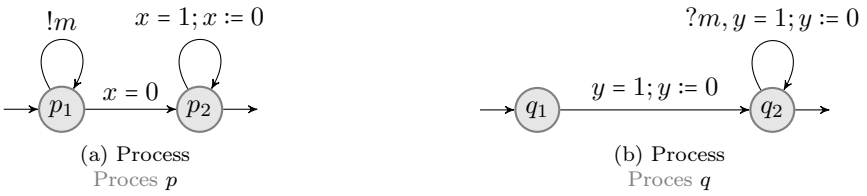
Over the more general dense time, we show decidability in the test-free case (cf. [Theorem 5, F][1]). This extends the previous work [**?**] regarding both decidable and undecidable topologies. First, our characterisation above vastly extends the class of decidable topologies: Not only $p \to q$ is decidable, but also, for instance, $p \to q \to r$ (and $p \to q \leftarrow r$) provided at most one of the two channels has emptiness tests/urgent semantics. Second, our characterisation also extends the class of undecidable topologies: For instance, the four process topology $p \to q \to r \to s$ is undecidable as soon as any two channels have emptiness tests/urgent semantics.

W bardziej ogólnym czasie gęstym, pokazujemy rozstrzygalność w przypadku bez testu na pustość (por. [Twierdzenie 5, F])[2]. 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).

In order to give an intuition of which technical difficulties have to be overcome in order to obtain decidability, consider the simple (decidable) communication topology $p \to q$. Without time, decidability of the reachability problem follows from the simple observation that we can restrict our attention to schedulings immediately matching every transmission $!m$ of p with a corresponding reception $?m$ of q, which keeps the channel length bounded (by 1); this yields a finite state system equivalent w.r.t. reachability to the original one. In the presence of time, this technique breaks down, and this happens already in discrete time. For an illustration, consider the two processes p and q in Fig. **??**. The two clocks x, y are initially 0. Process p starts in control location p_1

[1] In yet unpublished work we actually proved that Theorem **??** holds even in dense time [**?**].

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



(a) Process
Proces p

(b) Process
Proces q

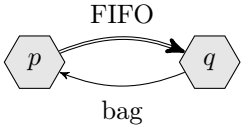
Figure 1: Channels cannot be bounded in timed communicating automata.
Kanały czasowych automatów komunikacyjnych nie mogą być ograniczone.

and from there can send an unbounded number of messages m to q. At some point it can move to control location p_2, *without elapsing time* as required by the guard $x = 0$. Process q starts in control location q_1, elapses one time unit and goes to q_2 (as required by the guard $y = 1$), and the clock y is reset (as requried by $y := 0$). From the latter location, process q can read one message $?m$ per time unit. The crucial point is that q needs to wait one time unit before any message can be received, while p can send an unbounded number of messages within the first time unit. Thus, it would not be complete to restrict the executions by bounding the size of the channel, because there are runs which cannot be rescheduled as to keep the channel bounded. This is the essential difficulty in analysis systems comprising channels and timing requirements.
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.

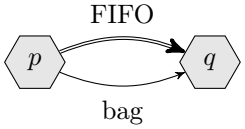
If we forbid emptiness tests (which do not increase the expressive power in the untimed setting), our characterisation from Theorem **??** is the same as in the untimed setting. However, in the timed setting the complexity provably worsens, and this happens already over discrete time. This is our second major result (cf. [Corollary 1, F]).
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 complexity złożoność [F]) *Already over discrete time, the reachability problem for* TCA *is not elementary.*
Złożoność obliczeniowa problemu osiągalności TCA *jest nieelementarna, już w czasie dyskretnym.*

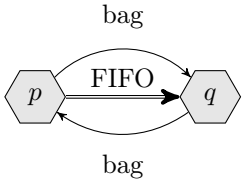
The lower bound above is shown by using connected topologies to simulate Petri nets/counter machines, and using the recent breakthrough on the complexity of the latter [**?**]. While the complexity above may seem discouraging, if we forbid emptiness tests and ignore the value of clocks at the end of the run, then the complexity becomes ExpSpace (by reduction to the coverability



(a) Opposite channels (decidable)
Przeciwległe kanały (rozstrzygalny)



(b) Parallel channels (decidable)
Równoległe kanały (rozstrzygalny)



(c) Both ways (undecidable)
Oba rodzaje kanałów (nierozstrzygalny)

problem for Petri nets/counter machines, which is in ExpSpace [**?**]).
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 Communicating automata with FIFO and bag channels
Automaty komunikacyjne z kanałami FIFO i bag [E]

The work [E] studies a generalisation of TCA over discrete time. Our proof of Theorem **??** [F] shows that we can simulate discrete-time TCA with untimed CA by introducing extra communication channels: More precisely, each TCA FIFO channel $p \to q$ can be simulated by replacing it with two (untimed) CA channels, one FIFO $p \to q$ and the other bag $q \to p$, going in the opposite direction. A *bag channel* is one where the messages can be freely reordered, and thus it is weaker than a FIFO channel in general. We show with a general argument that the resulting topology, shown in Fig. **??**, has a decidable reachability problem. On the other hand, if we allow the channel $q \to p$ to be FIFO, then we obtain an undecidable topology, since it forms an undirected FIFO cicle (i.e., it is not a polyforest [**?**, **?**]).
W pracy [E] badamy pewne uogólnienie TCA na dyskretny czas. Naszy dowód Twierdzenia **??** [F] 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 [**?**, **?**]).

This naturally raises the question of which communication topologies mixing FIFO and bag channels have a decidable reachability problem[3]. The motivation of studying such mixed topologies is not restricted to the simulation of discrete-time TCA. Bag channels can be used to model asynchronous procedure calls in models where such calls can be freely reordered [**?**, **?**, **?**]. Additionally, bag channels offer a non-trivial over-approximation of FIFO channels, turning undecidable FIFO topologies into decidable ones. For example, we prove that Fig. **??** and **??** are decidable, which would not be the case were both channels FIFO. We also have non-trivial examples of undecidable FIFO/bag topologies: The previous two decidable examples are maximal, since we show that combining them yields an undecidable topology; cf. Fig. **??**.
To oczywiście rodzi następne pytanie: Które topologie komunikacyjne z kanałami FIFO i typu bag są rozstrzygalne[4]? Motywacja studiowania takich mieszanych topologii nie ogranicza się do symulacji dyskretnego czasu. Kanały typu bag mogą modelować asynchroniczne wywołania procedur,

[3] The similar problem of mixing perfect and *lossy* FIFO channels has been studied in [**?**].
[4] Podobny problem topologii połączenia kanałów FIFO i stratnych został zbadany w [**?**].

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. ??.

Our main result is a complete characterisation of which topologies $\mathcal{T}$ of FIFO and bag channels have a decidable reachability problem. In the statement below, P is a decidable property of topologies involving certain cycles of FIFO and bag channels (cf. [Section 5, E] for a formal definition and [Theorem 5.3, E] for the characterisation); for instance P holds for the first two topologies above, but not for the third one.
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 (Characterisation of decidable topologies Charakteryzacja rozstrzygalnych topologii [E]) *The reachabiliy problem for* CA *over a topology $\mathcal{T}$ of FIFO and bag channels is decidable if, and only if, $P(\mathcal{T})$ holds.*
Problem osiągalności CA *na topologii $\mathcal{T}$ kanałów FIFO i bag jest rozstrzygalny, wtedy i tylko wtedy, gdy $P(\mathcal{T})$.*

Notice that a topology of only bag channels is effectively equivalent to a Petri net, and thus reachability is decidable. Thus, is always possible to over-approximate FIFO CA by turning *all* channels into bags. Thanks to our characterisation, a much finer analysis is obtained by selectively over-approximating only *some* of the FIFO channels (while preserving decidability).
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 Timed-register pushdown automata [B], [C], [D]
Czasowe automaty rejestrowe ze stosem (ang. *timed-register pushdown automata*) [B], [C], [D]

In the works [B], [C], and [D] we study the problem of introducing timing constraints into pushdown automata. An early attempt in the literature is *pushdown timed automata* (PDTA) [?], which extend classical pushdown automata by replacing the finite control with a timed automaton. Thus, the stack in a PDTA is just a classical untimed stack, which severely limits the expressive power of the model. For instance, PDTA cannot recognise natural timed context-free languages such as even-length *timed palindromes* (below w^R is the reversal of w):
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):

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

For a while, a candidate remedy to this situation seemed to be *dense-timed pushdown automata* (DTPDA) [?], which extend PDTA with *stack clocks* that can be pushed on and popped according to non-diagonal interval constraints. One can say that DTPDA are "infinite in two dimensions", since the stack is unbounded, and each stack symbol carries a clock, which is also unbounded.

Our work started from the observation that DTPDA semantically collapse to PDTA, in the sense that the two models are effectively equivalent w.r.t. the class of timed languages they recognise (cf. [Theorem II.1, C]).
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]) *For every* DTPDA *one can effectively construct a* PDTA *recognising the same timed language.*
Dla każdego DTPDA *można efektywnie skonstruować* PDTA *rozpoznający ten sam język czasowy.*

The somehow surprising result above is the consequence of the unexpected interplay between the monotonicity of time and the well-nested stack discipline. In fact, we show that the same holds even when allowing more liberal *diagonal* clock constraints between stack and control clocks. The question arises as to whether there is any sensible timed extension of pushdown automata not suffering from a semantic collapse to its "timeless stack" version (as it is the case with DTPDA). This is the starting point of our enquiry.
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 Set with atoms I: Oligomorphic and homogeneous atoms [D]
Zbiory z atomamy, część I: Oligomorficzne i homogeniczne atomy [D]

One answer to the problem above is to be found in the *sets with atoms* approach, a uniform way of extending automata theory—and in fact, mathematics—obtained by relaxing the notion of finiteness to orbit-finiteness [**?**] (cf. also the recent book [**?**]). The setting is parameterised with a relational structure $\mathbb{A}$, called *atoms*. The simplest such example is that of *equality atoms* $(\mathbb{N}, =)$, where the domain is any countably infinite set and the only relational symbol in the signature is equality. A richer structure is *(dense) total order atoms* $(\mathbb{Q}, \leq)$, which can be used to model a qualitative notion of dense time. We shall use total order atoms as an illustrative example for the rest of the section.
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.

The key intuition is that two sets with atoms are indistinguishable if there exists an atom *automorphism*[5] mapping one into the other; in this case, we say that they are in the same *orbit*. For instance, automorphisms of total order atoms are precisely the monotone isomorphisms of $\mathbb{Q}$; the two tuples $\bar{a} = (2, 1.1, 2)$ and $\bar{b} = (3, 2.9, 3)$ are indistinguishable since there exists an automorphism $\alpha : \mathbb{Q} \to \mathbb{Q}$ extending $[2 \mapsto 3, 1.1 \mapsto 2.9]$ s.t. $\alpha(\bar{a}) = \bar{b}$; more generally, the orbit of $\bar{a}$ is

[5] An automorphism of a relational structure $\mathbb{A}$ is an isomorphism of the domain of $\mathbb{A}$ which preserves and reflects the interpretation of all relational symbols of $\mathbb{A}$.

precisely
Kluczową intuicją jest to, że dwa zbiory z atomami są nierozróżnialne jeśli istnieje *automorfizm* struktury atomów[6] 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 $\{(a, b, c) \in \mathbb{Q}^3 \mid a = c, b < a\}$.

The crucial notion in the theory of sets with atoms is that of orbit-finiteness, which is a generalisation of finiteness: A set with atoms is *orbit-finite* if it can be partitioned into finitely many orbits. In other words, an orbit-finite set is finite up to automorphism. For example, $\mathbb{Q}$ and $\{(a, b) \in \mathbb{Q}^2 \mid a \neq b\}$ are both orbit-finite: The former has exactly one orbit (itself), and the second has two orbits,
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, $\{(a, b) \in \mathbb{Q}^2 \mid a < b\}$ and i $\{(a, b) \in \mathbb{Q}^2 \mid a > b\}$.

In our context, we are interested in generalising pushdown automata to the setting of sets with atoms. This is achieved by lifting all components in the usual textbook definition of pushdown automata (such as input and stack alphabets, control locations, and transition relation) from finite sets to sets with atoms definable in first-order logic. We call the resulting model *(first-order) definable pushdown automata* (DPDA). Each choice of the atoms $\mathbb{A}$ yields a different notion of pushdown automaton. For instance, if we take $\mathbb{A}$ to be equality atoms $(\mathbb{N}, =)$, then we obtain a model of register pushdown automata studied in [**?**]. More choices of atoms will be described below. We have studied the reachability problem for definable pushdown automata over well-behaved classes of atoms. We say that a relational structure $\mathbb{A}$ is *oligomorphic* if $\mathbb{A}^k$ is orbit-finite for every k [**?**], and that it is *(first-order) decidable* if the satisfiability problem for first-order logic over $\mathbb{A}$ is decidable. For instance, equality and total order atoms are oligomorphic and decidable; we will give later an example of non-oligomorphic structure. Our first result is that DPDA over decidable oligomorphic atoms have a decidable reachability problem (cf. [Theorem 4, D]).
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 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 (Oligomorphic Oligomorficzne DPDA [D]) *Let $\mathbb{A}$ be a decidable oligomorphic relational structure. The reachability problem is decidable for definable pushdown automata over $\mathbb{A}$. Niech $\mathbb{A}$ będzie rozstrzygalną oligomorficzną strukturą relacyjną. Problem osiągalności jest rozstrzygalny dla definiowalnych automatów ze stosem z atomami $\mathbb{A}$.*

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

(In fact we have shown a much more general result, namely that, under the assumptions of the theorem, DPDA enjoy a generic saturation procedure based on orbit-finite automata [**?**].) The result above is very general, since it only requires that the first-order satisfiability problem be decidable for the atoms, without any complexity assumption. This generality comes at the price that no complexity upper bound on the DPDA reachability problem can be given without further assumptions.
(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.

We say that a relational structure $\mathbb{A}$ is *homogeneous* if every partial isomoprhism extends to an automorphism of the whole structure [**?**]. Homogenuousness has the pleasant consequence that the number of orbits of $\mathbb{A}^k$ is exponentially bounded[7] In particular, homogeneous structures are oligimorphic. (There are oligimorphic relational structures which are not homogeneous, such as bit vector atoms [**?**].) Our second result is that the DPDA reachability problem becomes *fixed-parameter tractable* over homogeneous structures, in the sense that its running time is of the form $O(f(k) \cdot \mathsf{poly}(n))$, where n is the size of the DPDA and k the dimension (cf. [Theorem 7, Corollary 8, D]).
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[8]. 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 (Homogeneous Homogeniczne DPDA [D]) *Let $\mathbb{A}$ be a homogeneous relational structure. The reachability problem for* DPDA *over $\mathbb{A}$ is fixed-parameter tractable, with the dimension being the parameter. 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.*

Our third result is that, for commonly occurring structures, we can do even better. Let $\mathbb{A}$ and $\mathbb{B}$ be two relational structures. We say that $\mathbb{A}$ is a *substructure* of $\mathbb{B}$ if they have the same signature and the domain of $\mathbb{A}$ is included in that of $\mathbb{B}$; if additionally the intepretation of all relation symbols agree on the domain of $\mathbb{B}$, then we say that $\mathbb{A}$ is an *induced substructure* of $\mathbb{B}$. The *induced substructure problem* amounts to determine, given two relational structures $\mathbb{A}$ and $\mathbb{B}$, whether $\mathbb{A}$ is an induced substructure of $\mathbb{B}$. Our observation is that the induced substructure problem is efficiently solvable, i.e., in PTime, for the following commonly occurring homogenous relational structures ([Section 6, D]; cf. [**?**]):
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*

[7] Two tuples $\bar{a}, \bar{b} \in \mathbb{A}^k$ are isomorphic precisely when they satisfy the same relations of $\mathbb{A}$, and thus there are $O(2^{\mathsf{poly}(k)})$ equivalence classes of tuples. Since every partial isomorphism extends to an automorphism of $\mathbb{A}$, the same quantity above bounds the number of orbits of $\mathbb{A}^k$.

[8] 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 $\mathbb{A}^k$.

(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. [?]):

- Equality atoms $(\mathbb{N}, =)$: They can be used to model data values from an infinite set which can only be compared with equality.
  Atomy równościowe $(\mathbb{N}, =)$: Można ich użyć do modelowania wartości danych z nieskończonego zbioru, które można porównać tylko na równość.

- Total order atoms $(\mathbb{Q}, \leq)$: As mentioned above, they can be used to model the total order between real-time events.
  Atomy liniowego porządku $(\mathbb{Q}, \leq)$: Jak wspomniano powyżej, mogą być użyte do modelowania zdarzeń czasowych.

- Betweenness order atoms $(\mathbb{Q}, B)$, where $(x, y, z) \in B$ iff either $y < x < z$ or $z < x < y$: They can be used to model real-time events where one can only observe whether an event x occurs between two other events y and z, but has no further information on the ordering of y and z themselves.
  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.

- Cyclic order atoms $(\mathbb{Q}, K)$, where $(x, y, z) \in K$ iff $x < y < z$, or $y < z < x$, or $z < x < y$: They can be used to model the ordering of the fractional values of timestamps of events. This is a natural choice, since K is invariant under time elapse: $(x, y, z) \in K$ iff $(x \oplus \delta, y \oplus \delta, z \oplus \delta)$ for every $x, y, z \in [0, 1)$ and $\delta \in \mathbb{R}$, where $a \oplus b$ is the fractional part of $a + b$.
  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 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 $a + b$.

- Partial order atoms: They can be used to describe real-time events which can occurr independently of each other, but still need to satisfy certain causality constraints.
  Atomy porządku częściowego: Mogą być używane do opisywania zdarzeń, które mogą wystąpić niezależnie od siebie, ale nadal muszą spełniać pewne ograniczenia przyczynowości.

- Tree order atoms: They can be used to model the dynamic creation of processes which do not interact further in the future.
  Atomy porządku drzewiastego: Mogą być używane do modelowania dynamicznego tworzenia procesów, które nie będą dalej oddziaływać w przyszłości.

- Other examples include: Equivalence, preorder, graph, and tournament atoms.
  Inne przykłady to: atomy równoważności, preporządku, grafu i turnieju.

The following result applies to each of the previous examples (cf. [Corollary 10, D]).
Poniższy wynik dotyczy każdego z tych przykładów (por. [Wniosek 10, D]).

Theorem 7 (Tractable homogeneous Efektywne homogeniczne DPDA [D]) *Let $\mathbb{A}$ be a homogeneous relational structure with a* PTime *induced substructure problem. The reachability problem for definable pushdown automata over $\mathbb{A}$ is in* ExpTime*. Moreover, it is* ExpTime*-hard already for equality atoms $(\mathbb{N}, =)$.*
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* $(\mathbb{N}, =)$.

The list of atoms above shows the wide applicability of Theorem **??**[9]. The major limitation of all the structures previously mentioned is that they are essentially *qualitative*, and thus not apt at modelling more quantitative aspects of real-time events, such as their exact length. The next section is devoted to remedy this situation.
Powyższa lista struktur atomów pokazuje szerokie zastosowanie Twierdzenia **??**[10]. 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 Set with atoms II: Timed atoms [C]
Zbiory z atomami, część II: Atomy czasowe [C]

In order to model more quantitative features of real-time systems, we consider richer structures, such as (in increasing order of expressive power)
Aby modelować bardziej ilościowe cechy systemów czasu rzeczywistego (ang. *real-time systems*), rozważamy bogatsze struktury, takie jak (w rosnącej kolejności mocy wyrażania)

- *discrete time* atoms $(\mathbb{Z}, +1, \leq)$,
- *dense time* atoms $(\mathbb{Q}, +1, \leq)$, and
- *hybrid time* atoms $\mathbb{H} = (\mathbb{Z}, +1, \leq) \uplus (\mathbb{Q}, \leq)$, a two-sorted structure with a discrete component for integral values of timestamps and a dense component for fractional values of timestamps.

- 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).

We collectively call the structures above *timed atoms*, and we call *unconstrained timed-register pushdown automata* (unconstrained trPDA) definable pushdown automata over timed atoms. None of the timed atoms is oligomorphic (and thus not homogeneous)[11], and our generic technique from Sec. **??** cannot be applied to trPDA. Even worse, the reachability problem for definable finite automata over discrete or dense time atoms is in fact undecidable, and this holds already for discrete time atoms in dimension 3: The set of control locations is a subset of $\mathbb{Z}^3$ and the three components (x, y, z) simulate a two counter Minsky machine (which has an undecidable reachability problem [**?**]): Incrementing/decrementing counters are simulated with definable transitions $(x, y, z) \mapsto (x \pm 1, y, z)$ for the first counter and $(x, y, z) \mapsto (x, y \pm 1, z)$ for the second counter, and zero tests by $x = z$ and $y = z$, resp.
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

[9] The complexity upper bound on the induced substructure problem is a necessary assumption, since we can construct homogeneous structures whose induced substructure problem can simulate the membership problem in arbitrary subsets of $\mathbb{N}$ [Theorem 9, D].

[10] 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].

[11] For instance, already $\mathbb{Z}^2$ contains infinitely many orbits: For every integer $\mathbb{Z}^2$, the set $\{(x, x + k) \mid x \in \mathbb{Z}\}$ is a distinct orbit of $\mathbb{Z}^2$. This is due to the fact that automorphisms of $\mathbb{Z}$ need to preserve the distance between points, and thus there are "not enough" automorphism.

oligomorficzna (ani homogeniczna)[12], 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 $y = z$.

The source of undecidability is that one can remember timestamps which can be arbitrarily far from each other (such as x and z above). For definable automata, this issue is immediately solved by requiring that the set of control locations be orbit-finite. The resulting model of *orbit-finite automata* over timed atoms has a decidable reachability problem [**?**]. However, this is not sufficient for unconstrained trPDA, since one can still encode two Minsky counters even if control locations are orbit-finite, already in dimension one: The first counter is encoded as the height of the stack (as classically done), and the second counter is encoded as the difference between the control value and the value on top of the stack (cf. [Theorem IV.1, C]).

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 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]) *The reachability problem for unconstrained* trPDA *is undecidable.*
Problem osiągalności dla nieograniczonych trPDA *jest nierozstrzygalny.*

The simulation above is made possible by the fact that push operations are allowed to see the value on top of the stack; (cf. classical pushdown automata, where such a feature does not increase the expressiveness of the model). As a solution, we forbid push operations from reading the top of the stack altogether, and obtain what we call below just trPDA. The expressiveness of trPDA is witnessed by the fact that they recognise timed palindromes (**??**), which can be done as follows. The automaton works in two phases. In the first phase, it reads an input symbol of the form $(a, x) \in \Sigma \times \mathbb{R}$ and pushes it on the stack. Then, the automaton nondeterministically guesses the middle of the word, and switches to the second phase where it reads from the input (a, x) and simultaneously pops (a, y) from the stack provided $x = y$. A trPDA can even go beyond context-free languages: Consider the language of untimed palindromes over the alphabet $\Sigma = \{a, b\}$ containing the same number of a's and b's:

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:

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

[12]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.

Language M can be recognised by a trPDA as follows: Initially, the automaton pushes $(\bot, x)$ on the stack and remembers x in the state, where $x \in \mathbb{Z}$ is a guessed integer to be used at the end of the run. The rest is similar to timed palindromes; additionally, the local control value x is increased/decreased by 1 upon reading a/b. At the end of the run, in order to check that the word contained the same number of a's and b's, the automaton pops $(\bot, x)$ only if x is the same as the local control value.
Język M jest rozpoznawany przez trPDA w następujący sposób: Początkowo, automat wkłada $(\bot, x)$ 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 $(\bot, x)$ ze stosu tylko wtedy, gdy x jest taki sam jak lokalny rejestr.

Notwithstanding the fact that trPDA languages go beyond the class of context-free languages, our main result is that they are decidable (cf. [Theorem 1, B]).
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]) *The reachability problem for* trPDA *is in* 2ExpTime.
Problem osiągalności dla trPDA *jest rozstrzygalny, ze złożonością* 2ExpTime.

The result above is obtained by establishing a tight connection between trPDA and *branching vector addition systems* in dimension one with addition and subtraction ($\mathbb{Z}$-BVASS) [?]. The latter is an expressive generalisation of Petri nets/counter automata/vector addition systems, which in its basic variant with addition is decidable in PTime in dimension one [?]; in higher dimension, this is a long-standing open problem in the infinite-state systems community [?]. Our main technical result leading to Theorem ?? is that $\mathbb{Z}$-BVASS are decidable, and in fact, elementary (cf. [Theorem 6, B]).
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]) *The reachability problem for* $\mathbb{Z}$-BVASS *in dimension one is in* ExpTime.
Problem osiągalności modelu $\mathbb{Z}$-BVASS *dla wymiaru* 1 *jest rozstrzygalny, ze złożonością* ExpTime.

We show that the complexity improves for the following progressively less expressive subclasses of trPDA. In an *orbit-finite* trPDA we require that the combined set of control locations and top-most stack symbols be orbit-finite (hence the name). Under this restriction, we are able to show an improved upper complexity bound (cf. [Theorem IV.8, C]).
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 (orbit-finite orbitowo-skończone trPDA [C]) *The reachability problem for orbit-finite* trPDA *is in* NExpTime *and* ExpTime*-hard.*
Problem osiągalności modelu orbitowo skończonego jest rozstrzygalny w NExpTime *i* ExpTime*-trudny.*

Moreover, if we forbid timing information from the stack altogether, we obtain an even better complexity (cf. [Theorem IV.5, C]). This is particularly interesting, since TRPDA with timeless stack suffice to simulate DTPDA and PDTA [Corollary IV.10, C], and in this case the complexity is optimal since the latter models are already EXPTIME-hard [?].
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 with timeless-stack z bezczasowym stosem [C]) *The reachability problem for* TRPDA *with timeless stack is* EXPTIME*-complete.*
Problem osiągalności modelu TRPDA *z bezczasowym stosem jest* EXPTIME*-zupełny.*

Finally, and this may be the most interesting TRPDA subclass, TRPDA over *monotonic time* are also EXPTIME-complete [?, Corollary 6.8].
Wreszcie, i może to być najciekawsza podklasa modelu TRPDA, model TRPDA z *czasem monotonicznym* jest również EXPTIME-zupełny [?, Wniosek 6.8].

TRPDA constitute an expressive and yet decidable class of timed infinite-state systems combining recursion with real-time constraints. This was achieved by deviating from the standard clock-based approach, and studying instead a register-based model in the general framework of sets with atoms. The question remains whether we can find an expressive model of timed pushdown automata within the realm of real-time clocks. This is what we tackle in the next section.
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 Timed pushdown automata
Automaty czasowe ze stosem [A]

The semantic collapse of Theorem **??** shows that traditional clock constraints are not sufficient to exploit the power of a timed stack. For example, consider the language of timed odd length palindromes over $\Sigma = \{a, b\}$ s.t. matching symbols are at integral distance:
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:

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

Clearly, we need a timed stack to recognise N, since we need to remember the fractional part of the x_i's that we are pushing on the stack in the first part of the run in order to check that it equals the fractional part of the x_{2n-i}'s in the second part of the run. In Sec. **??**, we addressed this issue by introducing timed-register pushdown automata (TRPDA), a register-based model obtained by instantiating definable pushdown automata to timed atoms. However, the question arises as to whether a classical clock based model can recognise truly timed context-free languages such as N.
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.

We show in [A] that this is indeed the case. We introduce *timed pushdown automata* (TPDA), a model extending classical pushdown automata with both control and stack clocks, which can be pushed to and popped from the stack. Since the the stack is unbounded, TPDA are "infinite in two dimensions". As time elapses, all clocks increase their values at the same rate. Clocks can be compared according to boolean combinations of classical and fractional *clock constraints*
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[13]:

	(classical)	(fractional)
(non-diagonal)	$x \leq k, \quad k \in \mathbb{Z}$	$\{x\} = 0,$
(diagonal)	$x - y \leq k, \quad k \in \mathbb{Z}$	$\{x\} \leq \{y\}.$

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

Control clocks can be reset and compared according to the constraints above. At the time of a push operation, new stack clocks are created whose values are initialised, possibly non-deterministically, as to satisfy a given push constraint between stack clocks and control clocks; similarly, a pop operation requires that stack clocks to be popped satisfy a given pop constraint of analogous form.
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.

The timed stack of a TPDA cannot be untimed (unlike for DTPDA; cf. [**?**] for a similar observation) since we can construct a TPDA recognising N above: There is one control clock x, which is initially 0, and is never reset. In the first phase, the automaton reads input symbol $c \in \Sigma$, and pushes it on the timed stack together with a fresh stack clock y satisfying the diagonal constraint $x = y$. In the second phase, the automaton reads $c \in \Sigma$ and pops (c, y) from the stack, where y is the stack clock associated with stack symbol c, provided it has the same fractional value as the control clock x, i.e., $\{x\} = \{y\}$.
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. $\{x\} = \{y\}$.

In the context of TPDA, it is natural to consider the reachability relation ograniczona restricted to initial and final configurations with empty stack. Let X be the finite set of control clocks, let L be the finite set of control locations, and let $\mathbb{R}^X_{\geq 0}$ be the set of clock valuations. In the rest of this section, the *reachability relation* is the family of relations $\{\rightsquigarrow_{pq}\}_{p,q \in L}$ defined as follows: For two clock valuations $\mu, \nu \in \mathbb{R}^X_{\geq 0}$ and control locations $p, q \in L$,

[13] We denote with $\{x\}$ the fractional value of clock x Oznaczamy jako $\{x\}$ część ułamkową wartości zegara x.

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 kontrolnych, i niech $\mathbb{R}_{\geq 0}^X$ 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 $\mu, \nu \in \mathbb{R}_{\geq 0}^X$ i $p, q \in L$,

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

precisely when there is a run starting from control state p, control clocks valuation μ, and empty stack, ending in control state q, control clocks valuation ν, and empty stack. For fixed control locations p, q, the reachability relation $\rightsquigarrow_{pq}$ is a subset of $\mathbb{R}_{\geq 0}^X \times \mathbb{R}_{\geq 0}^X$ and can thus be described with a logical formula over the real numbers. We study the *binary reachability problem* for TPDA, which is more general than mere reachability: Instead of checking whether a *given* source configuration (p, μ) can reach a given target configuration (q, ν), i.e., $\mu \rightsquigarrow_{pq} \nu$, we are interested in expressing the reachability relation $\rightsquigarrow_{pq}$ itself with a formula φ_{pq} in a decidable logical formalism. We say that φ_{pq} *expresses* $\rightsquigarrow_{pq}$ if, for every clock valuations $\mu, \nu \in \mathbb{R}_{\geq 0}^X$,
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 $\mathbb{R}_{\geq 0}^X \times \mathbb{R}_{\geq 0}^X$ 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 $\mu, \nu \in \mathbb{R}_{\geq 0}^X$,

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

Reachability reduces to binary reachability, since given a formula φ_{pq} expressing $\rightsquigarrow_{pq}$, we can decide whether $\mu \rightsquigarrow_{pq} \nu$ holds by simply evaluating φ on μ, ν.
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.

Our main result is to show that TPDA reachability relations are expressible in an efficiently decidable logic. Let $\mathcal{A}_{\mathbb{Z}} = (\mathbb{Z}, \leq, (\equiv_m)_{m\in\mathbb{N}}, +, (k)_{k\in\mathbb{Z}})$ and $\mathcal{A}_{\mathbb{Q}} = (\mathbb{Q}, \leq, +, (k)_{k\in\mathbb{Z}})$, with the expected interpretation of symbols in the signature. The first-order languages of these structures are the well-known *Presburger arithmetic* and, resp., *rational arithmetic*. We consider a more general two-sorted structure $\mathcal{A} = \mathcal{A}_{\mathbb{Z}} \uplus \mathcal{A}_{\mathbb{Q}}$, whose first-order language we call *linear arithmetic*. Intuitively, we use the first sort to describe the integral values of clocks, and the second sort to describe the fractional values of clocks. The following is our main result in the algorithmic analysis of TPDA (cf. [Theorem 5, A], first part).
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]) *The* TPDA *reachability relation* $\rightsquigarrow$ *can be expressed by a family of existential linear arithmetic formulas computable in* 2ExpTime.
Relacja osiągalności modelu TPDA $\rightsquigarrow$ *może być wyrażona za pomocą rodziny formuł egzystencjalnych arytmetyki liniowej, obliczalnych w* 2ExpTime.

Without a stack, i.e., for *timed automata* (TA) [**?**], and, more generally, for a timeless stack, i.e., for PDTA [**?**], the complexity improves by one exponential (cf. [Theorem 5, A], second part).
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 (Timeless stack TPDA and TA **TPDA z bezczasowym sostem i TA [A])**
The reachability relation $\rightsquigarrow$ for timeless stack TPDA *and for* TA *can be expressed by a family of existential linear arithmetic formulas computable in* ExpTime.
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.

Our results above improve similar expressibility results from the literature on TA [**?**, **?**, **?**, **?**] and PDTA [**?**]. Along the way, we prove a result interesting on its own about effective elimination of quantifiers for a sublogic of linear arithmetic corresponding to clock constraints [Corollary 3, A]. Quantifier elimination allows us to reduce to TPDA where all constraints are fractional. Finally, we reduce fractional TPDA to DPDA over cyclic order atoms, introduced earlier in Sec. **??**, to which we apply our previous results [D]. This is interesting, since it shows the applicability of a register model, such as DPDA, to the analysis of a clock model, such as TPDA.
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

We have provided an extensive investigation of the reachability problem for expressive classes of infinite state systems combining unbounded discrete data structures with expressive timing constraints, and thus "infinite in two dimensions". In particular, we have considered FIFO communication channels in our works [E] and [F], and pushdown stacks in [A], [B], [C], and [D].
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].

A general picture emerging from our investigations is that, generally speaking, the reachability problem stays decidable, albeit the computational complexity sometimes worsens. For instance, in the case of communicating automata (CA; works [E] and [F]) it was known that reachability is decidable iff the topology is a polyforest. From our results it follows that, if we add time to the model (obtaining TCA), then the set of decidable topologies does not change (without emptiness tests; cf. Theorem **??**). However, the complexity is ExpSpace-complete if we allow the processes to elapse an arbitrary amount of time at the end of the run, and non-elementary in the general case (cf. Theorem **??**). This should be contrasted with the fact that, in the untimed setting, reachability is PSpace-complete.
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.

Similarly, for timed pushdown automata (TPDA) the reachability problem is still decidable, however, the complexity worsens from PTime in the untimed setting, to ExpTime-complete in the timed setting. On the positive side, the ExpTime upper bound holds for DPDA over a very large class of "qualitative data" (tractable homogeneous atoms; cf. Theorem **??** [D]), for trPDA and TPDA with timeless stack (cf. Theorem **??** [C] and, resp., **??** [A]), and trPDA over monotonic time [**?**, Corollary 6.8]. For the even more expressive classes of trPDA and TPDA (with timed stack), we have provided a 2ExpTime complexity upper bound (cf. Theorem **??** [B] and, resp., **??** [A]). As future work, it remains to investigate whether the 2ExpTime complexity above is optimal.

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 Other publications
Inne publikacje

5.1 Publications on higher-order recursive schemes
Prace o schematach rekurencyjnych wyższego rzędu

The following two papers concern the algorithmic analysis of *higher-order recursive schemes* (HORS), a very expressive formalism used to model higher-order recursion, i.e., functions taking other functions as argument.
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 [**?**].

  In this paper we propose an alternative approach on the verification of HORS based on tree-pushdown automata, which generalise ordinary pushdown automata by allowing the stack to be a tree instead of a word. We prove a general *preservation of regularity* result: The inverse image $(\to^*)^{-1}(T)$ of the reachability relation $\to^*$ of a regular set of trees T is itself a regular set of trees. The result is effective, in the sense that given a finite tree automaton for the target set T, we compute a finite tree automaton for the inverse image of T.
  W tej pracy proponujemy alternatywne podejście do weryfikacji HORS oparte na automatach ze stosem drzewiastym, uogólniających zwyczajyne 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 [**?**].

  The main contribution of this work is proving that the following non-trivial non-regular property of HORS is decidable: Is it the case that, for every bound $n \in \mathbb{N}$, the HORS recognises a finite tree where each letter from a finite alphabet occurrs at least n times? Note that such unboundedness property is not regular, and thus decidability does not follow from [**?**]. Decidability of the diagonal problem has a number of interesting consequence, such as computability of the downward closure over finite words [**?**], decidability of the separability problem by piecewise testable languages [**?**], and decidability of reachability of parameterized asynchronous shared-memory concurrent systems [**?**].
  Głównym 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 Publications on separability problems
Prace o problemach separowalności

A set S *separates* two sets A, B if $S \supseteq A$ and $S \cap B = \varnothing$. Given two classes of sets $\mathcal{A}$ and $\mathcal{S}$, the *separability problem* asks whether two given sets $A, B \in \mathcal{A}$ can be separated by some set $S \in \mathcal{S}$. The separability problem is a central topic in computer science (such as in formal language theory

and computability theory) and mathematics (for example in topology). In computer science separability is intimately related to other important problems, such as computing downward closures, the emptiness of intersection problem, and the characterisation problem. In the following papers we study two separability problems.
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 [**?**].

  We consider as input sets $\mathcal{A}$ the class of subsets of $\mathbb{N}^k$ which are reachability sets of vector addition systems (and generalisation thereof), and as separators $\mathcal{S}$ the class of modular and unary subsets of $\mathbb{N}^k$. Our main result is that the separability problem is decidable. This is a first step towards a more ambitious goal: Establishing decidability of regular separability of languages recognised by vector addition systems.
  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 [**?**].

  We consider as input sets $\mathcal{A}$ languages (i.e., subsets of Σ^*) recognised by Parikh automata (finite automata with acceptance by counting transitions), and as separators $\mathcal{S}$ the class of regular languages. Our main result is decidability of the separability problem. This is somewhat surprising, since the regularity problem for Parikh automata is undecidable. On the way, we show decidability of separability of semilinear sets of $\mathbb{N}^k$ by unary and modular sets.
  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 Publications on stochastic games
Prace na temat gier stochastycznych

The following two publications study stochastic games [**?**] on finite and infinite graphs.
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 [**?**] (cf. also the survey por. [**?**]).

  The *beyond worst-case* (BWC) problem combines determinstic 2-player games and stochastic 1-player games: In its original formulation [**?**], it asks whether the player can win against an arbitrary deterministic opponent, and at the same optimise the expected value w.r.t. a (fixed) stochastic model of the opponent. Our main result is coNPTime-completeness of the multi-dimensional BWC problem on finite graphs, for both finite-memory and unrestricted strategies; we also show that, if we relax the first requirement to almost-sure winning, then the problem becomes solvable in PTime. This generalises previous work in one-dimension [**?**].

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 [?] (special journal issue of the conference paper
  specjalna wersja czasopismowa artykułu konferencyjnego [?]).

  We study a class of stochastic games played on infinite, yet well-behaved graphs, namely, configuration graphs of lossy channel systems. We consider almost-sure winning w.r.t. ω-regular objectives and we show that such games are decidable under the constraint that both player use finite-memory strategies. This is a necessary assumption, since they are undecidable already for almost-sure co-Büchi objectives [?].
  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 Publications on language inclusion and automata simplification
Prace na temat inkluzji języków i upraszczania automatów

The following works tackle the related problems of efficiently deciding language inclusion of two automata and efficiently reducing the size of automata. Checking inclusion between is a central problem in automata theory, with important applications such as *model-checking* [?] and *size-change termination* [?]. Automata simplification is fundamental in automata-based decision procedures for logical theories [?], in the analysis of infinite state systems such as Petri nets [?], and in regular model checking [?]. Both problems are PSpace-complete in general, and thus efficient algorithms capable of solving practically arising instances are important. We focus of languages of infinite words recognised by nondeterministic and alternating Büchi automata.
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 [?].

  A successful approach to the reduction of the number of states of an automaton is that of quotienting. It relies on the identification of equivalence relations on states s.t. 1) computing the relation should be efficient, and 2) quotienting should preserve the language recognised by the automaton. Suitable simulations and bisimulation relations have previously been defined for nondeterministic [?] and alternating Büchi automata [?]. A generalisation to

multipebble simulations was proposed for nondeterministic Büchi automata [**?**]. We join the last to works and study multipebble simulations for alternating Büchi automata. We show that, multipebble simulations imply language inclusion, a variant thereof can be used for quotienting, and they can be computed in PTime for any fixed number of pebbles.
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 [**?**].

  We propose an efficient algorithm to solve the universality and inclusion problems for nondeterministic Büchi automata. It was previously shown how to use set-based subsumption to improve the efficiency of a Ramsey-based algorithm for the universality problem [**?**]. We improve the approach by using simulation relations to obtain even larger subsumption relations, and we extend the method from universality to the more general inclusion problem. We show that this results in a significant performance gain on both random automata and test cases taken from mutual exclusion algorithms.
  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 [**?**].

  We improve on the previous point by defining an even coarser subsumption relation based on forward as well as backward simulations, plus other algorithmic improvements.
  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 [**?**].

  We investigate generalisations of simulation relations for Büchi automata suitable for quotienting. We propose *fixed-word delayed simulation* to be the maximal such relation which can be used for quotienting. To support this claim, we show that its multipebble extensions collapses to the one-pebble case.
  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 [**?**] (journal version of
  wersja czasopismowa pracy [**?**]).

  We present efficient algorithms to reduce the size of nondeterministic finite and Büchi automata. We propose extensions of classic quotienting of the state space, as well as innovative

techniques for removing transitions (pruning) and adding transitions (saturation)[14]. These techniques use criteria based on combinations of backward and forward trace inclusions and simulation relations. Since trace inclusion relations are themselves PSPACE-complete, we introduce *lookahead simulations* as good PTIME approximations thereof. The quality of the reduction is witnessed by the fact that it often reduces universal automata to the trivial one-state universal automaton. In general, extensive experiments show that our algorithm consistently reduces the size far more than all previous techniques.

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.

[14] Adding transitions may cause more states to be quotiented together.