

Autoreferat

Piotr Hofman

1 Imię i nazwisko

Imię: Piotr Hofman

2 Posiadane dyplomy, stopnie naukowe lub artystyczne – z podaniem podmiotu nadającego stopień, roku ich uzyskania oraz tytułu rozprawy doktorskiej

<i>Mar. 2014</i>	Nadanie stopnia doktora informatyki (Wydział Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego), tytuł pracy: “Equivalence of infinite-state systems with silent steps”, promotor: prof. dr hab. Sławomir Lasota.
<i>2009 – 2013</i>	Studia doktoranckie na Wydziale Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego.
<i>Gru. 2008</i>	Uzyskanie tytułu magistra matematyki (Wydział Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego), tytuł pracy: “On some new facts about the EKG sequence”, promotor: prof. dr hab. Jerzy Urbanowicz.
<i>2003 – 2009</i>	Jednoczesne Studia Informatyczno-Matematyczne na Wydziale Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego.

3 Informacja o dotychczasowym zatrudnieniu w jednostkach naukowych lub artystycznych

<i>Mar. 2021 – teraz</i>	Delegacja do pracy w projekcie ERC “Challenging Problems in Infinite-State Systems” (INFSYS).
<i>Paź. 2017 – teraz</i>	Zatrudnienie na pozycji adiunkta na Wydziale Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego.
<i>Paź. 2016 – Wrz. 2017</i>	Zatrudnienie na pozycji Adiunkta naukowego na Wydziale Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego.
<i>Paź. 2014 – Wrz. 2016</i>	Staż podoktorski w grupie prof. Stefana Göllera, Laboratoire Spécification et Vérification, Centre National de la Recherche Scientifique & École Normale Supérieure de Cachan (LSV, CNRS & ENS de Cachan), Paryż, Francja.
<i>Lis. 2013 – Wrz. 2014</i>	Staż podoktorski w grupie prof. Wima Martensa, na Uniwersytecie w Bayreuth, Niemcy.

- 4 Omówienie osiągnięć, o których mowa w art. 219 ust. 1 pkt. 2 ustawy z dnia 20 lipca 2018 r. Prawo o szkolnictwie wyższym i nauce (Dz. U. z 2021 r. poz. 478 z późn. zm.). Omówienie to winno dotyczyć merytorycznego ujęcia przedmiotowych osiągnięć, jak i w sposób precyzyjny określać indywidualny wkład w ich powstanie, w przypadku, gdy dane osiągnięcie jest dziełem współautorskim, z uwzględnieniem możliwości wskazywania dorobku z okresu całej kariery zawodowej

Tytuł

Systemy nieskończone stanowe z danymi.

Publikacje uwzględnione w osiągnięciu naukowym:

Publikacje konferencyjne

- [4] P. Hofman, S. Lasota, R. Lazic, J. Leroux, S. Schmitz, and P. Totzke: *Coverability Trees for Petri Nets with Unordered Data*; FoSSaCs, 2016.

Równy wkład wszystkich autorów. 16,5 %

- [5] P. Hofman, J. Leroux, P. Totzke: *Linear combinations of unordered data vectors*; LICS, 2017.

Zaproponowałem temat badawczy i miałem znaczący wkład kombinatoryczny. Na przykład udowodniłem twierdzenie o charakteryzacji histogramów, które jest kluczowe dla pracy. 65 %

- [6] P. Hofman, S. Lasota: *Linear Equations with Ordered Data*. CONCUR 2018

Zaproponowałem temat badawczy i odegrałem główną rolę w zaprojektowaniu algorytmu. Sławomir Lasota jest głównym autorem dowodu trudności obliczeniowej problemu. 60 %

- [7] U. Gupta, P. Shah, S. Akshay, P. Hofman: *Continuous Reachability for Unordered Data Petri Nets is in PTime*. FoSSaCs 2019

Zaproponowałem temat badawczy oraz sposób jego rozwiązania. Większość szczegółów technicznych dowodów to wkład dwóch studentów. Ja i Akshay S. nadzorowaliśmy studentów podczas pisania publikacji. 40 %

- [9] P. Hofman, M. Jucepczuk, S. Lasota, M. Pattathurajan: *Parikh's theorem for infinite alphabets*. LICS 2021

Zaproponowałem temat badawczy i mam istotny wkład w dowody. Na przykład zaproponowałem wykorzystanie teorii ścieżek Hamiltona w grafach skierowanych w dowodzie jednego z dwóch głównych twierdzeń. Zaproponowałem także technikę 'cięcia i wklejania' opartą na kompatybilności węzłów w drzewie derywacji. Zaproponowałem również wykorzystanie twierdzenia Ramseya do udowodnienia, że kompatybilne węzły muszą istnieć w wystarczająco dużym drzewie derywacji. 40 %

- [10] A. Ghosh, P. Hofman, S. Lasota: *Solvability of orbit-finite systems of linear equations*. LICS 2022

Początkowo zaproponowałem badania tylko nad rozwiązaniami skończonymi. Perspektywa algebry liniowej pochodzi od Sławomira Lasoty i w związku z tym to jemu przypada autorstwo kierunku badań. Jestem autorem głównych pomysłów kombinatorycznych wykorzystywanych w algorytmie do rozwiązywania równań. Twierdzenie o istnieniu bazy zostało opracowane przez Arkę Ghosha. 33 %

- [11] A. Ghosh, P. Hofman, S. Lasota: *Orbit-finite linear programming. LICS 2023* (Artykuł wyróżniony).

Pytanie badawcze zostało postawione przez Arkę Ghosha. W algorytmie do rozwiązywania układów nierówności nad ciałem \mathbb{Q} zaproponowałem ulepszoną procedurę wyrównywania rozwiązań, podczas gdy Arka Ghosh jest autorem rozwiązania uzyskanego za jej pomocą układu równań wielomianowych. Odegrałem również wiodącą rolę w procesie dowodzenia nierozstrzygalności nierówności nad \mathbb{Z} . 33 %

Publikacje w czasopismach naukowych

- [8] Piotr Hofman, Jakub Różycki: *Linear equations for unordered data vectors in $[D]^k \rightarrow Z^d$. Logical Methods Computer Science 18(4) 2022*

Przedstawiłem ogólną strategię dowodu i przeprowadziłem dowód dla jednego z przypadków. Zaproponowałem studentowi Jakubowi Różyckiemu pracę nad uogólnieniem wyników. Jakub Różycki jest autorem większości technicznych lematów. Nadzorowałem studenta podczas pisania artykułu. 65 %

4.1 Opis osiągnięcia

4.1.1 Omówienie

Wyniki, z których składa się osiągnięcie naukowe, skupiają się wokół modelu sieci z danymi. Model ten rozszerza sieci Petriego. Osiągnięcie dotyczy nowych technik dla tego modelu obliczeniowego, ale wskazuje także na możliwość zastosowania opracowanych technik dla innych modeli. Największy wkład to, opracowane metod rozwiązywania nieskończonych systemów równań i nierówności liniowych [5, 6, 10, 11, 8], które mogą pojawić się, gdy próbujemy rozszerzyć wyniki znane dla zwykłych sieci Petriego do sieci z danymi. Ponadto prezentuje rozszerzenie do sieci z danymi dwóch innych technik znanych dla sieci Petriego [4, 7]. Ostatni element osiągnięcia wskazuje gdzie można wykorzystać dotychczas opracowane techniki do analizy systemów opisanych przy pomocy innych formalizmów [9].

4.1.2 Wprowadzenie

Jednym z ugruntowanych działów teoretycznej informatyki jest analiza różnych modeli obliczeniowych. Najbardziej podstawowym z nich jest automat skończony, który jest maszyną o skończonej liczbie możliwych konfiguracji. Do klasycznych rozszerzeń tego modelu zaliczyć można automaty ze stosem, sieci Petriego¹, i maszyny Turinga. W tych modelach nieskończona liczba możliwych konfiguracji wynika z nieograniczonej liczby komórek pamięci.

Przełom lat 80. i 90. ubiegłego wieku wyraźnie pokazał, że te klasyczne podejścia nie są przydatne do modelowania pewnych aspektów systemów, takich jak na przykład czas. Dlatego naukowcy zaczęli rozważać automaty skończone rozszerzone o skończoną liczbę komórek pamięci zdolnych przechowywać dane z dziedziny nieskończonej, na przykład wspomniany już czas. Oczywiście, w tych modelach sposoby manipulacji danymi są ograniczone. Niektóre z najbardziej znanych modeli obliczeń tego rodzaju to automaty czasowe [12] i automaty rejestrowe [13]. Naturalnym kolejnym krokiem było połączenie nieskończonej liczby komórek pamięci i alfabetów nieskończonych, jak to ma miejsce w automatach danowych ze stosem [14], rachunku π [15], czy sieciach Petriego z danymi [24].

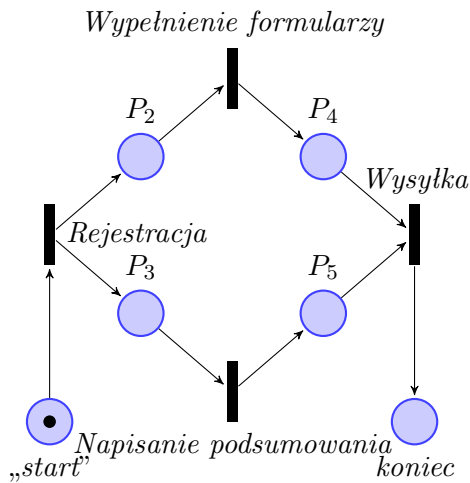
Punkt wyjścia dla tej pracy badawczej sistanowiło pytanie dotyczące osiągalności w sieciach z danymi bez porządku. Rozpocniemy od opisu tego modelu i pytania o osiągalność, które jest centralne dla tej serii prac. Następnie przejdziemy do wstępnych wyników [4] i [5]. W dalszej kolejności wskażę, jak późniejsze badania wynikły z wstępnych prac nad osiągalnością i opiszę uzyskane rezultaty.

4.1.3 Nieformalne wprowadzenie do sieci z danymi

Sieci Petriego zostały opracowane przez Carla Adama Petriego [16] i są badane od ponad 50 lat. Sieci Petriego są zdefiniowane poniżej za pomocą reprezentacji graficznej i prostej semantyki. Z jednej strony

¹Początkowo Petri definiował je jako "proste sieci"; tutaj odnosimy się do sieci PT (PT-nets)

są one proste i przydatne do modelowania procesów nawet dla osób, które nie są informatykami. Z drugiej strony ich potencjalnie bardzo złożone zachowanie [17] sprawia, że stanowią one interesujący i wymagający temat badań.



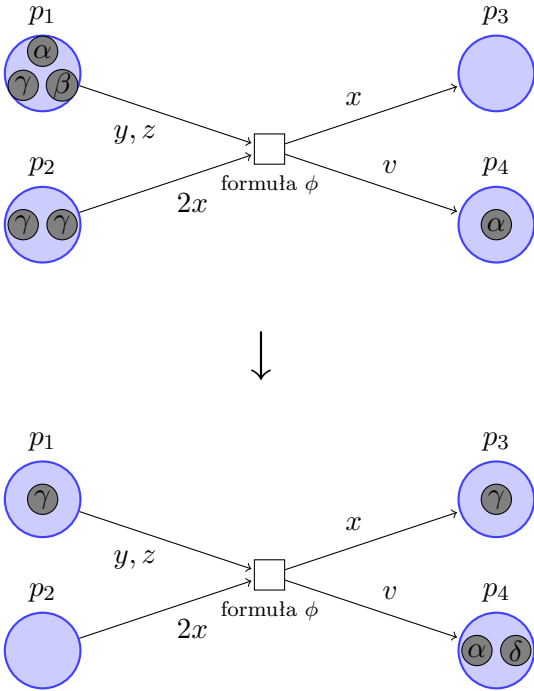
Rysunek 1:

Definicja 1 Sieć Petriego jest zdefiniowana jako trójka $(\mathbb{P}, \mathbb{T}, \mathbb{E})$, gdzie \mathbb{P} reprezentuje miejsca (ilustrowane jako niebieskie koła), \mathbb{T} reprezentuje tranzycje (ilustrowane jako czarne prostokąty), a \mathbb{E} reprezentuje krawędzie (ilustrowane jako czarne strzałki). Graf Sieci Petriego jest dwudzielny; nie ma krawędzi między dwoma miejscami ani między dwoma przejściami. Do sieci Petriego wprowadzamy żetony (ilustrowane jako czarne kropki na diagramie). Formalnie, mówimy o markowaniu sieci Petriego. Sieci Petriego symulują procesy poprzez tworzenie i usuwanie żetonów zgodnie z następującą zasadą: Wybierz przejście $t \in \mathbb{T}$, usuń jeden żeton z każdego miejsca połączonego z t za pomocą krawędzi wchodzących i dodaj jeden żeton do każdego miejsca połączonego z t za pomocą krawędzi wychodzących. Ten proces nazywamy odpaleniem tranzycji t .

Przykład 2 W sieci przedstawionej na rysunku 1 możemy odpalić tranzycję „Rejestracja”, które usuwa jeden żeton z miejsca „start” i umieszcza jeden żeton zarówno w P_2 , jak i P_3 . W tym momencie mamy wybór co do kolejnych tranzycji: albo „Wypełnienie formularzy”, albo „Napisanie podsumowania”. Odpalenie pierwszej tranzycji przenosi jeden żeton z P_2 do P_4 , podczas gdy odpalenie drugiego przenosi jeden żeton z P_3 do P_5 . Następnie możemy odpalić tranzycję „Wysyłka”. Ważne jest zaznaczenie, że nie możemy odpalić tranzycji „Wysyłka” dopóki nie zakończymy pisania podsumowania i nie skończymy z formularzami, ponieważ odpalenie tranzycji „Wysyłka” wymaga żetonów w P_4 i P_5 .

Mimo wielu sukcesów, sieci Petriego nie zawsze wystarczają do modelowania pewnych aspektów systemów. Pytania algorytmiczne dotyczące rozszerzeń i modyfikacji sieci Petriego są przedmiotem wielu badań, co czyni je ważną częścią informatyki teoretycznej [18, 4, 19, 20, 21, 22, 23].

Relatywnie nowy obszar badań pojawił się, gdy grupa naukowców, inspirując się automatami rejestrowymi, zdefiniowała sieci z danymi jako sieci Petriego, których żetony przenoszą dane [24, 25, 26].



Rysunek 2:

Pierwszą istotną różnicą między sieciami Petriego a sieciami z danymi jest to, że w sieci z danymi każdy żeton przenosi pewne dane (w początkowych pracach pojedynczą daną) z dziedziny danych \mathbb{D} . Dane reprezentujemy za pomocą greckich liter wewnątrz żetonów. Oczywiście dane wpływają na zachowanie sieci. Osiąga się to poprzez zmienne, ze skończonego zbioru \mathbb{V} , etykietujące krawędzie i formułę logiczną ϕ etykietującą tranzycję. Formuła ϕ wykorzystuje zmienne etykietujące krawędzie oraz pewne predykaty, których jeszcze nie określiliśmy. Aby odpalić tranzycję, przypisujemy wartości z \mathbb{D} do zmiennych, tak aby formuła była spełniona. Następnie zużywamy i produkujemy żetony zgodnie z wartościowaniem zmiennych. Na przykład, jeśli formuła ϕ określa, że wszystkie zmienne muszą być różne, tj. $\phi = (x \neq y \neq z \wedge x \neq z \wedge x \neq u \neq y \wedge u \neq z)$, to możemy przypisać następujące wartości zmiennym: $x = \gamma, y = \alpha, z = \beta, u = \delta$. Aby odpalić tranzycję z proponowanym wartościowaniem zmiennych, zużywamy z miejsca p_1 jeden żeton z daną α i jeden żeton z daną β , z miejsca p_2 zużywamy 2 żetony z daną γ , w miejscu p_3 produkujemy żeton z daną γ , a w miejscu p_4 tworzymy żeton z daną δ .

Różne klasy sieci z danymi odzwierciedlają różne dziedziny danych \mathbb{D} , na przykład \mathbb{D} może być uporządkowanym zbiorem lub zbiorem bez porządku, oraz z różne dostępne predykaty, których można użyć w formułach etykietujących tranzycję. Motywacją do rozważania różnych dziedzin danych są różne potrzeby pojawiające się przy modelowaniu. Na przykład, z jednej strony, dane nieuporządkowane z nierównościami (\neq) mogą być używane do modelowania identyfikatorów procesów lub unikalnych zasobów, z drugiej strony, jeśli dane są uporządkowane i można je porównywać, możemy zakodować w nich zależności czasowe. Oczywiście, im bardziej skomplikowana struktura danych, tym bardziej skomplikowana obliczeniowo staje się ich analiza. Dlatego w sieciach z danymi naukowcy skupiają się tylko na bardzo prostych dziedzinach².

Na przykład, w pracy [24] autorzy skoncentrowali się na $\mathbb{D} = \mathbb{Q}$, tj. zbiorze liczb wymiernych i pozwolili tylko na predykaty sprawdzające nierówności. Ważną właściwością takich sieci jest to, że jeśli π jest monotoniczną bijekcją $\mathbb{Q} \rightarrow \mathbb{Q}$, a σ jest wartościowaniem zmiennych $\mathbb{V} \rightarrow \mathbb{Q}$, to $\sigma \circ \pi$ również jest poprawnym wartościowaniem. To spostrzeżenie jest kluczowe dla sformalizowania sieci z danymi jako obiektów matematycznych, z którymi wygodniej jest pracować, przynajmniej z perspektywy ich analizy.

Powyższe definicje sieci Petriego i sieci z danymi są skrojone do modelowania. Z czysto matematycznego punktu widzenia wielu badaczy, w tym ja sam, woli pracować z *Systemami Dodawania Wektorów (VAS)*, które są równoważne sieciom Petriego. O sieciach wspominamy tutaj, ponieważ ta perspektywa jest bardziej intuicyjna i pomaga zrozumieć różne definicje oraz stawiane pytania badawcze, które są rozwiązywane w zbiorze prac składających się na to osiągnięcie.

²W porównaniu z kolorowanymi sieciami Petriego.

4.1.4 Systemy Dodawania Wektorów i Systemy Dodawania Wektorów z Danymi³

W dalszej części używamy liter $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{Q}_{\geq 0}$, aby oznaczać zbiory liczb naturalnych, całkowitych, wymiernych i nieujemnych wymiernych. Używamy czcionki \mathbf{m} i dużych liter do oznaczania zbiorów oraz pogrubionych małych liter do oznaczania wektorów.

Porządek jest *dobrym quasi-porządkiem* (WQO), jeśli nie zawiera nieskończonego antyłańcucha ani nieskończonego ciągu malejącego. Na przykład \mathbb{N} jest WQO, ale \mathbb{Z} nie jest. Dickson udowodnił, że jeśli \mathbb{X} i \mathbb{Y} są WQO, to $\mathbb{X} \times \mathbb{Y}$ z porządkiem po współrzędnych $(x, y) \sqsubseteq (x', y') \iff (x \leq x') \text{ i } (y \leq y')$ jest także WQO. Dla podzbioru \mathbb{X}' uporządkowanego zbioru \mathbb{X} , przez domknięcie w dół $\mathbb{X}' \downarrow$ rozumiemy najmniejszy zbiór \mathbb{X}'' taki, że $\mathbb{X}' \subseteq \mathbb{X}''$ i jeśli $y \in \mathbb{X}''$ oraz $x < y$, to $x \in \mathbb{X}''$. Podobnie, podzbiór \mathbb{X}'' jest domknięciem w górę zbioru \mathbb{X}' , oznaczanym przez $\mathbb{X}' \uparrow$, jeśli jest to najmniejszy zbiór taki, że $\mathbb{X}' \subseteq \mathbb{X}''$ i jeśli $x \in \mathbb{X}''$ oraz $x < y$, to $y \in \mathbb{X}''$.

System tranzycyjny to skierowany graf etykietowany; jego wierzchołki nazywane są konfiguracjami, a krawędzie - krokami. Zapisujemy $m \xrightarrow{e} m'$, aby oznaczać krawędź opisaną etykietą e z wierzchołką m do wierzchołką m' . Zapisujemy także $m \rightarrow m'$, jeśli istnieje etykieta e tak, że $m \xrightarrow{e} m'$. Ponadto, przez $m \xrightarrow{*} m'$ rozumiemy domknięcie przechodnie relacji \rightarrow . Nazywamy to relacją osiągalności i mówimy, że istnieje *bieg* z konfiguracji m do konfiguracji m' . Dla zbioru konfiguracji \mathbb{X} zapisujemy $m \rightarrow \mathbb{X}$, jeśli istnieje $x \in \mathbb{X}$ takie, że $m \xrightarrow{*} x$; nazywamy to osiągalnością zbioru \mathbb{X} . Zbiór konfiguracji osiągalnych z danej konfiguracji m oznaczamy jako $Reach(m)$.

Definicja 3 System Dodawania Wektorów (VAS) to zbiór \mathbb{T} wektorów w przestrzeni \mathbb{Z}^k , gdzie k to liczba miejsc w odpowiadającej mu sieci Petriego. Stąd często zamiast o wymiarach od 1 do k mówimy o miejscach w VAS.

VAS indukuje system tranzycyjny, gdzie konfiguracje (tj. markowania) to wektory w przestrzeni \mathbb{N}^k . W systemie tranzycyjnym istnienie krok $\mathbf{m} \xrightarrow{\mathbf{t}} \mathbf{m}'$, jeśli istnieje $\mathbf{t} \in \mathbb{T}$ taki, że $\mathbf{m}' = \mathbf{t} + \mathbf{m}$.

Dla przykładu VAS odpowiadający sieci Petriego z Rysunku 1 ma cztery wektory:

$$\begin{pmatrix} -1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ -1 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ -1 \\ 0 \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ -1 \\ -1 \\ 1 \end{pmatrix} \text{ a konfiguracją początkową jest } \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Co więcej, odpalenie tranzycji "Rejestracja" z rysunku 1, tłumaczy się na

$$\text{następujący krok } \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \xrightarrow{\begin{pmatrix} -1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Dla wektora $\mathbf{t} \in \mathbb{T}$ używamy oznaczeń $\bullet \mathbf{t}$ i $\mathbf{t} \bullet$ do określenia zbiorów miejsc (wymiarów) p , takich że odpowiednio $\mathbf{t}(p) < 0$ i $\mathbf{t}(p) > 0$. Dla wektora $\mathbf{v} \in \mathbb{Q}_{\geq 0}^{\mathbb{T}}$, $\bullet \mathbf{v}$ oznacza $\bigcup_{\mathbf{t} \in \{\mathbf{t} \in \mathbb{T} \mid \mathbf{v}(\mathbf{t}) > 0\}} \bullet \mathbf{t}$. Podobnie definiujemy $\mathbf{v} \bullet$.

Zbiór konfiguracji to \mathbb{N}^k , zatem jest on w naturalny sposób uporządkowane po współrzędnych \sqsubseteq .

Dla danego VAS-u \mathbb{T} i dwóch jego konfiguracji \mathbf{m} i \mathbf{m}' , społeczność definiuje kilka klasycznych problemów:

³Zdecydowałem się używać bardziej klasycznego słowa "dane" w tym dokumencie, chociaż nazwa, która jest obecnie najczęściej używana, to "atomy". Podkreśla ona koncepcję enkapsulacji danych. Aby dowiedzieć się więcej [1, 2, 3].

- Problem osiągalności: pytamy, czy $\mathbf{m}' \in \text{Reach}(\mathbf{m})$.
- Problem pokrywalności: sprawdzamy, czy $\mathbf{m}' \in \text{Reach}(\mathbf{m}) \downarrow$.
- Problem braku terminacji: badamy, czy istnieje nieskończona ścieżka rozpoczynająca się w \mathbf{m} .
- Problem nieograniczoności i nieograniczoności miejsca p : badamy, czy zbiór $\text{Reach}(\mathbf{m})$ jest nieskończony, oraz czy dla każdego $n \in \mathbb{N}$ istnieje $\mathbf{m}'' \in \text{Reach}(m)$ taka, że $\mathbf{m}''(p) > n$.
- Problem istnienie blokady: ten problem jest równoważny osiągalności pewnego domkniętego w dół zbioru.
- Problem żywotność: sprawdzamy, czy dla każdego $\mathbf{m}' \in \text{Reach}(\mathbf{m})$ i $\mathbf{t} \in \mathbb{T}$ istnieje konfiguracja $\mathbf{m}'' \in \text{Reach}(\mathbf{m}')$ taka, że $\mathbf{t} \sqsubseteq \mathbf{m}''$.

Każdy z powyższych problemów, z wyjątkiem ostatnich dwóch, które można zredukować do osiągalności, wiąże się z konkretnymi technikami, które zostały opracowane na przestrzeni lat:

- **Osiągalność:** Trudno jest zidentyfikować pojedynczą technikę, na której opiera się algorytm osiągalności [27, 28, 29, 30, 31], a także jednoznacznie opisać pomysły kluczowe dla niedawnych dowodów trudności problemu [32, 33, 34]. W następnej sekcji zajmiemy się niektórymi elementami składowymi algorytmu osiągalności.
- **Pokrywanie:** Teoria dobrze ustrukturyzowanych systemów tranzycyjnych, a konkretnie algorytm pokrywania wstecz [35].
- **Problem braku terminacji:** Teoria dobrze ustrukturyzowanych systemów tranzycyjny, tym razem konkretnie wskazujemy na algorytm wprzód oraz na drzewo-pokrywalności [35].
- **Problem ograniczoności i ograniczoności miejsca p :** Najczęściej wykorzystywany jest algorytm drzewa pokrywalności [36].

VAS z danymi

Z tym wprowadzeniem jesteśmy gotowi, aby zdefiniować VAS z danymi. Ponieważ każdy token przenosi dane, przeddefiniujemy konfiguracje jako wektory z danymi. Wybieramy nieskończoną dziedzinę danych \mathbb{D} ; używamy małych liter greckich $\alpha, \beta, \gamma, \delta, \sigma, \phi$ do oznaczenia poszczególnych wartości danych, a greckiej litery ξ do oznaczania zmiennej przyjmującej wartości w \mathbb{D} . Niech \mathbf{Aut} oznacza zbiór automorfizmów dziedziny danych, a literę π rezerwujemy na jego elementy.

Definicja 4 (Wektory danych.) Niech $k \in \mathbb{N}$. Wektor danych to funkcja o skończonym nośniku $\mathbf{v}: \mathbb{D} \rightarrow \mathbb{Q}^k$, gdzie skończony nośnik oznacza, że $\mathbf{v}(\xi) = \mathbf{0}$ dla wszystkich, poza skończoną liczbą, danych $\xi \in \mathbb{D}$.

Wektory danych można sumować po współrzędnych, tzn. $(\mathbf{v} + \mathbf{v}')(\xi) \stackrel{\text{def}}{=} \mathbf{v}(\xi) + \mathbf{v}'(\xi)$ dla wszystkich $\xi \in \mathbb{D}$. Podobnie, dla $a \in \mathbb{Q}$, można mnożyć wektor z danymi \mathbf{v} przez stałą a , precyzyjnie $a\mathbf{v}(\xi) = a \cdot \mathbf{v}(\xi)$ dla każdego $\xi \in \mathbb{D}$.

Mówimy, że wektor z danymi jest nieujemny, jeśli jest funkcją $\mathbf{v}: \mathbb{D} \rightarrow \mathbb{Q}_{\geq 0}^k$.

Wreszcie, dla $\pi \in \mathbf{Aut}$, przez $\mathbf{v} \circ \pi$ mamy na myśli wektor z danymi $(\mathbf{v} \circ \pi)(\xi) = \mathbf{v}(\pi(\xi))$ dla wszystkich $\xi \in \mathbb{D}$.

Podstawowa koncepcja sieci z danymi została wprowadzona w [24]. Poniżej definicja wyrażona w języku VAS-ów z danymi.

Definicja 5 System dodawania wektorów z danymi to (zwykle nieskończony) zbiór \mathbb{T} wektorów z danymi w $\mathbb{D} \rightarrow \mathbb{Z}^k$, który jest zamknięty względem działania elementami grupy \mathbf{Aut} , oraz ma skończenie wiele orbit względem tego działania.

Ponieważ VAS z danymi ma skończoną liczbę orbit, można go przedstawić jako reprezentantów każdej orbity wraz z opisem grupy automorfizmów \mathbf{Aut} .

Indukuje on etykietowany system tranzycyjny, gdzie konfiguracje są nieujemnymi całkowitoliczbowymi wektorami z danymi, to znaczy należą do $\mathbb{D} \rightarrow \mathbb{N}^k$, a krok $\mathbf{v} \xrightarrow{\mathbf{t}} \mathbf{v}'$ dla $\mathbf{t} \in \mathbb{T}$ istnieje w systemie, jeśli $\mathbf{v}' = \mathbf{v} + \mathbf{t}$.

W przypadku biegu w etykietowanym systemie tranzycyjnym indukowanym przez VAS z danymi, mówimy, że dana $\xi \in \mathbb{D}$ pojawia się w biegu, jeśli istnieje konfiguracja \mathbf{m} w trakcie biegu, taka że $\mathbf{m}(\xi) \neq \mathbf{0}$.

Przykład 6 System VAS z danymi odpowiadający pojedynczej tranzycji na rysunku 2 jest zdefiniowany dla nieskończonej dziedziny danych \mathbb{D} , gdzie \mathbf{Aut} to wszystkie bijekcje $\mathbb{D} \rightarrow \mathbb{D}$. Ma jedną orbitę i może być reprezentowany jako pojedynczy wektor z danymi:

$$\begin{bmatrix} & \alpha & \beta & \gamma & \delta & \sigma & \dots \\ p_1 & -1 & -1 & 0 & 0 & 0 & 0 \\ p_2 & 0 & 0 & -2 & 0 & 0 & 0 \\ p_3 & 0 & 0 & 1 & 0 & 0 & 0 \\ p_4 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (1)$$

Jeśli zastosujemy do niego automorfizm danych, który zamienia α i γ , pozostawiając wszystkie inne dane bez zmian, otrzymamy inną możliwą instancję tranzycji:

$$\begin{bmatrix} & \alpha & \beta & \gamma & \delta & \sigma & \dots \\ p_1 & 0 & -1 & -1 & 0 & 0 & 0 \\ p_2 & -2 & 0 & 0 & 0 & 0 & 0 \\ p_3 & 1 & 0 & 0 & 0 & 0 & 0 \\ p_4 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

Odpalenie tranzycji z Rysunku 2 tłumaczymy jako krok

$$\begin{bmatrix} & \alpha & \beta & \gamma & \delta & \sigma & \dots \\ p_1 & 1 & 1 & 1 & 0 & 0 & 0 \\ p_2 & 0 & 0 & 2 & 0 & 0 & 0 \\ p_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ p_4 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} & \alpha & \beta & \gamma & \delta & \sigma & \dots \\ p_1 & -1 & -1 & 0 & 0 & 0 & 0 \\ p_2 & 0 & 0 & -2 & 0 & 0 & 0 \\ p_3 & 0 & 0 & 1 & 0 & 0 & 0 \\ p_4 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} & \alpha & \beta & \gamma & \delta & \sigma & \dots \\ p_1 & 0 & 0 & 1 & 0 & 0 & 0 \\ p_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ p_3 & 0 & 0 & 1 & 0 & 0 & 0 \\ p_4 & 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

W pracy [24] różne sieci z danymi powstają z różnych grup automorfizmów. Autorzy rozważali dwie podstawowe sieci z danymi. Oto odpowiadające im VAS-y z danymi.

Definicja 7 VAS z danymi bez porządku (*UDPN*) powstają, gdy \mathbf{Aut} to zbiór wszystkich bijekcji z $\mathbb{D} \rightarrow \mathbb{D}$.

Definicja 8 Aby zdefiniować VAS z uporządkowanymi danymi (*UDPN*), wyposażamy dziedzinę danych \mathbb{D} w gęsty porządek liniowy, najprostszym z nich jest zbiór liczb wymiernych \mathbb{Q} . Jako grupę automorfizmów danych \mathbf{Aut} przyjmujemy wszystkie bijekcje zachowujące porządek z \mathbb{D} do \mathbb{D} .

W pracy [24] autorzy udowodnili, że problemy związane z terminacją i pokrywalnością są rozstrzygalne. Udowodnili tym samym, że sieci z danymi należą do ciekawej klasy z potencjałem do analizy symbolicznej. Nie jest trudno udowodnić, że problem osiągalności dla VAS-ów z uporządkowanymi danymi jest nierozstrzygalny, można to zrobić wykorzystując nierozstrzygalność osiągalności w sieciach Petriego z lukami resetującymi [37].

Dlatego pierwszy problem badawczy, który przyciągnął moją uwagę, to problem osiągalności dla VAS-ów z danymi bez porządku:

Problem badawczy 9 Mając dany VAS z danymi bez porządku \mathbb{T} oraz dwa wektory z danymi \mathbf{m}_0 i \mathbf{m} , pytamy czy \mathbf{m} należy do $\text{Reach}(\mathbf{m}_0)$?

4.1.5 Próby rozwiązania problemu osiągalności.

Naturalnym podejściem do rozwiązania problemu osiągalności dla VAS-ów z danymi bez porządku jest analiza algorytmu rozwiązywania problemu osiągalności dla Vas-ów bez danych [27] i zastanowienie się, czy podobne podejście może zadziałać. Dwa fundamentalne składniki algorytmu Kosaraju [27] to konstrukcja drzewa pokrywalności i równanie stanu. Oba te elementy można przenieść do VAS-ów z danymi bez porządku [4, 5]. Niemniej jednak te obiecujące wyniki są niewystarczające do rozszerzenia całego algorytmu Kosaraju na VAS-y z danymi bez porządku. Kluczowe trudności pojawiają się przy próbach udowodnienia terminacji algorytmu. Krytyczną własnością wykorzystywaną przez Kosaraju jest ograniczenie na ilość wymiarów rozważanego VAS-a. Niestety, w kontekście danych, każda dana generuje własny zestaw wymiarów, co prowadzi do nieskończonej liczby wymiarów. Niemniej jednak oba wyniki [4, 5] są interesujące, mają potencjalne zastosowania i zostały docenione przez recenzentów.

4.1.6 Drzewo pokrywalności [4]

Jedną z ważnych właściwości VAS-ów jest *monotoniczność*. Precyzyjnie, dla danych dwóch konfiguracji \mathbf{m}_0 i \mathbf{m}'_0 , takich że $\mathbf{m}_0 \sqsubseteq \mathbf{m}'_0$, jeśli istnieje krok $\mathbf{m}_0 \xrightarrow{t} \mathbf{m}_1$, to istnieje także krok $\mathbf{m}'_0 \xrightarrow{t} \mathbf{m}'_1$, i $\mathbf{m}_1 \sqsubseteq \mathbf{m}'_1$. Czyli dowolne zachowanie, które można obserwować z konfiguracji mniejszej, może zostać zaobserwowane z konfiguracji większej. Naturalnym pytaniem jest obliczenie/opisanie domknięcia w dół zbioru konfiguracji osiągalnych, tj. $Reach(\mathbf{m}) \downarrow$. Klasyczny algorytm drzewa pokrywalności pozwala obliczyć domknięcie w dół zbioru konfiguracji osiągalnych. Jednym z jego konsekwencji jest możliwość rozstrzygnięcia problemu ograniczoności zbioru konfiguracji osiągalnych i ograniczoności miejsca.

W VAS-ach z danymi bez porządku, porządek \sqsubseteq jest naturalnie uogólniany. $\mathbf{m} \sqsubseteq_{\mathbb{D}} \mathbf{m}'$, jeśli dla każdej danej $\xi \in \mathbb{D}$ mamy $\mathbf{m}(\xi) \sqsubseteq \mathbf{m}'(\xi)$. Autorzy pracy [26] zauważyli, że pojęcie ograniczoności miejsca można doprecyzować na dwa sposoby:

- Istnieje dana ξ i miejsce p , takie że dla każdego $n \in \mathbb{N}$, istnieje konfiguracja $\mathbf{m} \in Reach(\mathbf{m}_0)$, dla której $\mathbf{m}(\xi)(p) > n$ (brak ograniczenia na głębokości).
- Istnieje miejsce p , takie że dla każdego $n \in \mathbb{N}$, istnieje konfiguracja \mathbf{m}_n oraz różne dane $\{\xi_1, \xi_2, \dots, \xi_n\} \subseteq \mathbb{D}$, takie że dla każdej $\xi \in \{\xi_1, \xi_2, \dots, \xi_n\}$, $\mathbf{m}_n(\xi)(p) > 0$ (brak ograniczenia szerokości miejsca).

Pokazali również, że pierwszy z tych dwóch problemów (brak ograniczenia głębokości) jest nierozstrzygalny dla VAS-ów z uporządkowanymi danymi. Ten wynik opiera się na nierozstrzygalności problemu braku ograniczenia dla sieci Petriego z łukami resetującymi [37], oraz wspomnianej już zależności między VAS-ami z uporządkowanymi danymi a sieciami Petriego z łukami resetującymi.

W pracy [4] wraz z Sławomirem Lasotą, Ranko Lazicem, Jérôme Leroux, Sylvain Schmitz i Patrickiem Totzke uogólniliśmy konstrukcję algorytmu drzewa pokrywalności na VAS-y z danymi bez porządku. Nierozstrzygalność udowodniona przez autorów pracy [26] pokazuje, że nasza konstrukcja nie może być prostą adaptacją klasycznego algorytmu, ponieważ musimy uwzględnić różnice między dwiema grupami automorfizmów, wszystkich bijekcji w $\mathbb{D} \rightarrow \mathbb{D}$ oraz wszystkich bijekcji monotonicznych w $\mathbb{D} \rightarrow \mathbb{D}$. Główne wyniki opisane w pracy [4] są następujące:

Twierdzenie 10 (Twierdzenie 2.5. w [4]) *W VAS-ach z danymi bez porządku ograniczenie na głębokość miejsca implikuje ograniczenie szerokości miejsca.*

Twierdzenie 11 (Twierdzenie 2.6. w [4]) *Problem braku ograniczenia głębokości i problem braku ograniczenia szerokości miejsca dla VAS-ów z danymi bez porządku są w klasie F_{ω} , czyli w klasie złożoności Hiper-Ackermannowskiej.*

Problemy związane z ograniczonością są ważne. Jeśli możemy udowodnić, że dany system jest ograniczony, to przynajmniej w teorii możemy skonstruować cały graf osiągalnych konfiguracji i sprawdzić jego właściwości. W przypadku ograniczenia szerokości miejsc, możemy uprościć VAS z danymi. Ponieważ ilość danych, które mogą pojawić się w osiągalnych konfiguracjach jest ograniczona, to możemy je reprezentować eksplicite (z dokładnością do automorfizmu danych) jako oddzielne miejsca. W tej sytuacji możemy zamienić VAS z danymi na normalny VAS o większej liczbie miejsc, a co za tym idzie używać algorytmów i technik opracowanych do analizy VAS-ów.

4.1.7 Równanie stanów [5]

Aby wprowadzić *równanie stanu* dla VAS-ów, przeprowadzamy następujący rozumowanie. Załóżmy, że w pewnym VAS-sie istnieje sekwencja kroków

$$\mathbf{m}_0 \xrightarrow{\mathbf{t}_1} \mathbf{m}_2 \xrightarrow{\mathbf{t}_2} \mathbf{m}_3 \xrightarrow{\mathbf{t}_3} \dots \xrightarrow{\mathbf{t}_n} \mathbf{m}_n.$$

Wówczas zachodzi następujące równanie:

$$\mathbf{m}_0 + \mathbf{t}_1 + \mathbf{t}_2 + \dots + \mathbf{t}_n = \mathbf{m}_n.$$

W rezultacie możemy sformułować następujące lemat, który jest dobrze znany w społeczności zajmującej się sieciami Petriego:

Lemat 12 *Jeśli \mathbf{m}_n jest osiągalna z \mathbf{m}_0 , to $\mathbf{m}_n - \mathbf{m}_0$ można przedstawić jako sumę wektorów ze zbioru \mathbb{T} .*

Zazwyczaj wyrażamy to jako

$$\mathbf{m}_n - \mathbf{m}_0 = \Delta \mathbf{x}, \quad (2)$$

gdzie Δ to macierz, jej kolumny to wektory z \mathbb{T} , a my szukamy rozwiązania $\mathbf{x} \in \mathbb{N}^{|\mathbb{T}|}$.

Zauważmy, że tzw. *relacja osiągalności całkowito liczbowej*, zdefiniowana jako $R^{Int}(\mathbf{m}_0, \mathbf{m}_1) = \{(\mathbf{m}_0, \mathbf{m}_1) \in \mathbb{Z}^k \times \mathbb{Z}^k \mid \mathbf{m}_1 - \mathbf{m}_0 = \Delta \mathbf{x} \text{ dla pewnego } \mathbf{x} \in \mathbb{N}^{|\mathbb{T}|}\}$, jest nadaproksymacją relacji osiągalności, tj. $\mathbf{m}_1 \in Reach(\mathbf{m}_0) \implies R^{Int}(\mathbf{m}_0, \mathbf{m}_1)$.

Możemy bezpośrednio uogólnić to równanie na VAS-y z danymi, po prostu mówiąc, że różnica $\mathbf{m}_n - \mathbf{m}_0$, gdzie \mathbf{m}_0 i \mathbf{m}_n to wektory z danymi, może być przedstawiona jako suma wektorów z danymi ze zbioru \mathbb{T} . Aby uzyskać reprezentację macierzową, powinniśmy utożsamiać wektory z danymi postaci $\mathbb{D} \rightarrow \mathbb{Z}^k$ z funkcjami $\mathbb{D} \times \{1, \dots, k\} \rightarrow \mathbb{Z}$. W reprezentacji macierzowej, Δ ma nieskończoną liczbę kolumn i wierszy, a \mathbf{x} jest ograniczone do wektorów w $\mathbb{N}^{|\mathbb{T}|}$, które mają tylko skończoną liczbę niezerowych koordynat (ponieważ szukamy skończonych biegów).

W artykule [5] badałem ten problem razem z Jérômeem Leroux i Patrickiem Totzke. Udowodniliśmy następujące twierdzenie:

Twierdzenie 13 (Wniosek 8, [5]) *Problem wyrażalności dla wektorów z danymi bez porządku o wartościach w $(\mathbb{Z}^k, +)$ jest NP-zupełny.*

Algorytm rozwiązywania równania stanu dla VAS-ów z danymi bez porządku opiera się na koncepcji histogramów, która jest jednym z głównych wkładów artykułu. Aby wprowadzić histogramy, rozpoczniemy od przykładu. Rozkładamy krok \mathbf{t} z równania 1 jako iloczyn dwóch macierzy: macierz przejścia oznaczoną jako $m(\mathbf{t})$, która zależy tylko od orbity przejścia (orbity wektora z danymi), oraz macierz instancjacji $h(\mathbf{t})$, która wiąże wartości danych ze zmiennymi używanymi do reprezentacji przejścia w sposób symboliczny. Druga macierz nazywana jest *prostym histogramem*.

$$\begin{bmatrix} & \alpha & \beta & \gamma & \delta & \sigma & \dots \\ p_1 & -1 & -1 & 0 & 0 & 0 & 0 \\ p_2 & 0 & 0 & -2 & 0 & 0 & 0 \\ p_3 & 0 & 0 & 1 & 0 & 0 & 0 \\ p_4 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} & y & z & x & u \\ p_1 & -1 & -1 & 0 & 0 \\ p_2 & 0 & 0 & -2 & 0 \\ p_3 & 0 & 0 & 1 & 0 \\ p_4 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} & \alpha & \beta & \gamma & \delta & \sigma & \dots \\ x & 1 & 0 & 0 & 0 & 0 & 0 \\ y & 0 & 1 & 0 & 0 & 0 & 0 \\ z & 0 & 0 & 1 & 0 & 0 & 0 \\ u & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (3)$$

Niech $\mathbf{t}_i = \mathbf{t} \circ \pi_i$ dla pewnych $\pi_i \in \mathbf{Aut}$ oraz $i \in 1, \dots, n$. Ze względu na liniowość, suma wektorów z danymi z tej samej orbity może być zapisana następująco: $\sum_{i \leq n} \mathbf{t}_i = m(\mathbf{t}) \cdot (\sum_{i \leq n} h(\mathbf{t}_i))$. Rozważmy macierz $(\sum_{i \leq n} h(\mathbf{t}_i))$. Posiada ona dwie proste właściwości: suma elementów w każdym wierszu wynosi n , a suma elementów w każdej kolumnie jest nie większa niż n . Teraz jesteśmy gotowi do zdefiniowania histogramów.

Definicja 14 (Definicja 2 w [5]) *Histogram nad skończonym zbiorem \mathbb{S} zmiennych to funkcja całkowitoliczbowa $H : \mathbb{S} \times \mathbb{D} \rightarrow \mathbb{N}$ taka, że dla pewnego $n \in \mathbb{N}$, nazywanego stopniem H , zachodzą następujące dwa warunki:*

1. $\sum_{\beta \in \mathbb{D}} H(x, \beta) = n$ dla dowolnego $x \in \mathbb{S}$
2. $\sum_{x \in \mathbb{S}} H(x, \beta) \leq n$ dla dowolnego $\beta \in \mathbb{D}$.

Zachodzi następujący twierdzenie.

Twierdzenie 15 (Twierdzenie 3 w [5]) *Funkcja $H : \mathbb{S} \times \mathbb{D} \rightarrow \mathbb{N}$ jest histogramem stopnia $n \in \mathbb{N}$ wtedy, i tylko wtedy, gdy H jest sumą n prostych histogramów nad \mathbb{S} .*

Dowód tego twierdzenia opiera się na Twierdzeniu o Małżeństwach Halla, tj. że istnieje doskonałe skojarzenie w grafie dwudzielnym (X, Y, E) wtedy i tylko wtedy, gdy dla każdego podzbioru $X' \subseteq X$, moc jego sąsiedztwa jest nie mniejsza niż $|X'|$.

Korzystając z histogramów, udowodniliśmy, że jeśli istnieje rozwiązanie równania stanu, to istnieje takie, które używa małej liczby danych. Wreszcie, aby rozwiązać równanie, trzeba zgadnąć histogramy (które używają małej liczby danych), po jednym dla każdej orbity tranzycji w \mathbb{T} , i sprawdzić, czy zgadnięte histogramy, gdy pomnożone przez odpowiadające im macierze przejść, sumują się do celu (lewa strona równania 2).

Podsumowując, w [5] rozwiązaliśmy równanie stanu dla VAS-ów z danymi bez porządku, wprowadzając skompresowaną formę rozwiązania o nazwie histogram i pokazując, że histogram można przekształcić w rozwiązanie, jeśli spełnia pewne właściwości, które można sprawdzić w czasie wielomianowym.

4.1.8 Dalsze kierunki badań

Po wielu nieudanych próbach rozwiązania problemu osiągalności dla VAS z danymi, w ramach zespołu, w składzie którego byli Sławomir Lasota, Ranko Lazic, Jérôme Leroux, Radosław Piórkowski i ja, zaproponowałem inny kierunek badań.

Motywacją było to, że równanie stanu i inne techniki oparte na algebrze liniowej stanowią rdzeń wielu wyników dotyczących podklas sieci Petriego, a także są skutecznie wykorzystywane do poprawy wydajności narzędzi. Na przykład, rozważmy prostą metodę poprawy algorytmu naiwnego służącego do sprawdzenia osiągalności. Naiwny algorytm przeszukuje graf osiągalnych konfiguracji. Jeśli w pewnym momencie znajdzie docelową konfigurację \mathbf{m}_1 , ogłasza sukces. Jeśli przejdzie przez cały graf, kończy się niepowodzeniem, a jeśli zabraknie mu pamięci lub czasu, zatrzymuje się bez jednoznacznej odpowiedzi. Możemy go udoskonalić na dwa sposoby.

- Pierwszy z nich polega na tym, że dla każdej osiągniętej konfiguracji \mathbf{m} , algorytm sprawdza, czy istnieje rozwiązanie równania stanu $\mathbf{m}_1 - \mathbf{m} = \Delta \mathbf{x}$. W przypadku negatywnej odpowiedzi przerywa eksplorację tej części grafu, do której można dotrzeć z \mathbf{m} , ponieważ \mathbf{m}_1 nie może do niej należeć. Należy zauważyć, że dowolna nadaproxymacja relacji osiągalności może być użyta do przycięcia przestrzeni poszukiwań algorytmu naiwnego w podobny sposób.
- Drugi sposób opiera się na użyciu algorytmu A^* z heurystyką wybierania ścieżki do rozszerzenia na podstawie normy \mathbf{x} (minimalizuje normę).

Te dwa podejścia nie byłyby wykonalne bez przełomu w SAT i SMT solverach, które są wykorzystywane do rozwiązywania równania stanów. Powyższe proste obserwacje wywołały ciąg badań nad równaniami stanu dla VAS-ów z danymi, które odpowiadają różnym rodzajom sieci z danymi.

4.1.9 Równanie stanów dla sieci z uporządkowanymi danymi [6]

Biorąc pod uwagę powyższe zaproponowałem zbadanie równanie stanów dla VAS-ów z uporządkowanymi danymi. Miałem nadzieję na szybkie zastosowania rozwijanych teorii, ze względu na wcześniejsze prace innych badaczy:

- W 2015 roku Michael Blondin, Alain Finkel, Christoph Haase i Serge Haddad zaprezentowali bardzo proste narzędzie do sprawdzania pokrywalności w sieciach Petriego [39]. Narzędzie to wykorzystuje ciągłą osiągalność (technikę podobną do równania stanów) jako przybliżenie relacji

osiągalności w celu przycięcia przestrzeni poszukiwań w algorytmie pokrywalności wstecz. Pomimo swojej prostoty narzędzie, które stworzyli, dorównywało wiodącym narzędziom do sprawdzania pokrywalności.

- W tym samym roku Thomas Geffroy, Jérôme Leroux i Grégoire Sutre przedstawili podobną co do wydajności implementację algorytmu pokrywalności wstecz z przycinaniem opartym na równaniu stanów, a nie na osiągalności ciągłej [40].
- W 2001 roku Parosh Aziz Abdulla i Aletta Nylén wprowadzili sieci Petriego z zegarami [41], w których każdy token ma stempel czasowy reprezentujący "wiek" tokenu. Ten model, chociaż niezby popularny, uogólnia wiele innych rodzajów sieci Petriego z czasem rozważanych w literaturze. Co ciekawe, w pracy [42] Rémi Bonnet, Alain Finkel, Serge Haddad i Fernando Rosa-Velardo pokazali, że pokrywalność w sieciach Petriego z zegarami wprowadzonych przez Parosha Aziza Abdullę można zredukować do pokrywalności w VAS-ach z uporządkowanymi danymi.

Ponieważ algorytm wstecznej pokrywalności działa dla VAS-ów z uporządkowanymi danymi [24], naturalnym pomysłem jest wykorzystanie równania stanu dla VAS-ów z uporządkowanymi danymi do budowy narzędzia do sprawdzania pokrywalności w sieciach Petriego z zegarami. Te ostatnie są formalizmem wystarczająco ekspresywnym, aby znaleźć zastosowanie w przemyśle.

Wspólnie z Sławomirem Lasotą opublikowaliśmy wyniki naszych badań w pracy [6]. Udowodniliśmy, że równanie stanu można rozwiązać dla VAS-ów z uporządkowanymi danymi, niestety okazało się jednocześnie, że jest to problem Ackermann trudny.

Twierdzenie 16 (Twierdzenie 1 w [6]) *Problem osiągalności w VAS można w czasie wielomianowym zredukować do problemu równania stanów dla VAS-a z uporządkowanymi danymi. W przeciwnym kierunku redukcja wymaga czasu wykładniczego.*

Algorytm opiera się na histogramach, ale na histogramach dostosowanych do uporządkowanej dziedziny danych. Konkretnie udowodniliśmy, że jeśli istnieje rozwiązanie, to istnieje inne rozwiązanie, które można przedstawić za pomocą histogramów w specjalnej postaci. Istnienie takich histogramów zostało następnie zredukowane do problemu osiągalności w starannie zaprojektowanym VAS-ie.

Aby udowodnić trudność problemu, zakodowaliśmy w nim osiągalność w VAS-ach. Użyliśmy tranzycji, które produkują dane z większymi wartościami niż wartości danych, które są konsumowane. Zatem można traktować dane jako znaczniki czasowe i w ten sposób zakodować, w sumie wektorów z uporządkowanymi danymi, kolejność wykonywania przejść w trakcie działania VAS-a.

Twierdzenie 16 pokazuje, że równanie stanu dla VAS-ów z uporządkowanymi danymi, nie będzie przydatne jako szybka heurystyka do analizy sieci Petriego z zegarami. Niespodziewanie złożoność spada do czasu wielomianowego, jeśli rozwiązujemy równanie stanów nad dodatnimi liczbami wymiernymi $\mathbb{Q}_{\geq 0}$, a nie nad liczbami naturalnymi. Wynik ten jest uzyskiwany w podobny sposób jak Twierdzenie 16, ale aby znaleźć histogramy, należy użyć ciągłej osiągalności w VAS, która jest w czasie wielomianowym [18, 43] (zamiast złożoności Ackermanowskiej [34, 33] dla normalnej osiągalności). To zainspirowało mnie do podjęcia prac nad ciągłą osiągalnością dla VAS-ów z danymi: najpierw nad ciągłą osiągalnością dla VAS-ów z danymi bez porządku, a następnie dla VAS-ów z uporządkowanymi danymi.

4.1.10 Ciągła Osiągalność dla VAS-ów z Danymi[7]

Definicja 17 *W semantyce ciągłej dla VAS-ów, zbiór konfiguracji jest rozszerzony do funkcji ze zbioru miejsc w nieujemne liczby wymierne $\mathbb{Q}_{\geq 0}$. Mówimy także, że istnieje krok z konfiguracji \mathbf{m} do \mathbf{m}' , jeśli istnieje nieujemna liczba wymierna $a \in \mathbb{Q}_{\geq 0}$ i przejście $\mathbf{t} \in \mathbb{T}$ takie, że $\mathbf{m}' = \mathbf{m} + a \cdot \mathbf{t}$.*

Ciągła osiągalność jest nadaproxymacją relacji osiągalności. W [18], autorzy zaprezentowali pełną charakteryzację relacji ciągłej osiągalności w czasie wielomianowym. Jednym z kluczowych elementów tej charakteryzacji jest następujące twierdzenie, pokazujące związek między ciągłą osiągalnością a równaniem stanów.

Lemat 18 (Lemat 12, [18]) Niech $(\mathbb{T}, \mathbf{m}_0)$ będzie siecią Petriego z początkowym markowaniem \mathbf{m}_0 oraz niech $\mathbf{v} \in \mathbb{Q}_{\geq 0}^{\mathbb{T}}$, taki, że zachodzą:

- $\mathbf{m} = \mathbf{m}_0 + \Delta \mathbf{v}$;
- $\forall p \in \bullet \mathbf{v}$, zachodzi $\mathbf{m}_0(p) > 0$;
- $\forall p \in \mathbf{v}^\bullet$, zachodzi $\mathbf{m}(p) > 0$.

Wówczas istnieje bieg z \mathbf{m}_0 do \mathbf{m} w semantyce ciągłej.

Na początku 2018 roku razem z Aksheyem S. z IIT Bombay, i jego dwoma studentami Utkarsh-em Gupta i Preey-em Shah-em rozpocząłem prace nad uogólnieniem ciągłej osiągalności na VAS-y z danymi bez porządku. Wyniki, w tym główne twierdzenie przedstawione poniżej, zostały opublikowane na konferencji FoSSaCS 2019 [7].

Twierdzenie 19 (Twierdzenie 1 w [7]) Jeśli konfiguracja \mathbf{m}' jest osiągalna w sposób ciągły z konfiguracji początkowej \mathbf{m} , to istnieje bieg $\mathbf{m} \xrightarrow{c} \mathbf{m}'$ w semantyce ciągłej, taki że zbiór użytych danych w trakcie biegu jest ograniczony przez funkcję liniową od wielkości wejścia.

Korzystając z powyższego ograniczenia, otrzymujemy algorytm o wielomianowej złożoności dla problemu ciągłej osiągalności w VAS-ach z danymi bez porządku.

Twierdzenie 20 (Twierdzenie 3 w [7]) Niech \mathbb{T} to VAS z danymi bez porządku, \mathbf{m}, \mathbf{m}' to dwie konfiguracje. Rozstrzygnięcie, czy \mathbf{m}' jest osiągalna w semantyce ciągłej z \mathbf{m} , jest problemem w klasie PTIME.

Aby udowodnić Twierdzenie 19, bierzemy pewien bieg i pokazujemy, że istnieje wtedy też inny bieg, w którym nieformalnie mówiąc transycje występują w sposób bardziej jednorodny⁴. Dla takiego biegu, który ma jednorodną strukturę udało się pokazać, że jeśli wykorzystuje on wiele różnych danych, które nie występują ani w konfiguracji początkowej, ani w końcowej, to istnieje inny bieg jednorodny, który używa mniejszej liczby danych. Korzystając z tego, stopniowo zmniejszamy liczbę danych występujących w biegu.

Aby udowodnić Twierdzenie 20, dostosowujemy techniki używane dla ciągłej osiągalności w sieciach Petriego bez danych z [18, 44] do przypadku z danymi i uzyskujemy charakteryzację ciągłej osiągalności dla VAS-ów z danymi bez porządku. Następnie pokazujemy, jak charakteryzacja ta może być połączona z powyższym ograniczeniem na ilość danych oraz technikami opartymi o histogramy [5], aby uzyskać wielomianowej wielkości system równań liniowych z dodatkowymi warunkami, postaci jeśli $x > 0$ to $y > 0$. Następnie wystarczy rozwiązać stworzony układ równań.

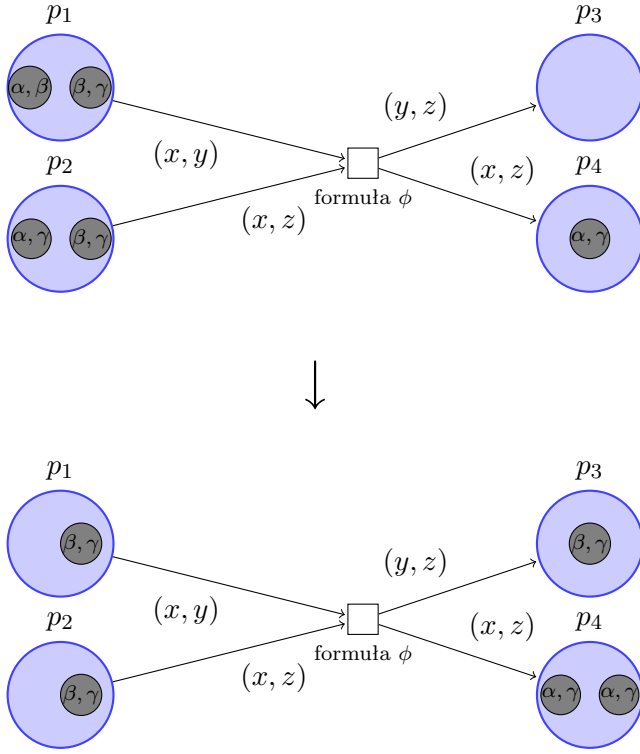
Problem ciągłej osiągalności w dziedzinie danych uporządkowanych jest znacznie bardziej wymagający i pozostaje nierozwiązany. Główna trudność w tym przypadku wynika z relatywnie niskiego poziomu symetrii występujących w dziedzinie danych uporządkowanych, co sprawia, że trudno jest zmniejszyć liczbę używanych danych w trakcie biegu w sposób podobny do przedstawionego w Twierdzeniu 19.

4.1.11 Tokeny niosące wiele danych [8]

W 2016 roku Mikołaj Bojańczyk zapytał mnie, czy możliwe jest rozwiązanie równania stanów w przypadku, gdy tokeny mogą przenosić krotki danych. Najpierw wytłumaczę jak dopuszczenie tokenów, które mogą przenosić wiele danych, zmienia definicję VAS-ów z danymi. Następnie opiszę wyniki i pytania, które pojawiły się w trakcie badań.

Konstrukcja sieci z danymi jest prosta, patrz na Rysunek 3.

⁴Patrz dowód Twierdzenia 19 w [7].



Rysunek 3:

Na lewo widzimy odpalenie tranzycji w sieci z danymi, w której każdy token przenosi parę danych. Załóżmy, że formuła określa, że każde dwie zmienne są różne. Przypisujemy $x = \alpha$, $y = \beta$, $z = \gamma$ a następnie usuwamy i produkujemy tokeny zgodnie z etykietowanymi krawędziami. W tym modelu wiadomo, że zarówno problem osiągalności, jak i problem pokrywalności są nierozstrzygalne, nawet dla dziedziny danych bez porządku [45]. Niemniej jednak, w praktyce rozwój narzędzi do analizy takich sieci jest potrzebny. Na przykład, w środowisku zajmującym się modelowaniem procesów biznesowych, analiza przepływów pracy z danymi stanowi jeden z badanych obszarów [46].

Aby zdefiniować odpowiadający VAS z danymi, musimy zacząć od rozszerzenia definicji wektora z danymi. W kolejnych sekcjach będziemy zakładać, że **Aut** to zbiór wszystkich bijekcji w $\mathbb{D} \rightarrow \mathbb{D}$, to znaczy wszystkie dalsze wyniki dotyczą danych bez porządku.

Definicja 21 (Wektor z Danymi uogólnienie) *Załóżmy, że \mathbb{D} to dziedzina danych, a **Aut** to zbiór jej automorfizmów. Niech \mathbb{B} to zbiór rozkładający się na skończenie wiele orbit działania grupy **Aut**. Wektor danych nad \mathbb{B} jest definiowany jako dowolna funkcja \mathbf{v} z \mathbb{B} do \mathbb{Z} , która przypisuje 0 wszystkim, oprócz skończonej liczby, elementom zbioru \mathbb{B} .*

Przykład 22 *Załóżmy, że zbiór \mathbb{B} jest zbiorem wszystkich par danych. Ma on dwie orbity wygenerowane przez pary (α, α) i (α, β) , zakładając, że $\pi((x, y)) = (\pi(x), \pi(y))$ dla $\pi \in \mathbf{Aut}$.*

Uwaga 23 *Nie jest to najbardziej ogólna postać wektorów z danymi, które rozważamy. Rozważane są też wektory z danymi, które mają skończone wsparcie⁵. Wektor danych ma skończone wsparcie, jeśli jest zbiorem par (element dziedziny, wartość) o skończonym wsparciu. Zbiór \mathbb{S} ma skończone wsparcie, jeśli istnieje skończony podzbiór $\mathbb{X} \subseteq \mathbb{D}$, taki że \mathbb{S} jest niezmienniczy ze względu na działania podgrupy $\mathbf{Aut}_{\mathbb{X}} \subset \mathbf{Aut}$, gdzie $\mathbf{Aut}_{\mathbb{X}}$ to podgrupa **Aut** złożona z elementów będących identycznością na zbiorze \mathbb{X} (stabilizator \mathbb{X}). Na przykład wektor z danymi $\mathbf{v} : \mathbb{D} \rightarrow \mathbb{N}$ taki, że $\mathbf{v}(\alpha) = 1$, $\mathbf{v}(\beta) = 2$, $\mathbf{v}(\xi) = 3$ dla każdego $\xi \in \mathbb{D} \setminus \{\alpha, \beta\}$ ma skończone wsparcie $\mathbb{X} = \{\alpha, \beta\}$.*

Aż do Sekcji 4.1.13 wystarczy jednak zrozumienie jedynie definicji 21.

VAS z danymi, jak wcześniej, jest zbiorem wektorów z danymi zamkniętym na działanie grupy **Aut** i mającym skończoną liczbę orbit. Indukuje on system tranzycyjny, gdzie konfiguracjami są wektory z danymi z $\mathbb{B} \rightarrow \mathbb{N}$.

W 2017 roku opracowałem algorytm dla przypadku, gdy \mathbb{B} jest zbiorem dwuelementowych zbiorów danych. W tym przypadku możemy zidentyfikować dane z wierzchołkami, a pary danych z krawędziami grafu nieskierowanego. Dlatego wektor z danymi \mathbf{v} można zidentyfikować z grafem z krawędziami etykietowanymi liczbami całkowitymi, gdzie każda krawędź (α, β) ma wagę $\mathbf{v}(\alpha, \beta)$.

⁵Tutaj występuje konflikt w nazewnictwie angielskim. Pojęcie skończonego supportu zostało wprowadzone do teorii zbiorów z atomami wiele lat temu, kiedy nikt nie rozważał jeszcze funkcji w ramach tej teorii.

Przykład 24 (Przykład 1.2 w [8]) Rozważmy wektory z danymi postaci $\mathbb{B} \rightarrow \mathbb{Z}$, gdzie $\mathbb{B} = \{\{\xi, \xi'\} \mid \xi, \xi' \in \mathbb{D}, \xi \neq \xi'\}$. Możemy je postrzegać jako grafy wagone, z wierzchołkami oznaczonymi elementami z \mathbb{D} . Załóżmy, że zbiór \mathbb{T} jest zbiorem trójkątów, gdzie wagi wszystkich krawędzi są równe 1, tzn.

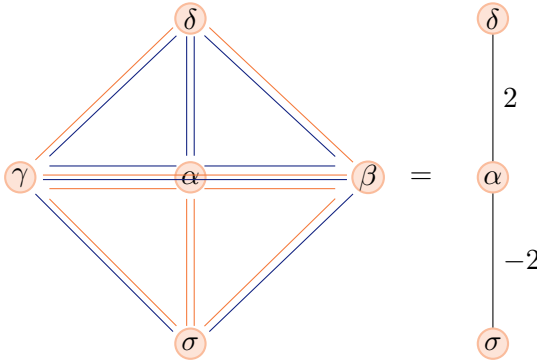
$$\mathbf{a}_{\xi\xi'\xi''}(x) = \begin{cases} 1 & \text{jeśli } x \in \{\{\xi, \xi'\}, \{\xi', \xi''\}, \{\xi, \xi''\}\} \\ 0 & \text{w przeciwnym razie} \end{cases}$$

i $\mathbb{T} = \{\mathbf{a}_{\xi\xi'\xi''} \in \mathbb{B} \rightarrow \mathbb{Z} \mid \xi, \xi', \xi'' \in \mathbb{D}, \xi \neq \xi' \neq \xi'' \neq \xi\}$. Zbiór \mathbb{T} jest zamknięty na automorfizmy w \mathbf{Aut} . Rzeczywiście, dla dowolnego $\pi \in \mathbf{Aut}$ i dowolnego $\mathbf{a}_{\xi\xi'\xi''} \in \mathbb{T}$, wektor z danymi $\mathbf{a}_{\pi(\xi)\pi(\xi')\pi(\xi'')} \in \mathbb{T}$.

Na koniec chcemy dowiedzieć się, czy poniższy wektor $\mathbf{a}_{\gamma\delta}$ może być wyrażony jako skończona suma wektorów z danymi z $\mathbb{T} \cup (-1) \cdot \mathbb{T}$.

$$\mathbf{a}_{\gamma\delta}(x) = \begin{cases} 6 & \text{if } x = \{\gamma, \delta\} \\ 0 & \text{w przeciwnym razie} \end{cases}$$

($\mathbf{a}_{\gamma\delta}$ to pojedyncza krawędź o wadze 6). Odpowiedź jest twierdząca, ale jest to nietrywialna suma. $\mathbf{a}_{\gamma\delta} = (\mathbf{a}_{\gamma\delta\sigma} + \mathbf{a}_{\gamma\delta\phi} + \mathbf{a}_{\sigma\phi\delta} - \mathbf{a}_{\gamma\sigma\phi}) + (\mathbf{a}_{\gamma,\delta,\rho} + \mathbf{a}_{\gamma\delta\omega} + \mathbf{a}_{\rho\omega\delta} - \mathbf{a}_{\gamma\rho\omega}) + (\mathbf{a}_{\gamma\delta\lambda} + \mathbf{a}_{\gamma\delta\kappa} + \mathbf{a}_{\lambda\kappa\delta} - \mathbf{a}_{\gamma\lambda\kappa}) - (\mathbf{a}_{\rho\delta\sigma} + \mathbf{a}_{\rho\delta\phi} + \mathbf{a}_{\sigma\phi\delta} - \mathbf{a}_{\rho\sigma\phi}) - (\mathbf{a}_{\omega\delta\lambda} + \mathbf{a}_{\omega\delta\kappa} + \mathbf{a}_{\lambda\kappa\delta} - \mathbf{a}_{\omega\lambda\kappa})$. Idea stojąca za tą sumą jest następująca:



Najpierw konstruujemy gadżet $\mathbf{g}_{\delta\alpha\sigma}$, jak przedstawiono po lewej stronie. Trójkąty o wagach 1 są kolorowane na niebiesko, a trójkąty o wagach -1 są kolorowane na pomarańczowo. $\mathbf{g}_{\delta\alpha\sigma} = \mathbf{a}_{\alpha\beta\delta} - \mathbf{a}_{\alpha\beta\sigma} + \mathbf{a}_{\alpha\gamma\delta} - \mathbf{a}_{\alpha\gamma\sigma} + \mathbf{a}_{\gamma\beta\sigma} - \mathbf{a}_{\gamma\beta\delta}$. Następnie, używamy takich gadżetów razem z trójkątami w następujący sposób: $\mathbf{g}_{\gamma\delta\alpha} + \mathbf{g}_{\delta\gamma\alpha} + 2\mathbf{a}_{\alpha\gamma\delta} = \mathbf{a}_{\gamma\delta}$.

Powyższy przykład ukazuje złożoność problemu. Zaprosiłem studenta, Jakuba Różyckiego, do udziału w projekcie. Razem zajęliśmy się przypadkiem, gdy \mathbb{B} jest skończoną sumą orbit zbiorów o tej samej wielkości. Na przykład $\mathbb{B} = \{0, 1\} \times \{\{\alpha, \beta\} \mid \alpha, \beta \in \mathbb{D} \text{ and } \alpha \neq \beta\}$ jest sumą 2 orbit. Wektory o dziedzinie równej \mathbb{B} odpowiadają grafom z krawędziami etykietowanymi parami liczb.

Wyniki naszej pracy zostały opublikowane w pracy [8]. Przeformułowaliśmy problem z równania stanu na problem \mathbb{X} -wyrażalności, sformułowany następująco: mając dany wektor z danymi \mathbf{v} i rodzinę wektorów z danymi \mathbb{T} o skończeniu wielu orbitach, pytamy, czy \mathbf{v} można wyrazić jako ważoną sumę elementów z \mathbb{T} , gdzie wagi pochodzą ze zbioru \mathbb{X} . Tak więc równanie stanu jest uchwycone przez $\mathbb{X} = \mathbb{N}$. Udowodniliśmy dwa ważne twierdzenia.

Twierdzenie 25 (Twierdzenie 2.1 w [8]) Niech $k, l \in \mathbb{N}$ i $\mathbb{B} = \{1 \dots l\} \times (\text{podzbiory } \mathbb{D} \text{ o rozmiarze } k)$. Załóżmy, że $\mathbb{T} \subset \mathbb{B} \rightarrow \mathbb{Z}$ jest skończenie orbitowy. Wówczas problem \mathbb{Z} -wyrażalności jest w $P\text{TIME}$ (zależność od k jest wykładnicza).

Twierdzenie 26 (Twierdzenie 2.2 w [8]) Dla k, l, \mathbb{B} , i \mathbb{T} jak w Twierdzeniu 25, problem N -wyrażalności jest w $NEXPTIME$.

Pierwsze z tych dwóch twierdzeń wynika z kombinatorycznej charakteryzacji wektorów z danymi, które można wyrazić jako \mathbb{Z} -ważone sumy elementów zbioru \mathbb{T} . Zidentyfikowaliśmy zbiór własności, które są konieczne i wystarczające do charakteryzacji wyrażalności. Jeśli spojrzymy na wektory z danymi jako na grafy wagone, to jedna przykładowa z tych własności wygląda następująco: każda

etykieta krawędzi w docelowym grafie musi być wyrażalna jako \mathbb{Z} -ważona suma etykiet krawędzi w grafach z rodziny \mathbb{T} . Pozostałe własności mają podobny charakter.

Dowód drugiego wyniku opiera się na redukcji do problemu \mathbb{Z} -rozwiązywalności. Po pierwsze, zidentyfikowaliśmy wektory z danymi w \mathbb{T} , które są odwracalne, tzn. $\mathbf{t} \in \mathbb{T}$ jest odwracalny, jeśli $-\mathbf{v}$ jest wyrażalny jako \mathbb{N} -ważona suma elementów zbioru \mathbb{T} . Wektory z danymi, które nie są odwracalne, nazwaliśmy nieodwracalnymi. Udowodniliśmy, że odwracalność można sprawdzić w PTIME. Należy zauważyć, że jeśli każdy $\mathbf{t} \in \mathbb{T}$ jest odwracalny, to problem \mathbb{N} -rozwiązywalności jest taki sam jak problem \mathbb{Z} -rozwiązywalności. Następnie udowodniliśmy wykładnicze ograniczenie na liczbę wektorów z danymi, które nie są odwracalne i mogą pojawić się w rozwiązaniu. Na podstawie powyższego zaproponowaliśmy następujący algorytm:

1. Niedeterministycznie zgadujemy nieodwracalne elementy sumy \mathbf{i}
2. sprawdzamy, czy pozostała część docelowego wektora danych \mathbf{v} może być wyrażona jako \mathbb{Z} -ważona suma wektorów z danymi odwracalnych w zbiorze \mathbb{T} .

Warto zauważyć, że ta technika redukcji \mathbb{N} -rozwiązywalności do \mathbb{Z} -rozwiązywalności działa także dla bardziej skomplikowanych zbiorów \mathbb{B} (dziedzin wektorów danych).

4.1.12 Od π -rachunku do obrazów Parikha [9]

Wstępne wyniki dotyczące sieci z danymi, w których każdy token przenosi zestaw danych [8], wywołały pytania o zastosowania poza sieciami z danymi, w szczególności w algebrach procesów. Zajęliśmy się tym razem z doktorantem Mohnishem Pattathurajanem, który był pod opieką Sławomira Lasoty. Znalazł on pracę autorstwa Fernando Rosy-Velardo i Marii Martos-Salgado, w której przedstawiono odwzorowanie z π -rachunku do ν -sieci [47]. Klasa ν -sieci mieści się pomiędzy VAS-ami z uporządkowanymi danymi a VAS-ami z danymi bez porządku. Zastanawialiśmy się, czy \mathbb{N} -rozwiązywalność mogłaby zostać zastosowana do analizy π -rachunku.

Spółeczność skupiona wokół algebr procesów jest dobrze znana z koncentracji na równoważnościach, takich jak bisymulacja. Naszym głównym celem było zaproponowanie nowej nadaproksymacji relacji bisimilarności. Upraszczając problem i dalej go abstrahując, doszliśmy do następującego scenariusza: Rozważmy procesy składające się z stanu kontrolnego i wewnętrznych danych przechowywanych w rejestrach. Każdy proces może wyemitować wiadomość w postaci $(a, x) \in \Sigma \times \mathbb{D}$, gdzie Σ to skończony alfabet, a następnie proces przekształca się w skończony zbiór procesów o tej samej formie. Ten model może być widziany jako rozszerzona wersja BPP (Basic Parallel Processes) lub jako przemienne gramatyka bezkontekstowa rozszerzenia o dane. W [48] Hirshfeld udowodnił nierozstrzygalność równoważności śladów nawet dla BPP bez danych. Postanowiliśmy zatem zbadać równoważność języków Parikha w przemienne gramatykach bezkontekstowych z danymi.

Definicja 27 *Niech \mathbb{A} będzie zbiorem. Homomorfizm Parikha Par to odwzorowanie z wolnego monoidu nad \mathbb{A} (czyli słów nad alfabetem \mathbb{A}) do wolnego monoidu przemiennego nad \mathbb{A} , takie że dla każdego $a \in \mathbb{A}$, $Par(a) = a$.*

Obraz Parikha słowa w nad alfabetem \mathbb{A} jest oznaczany jako $Par(w)$.

Językiem Parikha systemu jest obraz Parikha języka śladów systemu.

Aby sprawdzić równoważność obrazów Parikha między dwiema przemieniami gramatykami bezkontekstowymi z danymi, najpierw trzeba umieć scharakteryzować obraz Parikha przemiennej gramatyki bezkontekstowej z danymi. Innymi słowy, pierwszym krokiem powinno być uogólnienie twierdzenia Parikha z gramatyk bezkontekstowych na gramatyki bezkontekstowe z danymi. Twierdzenie Parikha mówi, że obrazy Parikha gramatyk bezkontekstowych są takie same jak obrazy Parikha automatów skończonych i że są one zbiorami semiliniowymi w $\mathbb{N}^{|\mathbb{A}|}$ [49].

Wspólnie z Mohnishem Pattathurajanem i Sławomirem Lasotą w pracy [9] udało nam się uogólnić twierdzenie Parikha do przypadku gramatyk bezkontekstowych z danymi, gdzie każdy nieterminal (proces) przenosi pojedynczą daną. Główne wyniki pracy są ujęte w dwóch poniższych twierdzeniach.

Twierdzenie 28 (Twierdzenie 6 w [9]) *Obrazy Parikha języków automatów rejestrowych z jednym rejestrem są zbiorami regularnymi⁶.*

Twierdzenie 29 (Twierdzenie 7 w [9]) *Obrazy Parikha języków przemiennych gramatyk bezkontekstowych z jednym rejestrem są zbiorami regularnymi.*

Dowód pierwszego twierdzenia przebiega podobnie do dowodu dla automatów skończonych. Najpierw zmieniamy etykiety krawędzi na wyrażenia regularne odpowiadające pojedynczym literom. Następnie stopniowo upraszczamy automat. Usuwamy stany jeden po drugim i zmieniamy etykiety krawędzi na bardziej skomplikowane wyrażenia, zachowując obraz Parikha języka. Trudną częścią jest radzenie sobie z pętłami. Wyrażenia regularne z danymi nie są w stanie w pełni uchwycić złożoności pętli w automatach rejestrowych. Kluczowe jest to, że pracujemy z obrazami Parikha języków, a nie językami. Dzięki temu możemy zmieniać kolejność liter w słowie, tak aby odpowiadający bieg po modyfikacji mógł być opisany wyrażeniem regularnym z danymi. Co ciekawe, przeorganizowanie bazuje między innymi na twierdzeniu dotyczącym warunków wystarczających dla istnienia ścieżek Hamiltona w grafie skierowanym [50] (w przybliżeniu: w dostatecznie gęstych grafach istnieją cykle Hamiltona).

Dowód twierdzenia 29 opiera się na przekształcaniu dowolnego drzewa wyprowadzenia, w drzewo o określonym kształcie, mianowicie drzewo cienkie, zachowując przy tym jego obraz Parikha. Intuicyjnie, drzewo cienkie to takie, które ma niewiele długich gałęzi. Następnie pokazane jest, że obrazy Parikha słów wyprowadzonych za pomocą drzew cienkich są sumami obrazów Parikha języków automatów rejestrowych. Procedura przekształcania dowolnego drzewa polega na operacjach wycinania i wklejania. Fragmenty drzewa wyprowadzenia, które można bezpiecznie wyciąć i wkleić, są identyfikowane za pomocą twierdzenia Ramseya. Problemem tutaj jest unikanie zerwania zależności pomiędzy danymi w rejestrach. Twierdzenie Parikha dla ogólnych automatów rejestrowych i gramatyk bezkontekstowych z danymi pozostaje nadal otwartym problemem. Również pytanie o równość dwóch zbiorów regularnych z danymi pozostaje otwarte.

Z powodu tego projektu zostałem promotorem pomocniczym Mohnisha Pattathurajana.

4.1.13 Równania z danymi [10, 11]

Kolejnym owocem mojej pracy z Jakubem Różyckim [8] są wyniki dotyczące elementów algebry liniowej z danymi. Pod ich wpływem, Sławomir Lasota zaproponował ten kierunek badawczy mnie oraz swojemu doktorantowi, Arce Ghosh. Poprosił mnie także o współ-opiekę nad Arką. Pierwszym celem było opracowanie algorytmu rozwiązywania skończonych układów równań i nierówności nad liczbami wymiernymi. Wyniki tej współpracy zostały opublikowane w dwóch pracach [10, 11].

Najpierw zauważmy, że nie jest jasne, co oznacza rozwiązanie nieskończonego układu równań. Możemy ograniczyć się tylko do rozwiązań skończonych, ale możemy też pytać o dowolne rozwiązanie lub rozwiązanie skończenie reprezentowalne. Ostatnie dwie możliwości muszą być doprecyzowane. Na przykład, czy korzystamy z jakiegoś pojęcia zbieżności lub do jakiego rodzaju skończonych reprezentacji się ograniczamy? Zdecydowaliśmy, że rozwiązanie nie powinno zależeć od kolejności sumowania, a przez skończoną reprezentację rozumiemy wektor z danymi o skończonym wsparciu (Uwaga 23), ponieważ ogólnie rozważamy obiekty skończenie orbitowe. Przyjrzyjmy się przykładowi, aby zobaczyć różnice:

Przykład 30 (Przykład 4.1 w [10]) *Niech kolumny będą indeksowane przez $\mathbb{C} \subset \mathbb{D}^2$, gdzie \mathbb{C} to zbiór par różnych elementów z \mathbb{D} , a wiersze przez \mathbb{B} , który jest zbiorem dwuelementowych podzbiorów \mathbb{D} . Rozważmy układ równań*

$$x_{\xi\xi'} + x_{\xi'\xi} = 1 \quad (\xi, \xi' \in \mathbb{D}, \xi \neq \xi').$$

W postaci macierzowej można go przedstawić jako

⁶To jest uogólnienie zbiorów semiliniowych.

$$\begin{array}{c}
\alpha\beta \quad \beta\alpha \quad \alpha\gamma \quad \gamma\alpha \quad \beta\gamma \quad \gamma\beta \quad \alpha\delta \quad \dots \\
\left\{ \begin{array}{l} \alpha, \beta \\ \alpha, \gamma \\ \beta, \gamma \\ \alpha, \delta \\ \vdots \end{array} \right\} \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \cdot \begin{bmatrix} x_{\alpha\beta} \\ x_{\beta\alpha} \\ x_{\alpha\gamma} \\ x_{\gamma\alpha} \\ x_{\beta\gamma} \\ \vdots \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ \vdots \end{bmatrix}.
\end{array}$$

Wszystkie równania są skończone. Wektor $\mathbf{x}(\xi\xi') = \frac{1}{2}$ dla każdego $\xi\xi' \in \mathbb{C}$ jest rozwiązaniem. Układ nie ma rozwiązania skończonego⁷, ponieważ iloczyn wektora skończonego przez tę macierz nie może dać nieskończonego wyniku $\mathbf{t} = \mathbf{1}_{\mathbb{B}}$. Ponadto, układ nie ma również rozwiązania całkowitoliczbowego o skończonym wsparciu, ponieważ jakiegokolwiek takie rozwiązanie \mathbf{x} musiałoby spełniać, dla pewnych różnych danych $\xi, \xi' \in \mathbb{D}$, równość $\mathbf{x}(\xi\xi') = \mathbf{x}(\xi'\xi)$, co jest sprzeczne z $\mathbf{x}(\xi\xi') + \mathbf{x}(\xi'\xi) = 1$. Wreszcie, ma on nieskończone-całkowitoliczbowe rozwiązanie, które nie ma skończonego wsparcia. Mianowicie, jeśli wprowadzimy liniowy porządek na zbiorze \mathbb{D} , to każdej zmiennej $x_{\xi\xi'}$, gdzie $\xi < \xi'$, przypisujemy wartość 1, a każdej zmiennej $x_{\xi\xi'}$, gdzie $\xi' < \xi$, przypisujemy wartość 0.

Skupiliśmy się na rozwiązaniach o skończonym wsparciu, ponieważ są one bardziej ogólne niż rozwiązania skończone. Zaczęliśmy od definicji zbiorów $Fin - lin(\mathbb{B})$ i $Lin(\mathbb{B})$. Pierwszy z nich to zbiór wszystkich wektorów danych, które są niezerowe dla skończonej liczby elementów \mathbb{B} , a wartości przyjmują w \mathbb{K} , gdzie \mathbb{K} to dowolny pierścień przemienny. Drugi to zbiór wszystkich wektorów o skończonym wsparciu, z \mathbb{B} do \mathbb{K} .

Baza przestrzeni wektorów z danymi \mathbb{S} to orbitowo skończony zbiór wektorów z danymi, taki że każdy wektor z danymi z \mathbb{S} można jednoznacznie rozłożyć na skończoną kombinację liniową wektorów z danymi z bazy. Oczywiście, według definicji, zbiór wektorów charakterystycznych dla elementów \mathbb{B} jest bazą $Fin - lin(\mathbb{B})$. Co bardziej interesujące prawdziwe jest następujące twierdzenie:

Twierdzenie 31 (Twierdzenie 3.3 (Twierdzenie o Bazie Orbitowo-Skończonej) w [10]) *Dla każdego zbioru orbitowo-skończonego \mathbb{B} , przestrzeń $Lin(\mathbb{B})$ ma bazę orbitowo-skończoną.*

To twierdzenie pozwala nam z pytania o rozwiązania o skończonym wsparciu przejść do pytania o rozwiązania skończone (o skończonym nośniku). Wystarczy zmienić bazę, w której zapisane jest rozwiązanie. Dzięki temu byliśmy w stanie udowodnić następujące twierdzenie:

Twierdzenie 32 (Twierdzenie 4.4. w [10]) *Problem $Solv(\mathbb{K})$ jest rozstrzygalny dla każdego ustalonego efektywnego pierścienia komutatywnego \mathbb{K} . Gdzie $Solv(\mathbb{K})$ to problem rozwiązywania układu równań z danymi.*

Algorytm opiera się na procedurze stopniowego upraszczania układu równań, kończąc na równoważnym skończonym układzie równań liniowych.

W drugim artykule [11] postanowiliśmy zrozumieć nierówności. Tutaj, zaskakująco, rozwiązywanie ich nad \mathbb{Z} jest nierozstrzygalne.

Twierdzenie 33 (Twierdzenie 16 w [11]) *Sprawdzanie, czy istnieje skończone rozwiązanie w \mathbb{Z} orbitowo-skończonego układu nierówności, jest nierozstrzygalne.*

Pokazaliśmy, że w systemie nierówności możemy zakodować działanie maszyny Mińskiego. Używamy par danych, aby stworzyć łańcuch $(\alpha, \beta), (\beta, \gamma)(\gamma, \delta), \dots (\sigma, \phi)$, którego używamy do ustalenia porządku, wokół którego budujemy kolejne konfiguracje i kroki wykonywane przez maszynę Mińskiego. Do stworzenia tego łańcucha używamy wzmocnionego warunku Kirchhoffa. Dla każdego wierzchołka (danej), suma wag krawędzi wchodzących jest równa sumie wag krawędzi wychodzących i jest mniejsza równa niż 1. Oczywiście musimy określić źródło i cel ścieżki. Do określenia konfiguracji używamy

⁷o skończonej liczbie niezerowych elementów.

zmiennych i równań indeksowanych trójkami danych. Nie wiemy, czy nierozstrzygalność zachodzi, jeśli ograniczymy liczbę danych występujących w opisie zmiennych do dwóch (liczba danych pojawiająca się w indeksie równania odpowiada liczbie danych przenoszonych przez pojedynczy token w sieci z danymi).

Ten wynik może być zaskakujący, ponieważ wcześniej wspomnieliśmy, że rozwiązanie systemu równań nad \mathbb{N} można zredukować do pytania o rozwiązanie systemu równań nad \mathbb{Z} . Kluczowa różnica polega tutaj na tym, że w poprzednim wyniku zakładaliśmy i wykorzystywaliśmy fakt, że docelowy wektor ma skończony nośnik (ma tylko skończoną liczbę niezerowych elementów) i że szukamy rozwiązań o skończonym nośniku. Drugim istotnym elementem jest to, że jeśli zamienimy nieskończony system nierówności nad \mathbb{Z} na system równań nad \mathbb{N} , to będziemy musieli szukać rozwiązań nieskończonych. Pomysł wykorzystania Twierdzenia 31 (Twierdzenia o istnieniu bazy) także nie działa. Musimy wtedy dodać nieskończoną liczbę nierówności gwarantujących, że rozwiązanie zapisane w bazie jest nieujemne (co zabiera nas znowu do pierwotnego problemu).

Twierdzenie 34 (Twierdzenie 6 w [11]) *Sprawdzenie, czy istnieje rozwiązanie skończone nad \mathbb{Q} dla orbitowo-skończonego systemu nierówności, jest rozstrzygalne.*

Tutaj pomysł jest podobny do tego z pracy dotyczącej osiągalności ciągłej [7], ale istotnie pogłębia koncepcję ujednoczenia rozwiązania. Wykorzystując tę technikę, zamieniamy nieskończony system nierówności na skończony system nierówności wielomianowych, który rozwiązujemy przy użyciu arytmetyki Tarskiego. Procedura zachowuje funkcję celu w programowaniu liniowym, dzięki czemu możemy rozwiązywać orbitowo-skończone programy liniowe.

4.1.14 Podumowanie

Wskazane rezultaty stanowią spójną całość. Jasne jest też, że bardzo wiele ciekawych pytań pozostaje otwartych, a tematyka jest rozwojowa. Mam nadzieję kontynuować badania w tym obszarze przez najbliższych kilka lat.

5 Informacja o wykazywaniu się istotną aktywnością naukową albo artystyczną realizowaną w więcej niż jednej uczelni, instytucji naukowej lub instytucji kultury, w szczególności zagranicznej

Jestem współautorem 29 prac konferencyjnych (w tym 6 prac opublikowanych przed obroną mojej pracy doktorskiej) oraz 10 prac naukowych publikowanych w czasopiśmie (w tym 3 opublikowane przed obroną mojej pracy doktorskiej). Pracowałem za granicą od listopada 2013 roku do września 2016 roku. Najpierw na Uniwersytecie Bayreuth pod kierunkiem profesora Wima Martensa nad teoretycznymi aspektami baz danych. Następnie w Laboratoire Spécification et Vérification, Centre National de la Recherche Scientifique w Paryżu pod kierunkiem profesora Stefana Göllera nad systemami o nieskończonej liczbie stanów. W tym czasie zbudowałem znaczną część mojej sieci międzynarodowych kontaktów naukowych. Współpracowałem m.in. z Patrickiem Tozke z Uniwersytetu w Liverpoolu, Dmitrem Chistikovem z Uniwersytetu w Warwick, Christofem Haase z Uniwersytetu Oksfordzkiego, Georgiem Zetzche z Instytutu Maxa Plancka, Michaeliem Blondinem z Université de Sherbrooke, Joanną Ochremiak z CNRS, Aksheyem S. z Indian Institute of Technology Bombay, Jeromem Lerux z CNRS, Ranko Laziciem z Uniwersytetu w Warwick i Bartoszem Klinem z Uniwersytetu Oksfordzkiego.

Moje zainteresowania badawcze nie ograniczają się wyłącznie do sieci z danymi. Po obronie pracy doktorskiej publikowałem w obszarach takich jak języki VAS-ów [51, 52, 53], systemy jednolicknikowe [54, 55, 56, 57, 58], synteza strategii [59, 60], teoria baz danych [61, 62], systemy z czasem [63], równoważności systemów o nieskończonej liczbie stanów [64] i workflowy (opisy procesów biznesowych) [65].

6 Informacja o osiągnięciach dydaktycznych, organizacyjnych oraz popularyzujących naukę lub sztukę

6.1 Dydaktyka

- Zajęcia prowadzone na Wydziale Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego:

Ćwiczenia:

- ćwiczenia z Teorii współbieżności,
- ćwiczenia z Inżynierii oprogramowania,
- ćwiczenia z sieci Petriego,
- ćwiczenia z wstępu do programowania,
- ćwiczenia z algorytmicznej teorii liczb,
- ćwiczenia z Teorii informacji,
- ćwiczenia z wstępu do informatyki dla humanistów.

Wykłady:

- wykład z inżynierii oprogramowania,
 - wykład z teorii współbieżności,
 - w 2018 dałem pojedynczy wykład z algebraicznych metod dla sieci Petriego w Indian Institute of Technology (IIT) Bombay.
- byłem zaangażowany organizację zajęć przygotowujących do olimpiady matematycznej w XIV liceum ogólnokształcącym imienia Stanisława Staszica w Warszawie.
 - Studenci:
 - Byłem opiekunem pracy licencjackiej Jakuba Różyckiego (z matematyki).
 - Jestem promotorem pomocniczym Mohnish-a Pattathurajan-a (obrona pracy jest wyznaczona na 13 października 2023).
 - Jestem opiekunem doktoranta Arka-i Gosh-a (mam nadzieję zostać powołany na drugiego promotora lub promotora pomocniczego).

Zaproszone wykłady

22 Sie. – 25 Wrz. 2016 wydarzenie “Automata, Logic and Games”, National University of Singapore, tytuł wykładu: “State equation for Unordered Data Petri Nets”

Komitety programowe

7 – 9 Wrz. 2017 11th International Workshop on Reachability Problems (Londyn, Wielka Brytania)

10 – 14 Wrz. 2018 Developments in Language Theory 2018 (Tokio, Japonia)

27 Mar. – 1 Kwi. 2021 24th International Conference on Foundations of Software Science and Computation Structures (Luksemburg, Luksemburg; online)

17 – 19 Paź. 2022 16th International Conference on Reachability Problems (Kaiserslautern, Niemcy)

Komitety organizacyjne (co-chair)

- 5 – 9 Sie. 2019 Developments in Language Theory 2019 (Warszawa, Polska)
- 12 – 17 Wrz. 2022 33rd International Conference on Concurrency Theory (Warszawa, Polska)

Inne osiągnięcia i działalność

- Wrz. 2021 – teraz Wdrożenie i zarządzenie programem Zaawansowanych studiów magisterskich na Uniwersytecie Warszawskim.
- 2016 – 2020 Przedstawiciel innych nauczycieli akademickich w Radzie Wydziału.
- 2016 Organizator Autobózu 2016 - Wyjazd badawczy grupy Automata, Logika i Gry z Uniwersytetu Warszawskiego (<https://autoboz.org/>).
- 2015 – 2016 Organizator seminarium dla grupy „INFINI: Algorytmy do Weryfikacji Symbolicznej” w LSV, CNRS & ENS de Cachan.
- 2015 Organizator Autobózu 2015 - Wyjazd badawczy grupy Automata, Logika i Gry z Uniwersytetu Warszawskiego.
- 2014 Organizator Autobózu 2014 - Wyjazd badawczy grupy Automata, Logika i Gry z Uniwersytetu Warszawskiego.

Przed uzyskaniem stopnia doktora.

- 2013 Organizator Autobózu 2013 - Wyjazd badawczy grupy Automaty, Logika i Gry z Uniwersytetu Warszawskiego.
- 2010 – 2011 Członek Rady Doktorantów na Wydziale Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego.
- 2010 – 2014 Stypendium doktoranckie finansowane przez Program Operacyjny Kapitał Ludzki Unii Europejskiej („Międzyuczelniane Studia Doktoranckie z Matematyki”).
- 2008 – 2009 Przewodniczący Samorządu Studentów na Wydziale Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego (w trakcie studiów magisterskich).
- 2007 – 2008 Przewodniczący Studenckiego Koła Naukowego „Pasjonatów Matematyki” (w trakcie studiów magisterskich).
- 2003 Finalista LIV Olimpiady Matematycznej.

7 Oprócz kwestii wymienionych w pkt. 1-6, wnioskodawca może podać inne informacje, ważne z jego punktu widzenia, dotyczące jego kariery zawodowej

Literatura

- [1] Mikołaj Bojańczyk, Bartek Klin, Sławomir Lasota, and Szymon Torunczyk. Turing machines with atoms. In *28th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2013, New Orleans, LA, USA, June 25-28, 2013*, pages 183–192. IEEE Computer Society, 2013.
- [2] Mikołaj Bojańczyk. Modelling infinite structures with atoms. In Leonid Libkin, Ulrich Kohlenbach, and Ruy J. G. B. de Queiroz, editors, *Logic, Language, Information, and Computation - 20th International Workshop, WoLLIC 2013, Darmstadt, Germany, August 20-23, 2013. Proceedings*, volume 8071 of *Lecture Notes in Computer Science*, pages 13–28. Springer, 2013.
- [3] Bartek Klin. μ -calculi with atoms (invited talk). In Christel Baier and Jean Goubault-Larrecq, editors, *29th EACSL Annual Conference on Computer Science Logic, CSL 2021, January 25-28, 2021, Ljubljana, Slovenia (Virtual Conference)*, volume 183 of *LIPICs*, pages 1:1–1:1. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.
- [4] Piotr Hofman, Sławomir Lasota, Ranko Lazić, Jérôme Leroux, Sylvain Schmitz, and Patrick Totzke. Coverability trees for petri nets with unordered data. In Bart Jacobs and Christof Löding, editors, *Foundations of Software Science and Computation Structures - 19th International Conference, FOSSACS 2016, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2016, Eindhoven, The Netherlands, April 2-8, 2016, Proceedings*, volume 9634 of *Lecture Notes in Computer Science*, pages 445–461. Springer, 2016.
- [5] Piotr Hofman, Jérôme Leroux, and Patrick Totzke. Linear combinations of unordered data vectors. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017*, pages 1–11. IEEE Computer Society, 2017.
- [6] Piotr Hofman and Sławomir Lasota. Linear equations with ordered data. In Sven Schewe and Lijun Zhang, editors, *29th International Conference on Concurrency Theory, CONCUR 2018, September 4-7, 2018, Beijing, China*, volume 118 of *LIPICs*, pages 24:1–24:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.
- [7] Utkarsh Gupta, Preey Shah, S. Akshay, and Piotr Hofman. Continuous reachability for unordered data petri nets is in ptime. In Mikołaj Bojańczyk and Alex Simpson, editors, *Foundations of Software Science and Computation Structures - 22nd International Conference, FOSSACS 2019, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2019, Prague, Czech Republic, April 6-11, 2019, Proceedings*, volume 11425 of *Lecture Notes in Computer Science*, pages 260–276. Springer, 2019.
- [8] Piotr Hofman and Jakub Różycki. Linear equations for unordered data vectors in $[d]^k \setminus \{z\}^d$. *Log. Methods Comput. Sci.*, 18(4), 2022.
- [9] Piotr Hofman, Marta Jucepczuk, Sławomir Lasota, and Mohnish Pattathurajan. Parikh’s theorem for infinite alphabets. In *36th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2021, Rome, Italy, June 29 - July 2, 2021*, pages 1–13. IEEE, 2021.
- [10] Arka Ghosh, Piotr Hofman, and Sławomir Lasota. Solvability of orbit-finite systems of linear equations. In Christel Baier and Dana Fisman, editors, *LICS '22: 37th Annual ACM/IEEE*

- [11] Arka Ghosh, Piotr Hofman, and Sławomir Lasota. Orbit-finite linear programming. In *LICS*, pages 1–14, 2023.
- [12] Rajeev Alur and David L. Dill. A theory of timed automata. *Theor. Comput. Sci.*, 126(2):183–235, 1994.
- [13] Michael Kaminski and Shlomit S. Pinter. Finite automata on directed graphs. *J. Comput. Syst. Sci.*, 44(3):425–446, 1992.
- [14] Edward Y. C. Cheng and Michael Kaminski. Context-free languages over infinite alphabets. *Acta Informatica*, 35(3):245–267, 1998.
- [15] Robin Milner. *Communicating and mobile systems - the Pi-calculus*. Cambridge University Press, 1999.
- [16] Carl. A. Petri. Fundamentals of a theory of asynchronous information flow. In *IFIP Congress*, pages 386–390, 1962.
- [17] John Hopcroft and Jean-Jacques Pansiot. On the reachability problem for 5-dimensional vector addition systems. *Theoretical Computer Science*, 8(2):135 – 159, 1979.
- [18] Estíbaliz Fraca and Serge Haddad. Complexity analysis of continuous petri nets. *Fundam. Informaticae*, 137(1):1–28, 2015.
- [19] Wil M. P. Van Der Aalst, Kees M. Van Hee, and Hajo A. Reijers. Analysis of discrete-time stochastic petri nets. *Statistica Neerlandica*, 54(2):237–255, 2000.
- [20] Stefan Göller, Christoph Haase, Ranko Lazić, and Patrick Totzke. A polynomial-time algorithm for reachability in branching VASS in dimension one. *CoRR*, abs/1602.05547, 2016.
- [21] Kumar Neeraj Verma and Jean Goubault-Larrecq. Karp-miller trees for a branching extension of VASS. *Discrete Mathematics & Theoretical Computer Science*, 7(1):217–230, 2005.
- [22] Alain Finkel, Pierre McKenzie, and Claudine Picaronny. A well-structured framework for analysing petri net extensions. *Inf. Comput.*, 195(1-2):1–29, 2004.
- [23] Rémi Bonnet, Alain Finkel, and M. Praveen. Extending the rackoff technique to affine nets. In Deepak D’Souza, Telikepalli Kavitha, and Jaikumar Radhakrishnan, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2012, December 15-17, 2012, Hyderabad, India*, volume 18 of *LIPICs*, pages 301–312. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2012.
- [24] Ranko Lazić, Tom Newcomb, Joël Ouaknine, A.W. Roscoe, and James Worrell. Nets with tokens which carry data. *Fund. Inform.*, 88(3):251–274, 2008.
- [25] Fernando Rosa-Velardo and David de Frutos-Escrig. Name creation vs. replication in petri net systems. *Fundam. Informaticae*, 88(3):329–356, 2008.
- [26] Fernando Rosa-Velardo and David de Frutos-Escrig. Decidability and complexity of petri nets with unordered data. *Theor. Comput. Sci.*, 412(34):4439–4451, 2011.
- [27] Sambasiva Rao Kosaraju. Decidability of reachability in vector addition systems. In *STOC’82*, pages 267–281. ACM, 1982.
- [28] Jean-Luc Lambert. A structure to decide reachability in petri nets. *Theor. Comput. Sci.*, 99(1):79–104, 1992.

- [29] Ernst W. Mayr. An algorithm for the general petri net reachability problem. *SIAM J. Comput.*, 13(3):441–460, 1984.
- [30] Jérôme Leroux and Sylvain Schmitz. Demystifying reachability in vector addition systems. In *30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015, Kyoto, Japan, July 6-10, 2015*, pages 56–67. IEEE Computer Society, 2015.
- [31] Jérôme Leroux and Sylvain Schmitz. Reachability in vector addition systems is primitive-recursive in fixed dimension. In *34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019, Vancouver, BC, Canada, June 24-27, 2019*, pages 1–13. IEEE, 2019.
- [32] Wojciech Czerwiński, Sławomir Lasota, Ranko Lazić, Jérôme Leroux, and Filip Mazowiecki. The reachability problem for petri nets is not elementary. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 24–33. ACM, 2019.
- [33] Wojciech Czerwiński and Łukasz Orlikowski. Reachability in vector addition systems is ackermann-complete. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 1229–1240. IEEE, 2021.
- [34] Jérôme Leroux. The reachability problem for petri nets is not primitive recursive. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 1241–1252. IEEE, 2021.
- [35] Alain Finkel and Philippe Schnoebelen. Well-structured transition systems everywhere! *Theor. Comput. Sci.*, 256(1-2):63–92, 2001.
- [36] Richard M. Karp and Raymond E. Miller. Parallel program schemata. *J. Comput. Syst. Sci.*, 3(2):147–195, 1969.
- [37] Catherine Dufourd, Petr Jančar, and Philippe Schnoebelen. Boundedness of reset P/T nets. In Jiri Wiedermann, Peter van Emde Boas, and Mogens Nielsen, editors, *Automata, Languages and Programming, 26th International Colloquium, ICALP’99, Prague, Czech Republic, July 11-15, 1999, Proceedings*, volume 1644 of *Lecture Notes in Computer Science*, pages 301–310. Springer, 1999.
- [38] Leonardo de Moura and Nikolaj Bjørner. Z3: An efficient smt solver. In C. R. Ramakrishnan and Jakob Rehof, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, pages 337–340, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.
- [39] Michael Blondin, Alain Finkel, Christoph Haase, and Serge Haddad. Approaching the coverability problem continuously. *CoRR*, abs/1510.05724, 2015.
- [40] Thomas Geffroy, Jérôme Leroux, and Grégoire Sutre. Occam’s razor applied to the petri net coverability problem. *Theor. Comput. Sci.*, 750:38–52, 2018.
- [41] Parosh Aziz Abdulla and Aletta Nylén. Timed petri nets and bqos. In José Manuel Colom and Maciej Koutny, editors, *Application and Theory of Petri Nets 2001, 22nd International Conference, ICATPN 2001, Newcastle upon Tyne, UK, June 25-29, 2001, Proceedings*, volume 2075 of *Lecture Notes in Computer Science*, pages 53–70. Springer, 2001.
- [42] Rémi Bonnet, Alain Finkel, Serge Haddad, and Fernando Rosa-Velardo. Comparing petri data nets and timed petri nets. 2010.
- [43] Michael Blondin, Alain Finkel, Christoph Haase, and Serge Haddad. The logical view on continuous petri nets. *ACM Trans. Comput. Log.*, 18(3):24:1–24:28, 2017.

- [44] Michael Blondin and Christoph Haase. Logics for continuous reachability in petri nets and vector addition systems with states. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017*, pages 1–12. IEEE Computer Society, 2017.
- [45] Sławomir Lasota and Radosław Piórkowski. WQO dichotomy for 3-graphs. In Christel Baier and Ugo Dal Lago, editors, *Foundations of Software Science and Computation Structures - 21st International Conference, FOSSACS 2018, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2018, Thessaloniki, Greece, April 14-20, 2018, Proceedings*, volume 10803 of *Lecture Notes in Computer Science*, pages 548–564. Springer, 2018.
- [46] Nikola Trcka, Wil M. P. van der Aalst, and Natalia Sidorova. Data-flow anti-patterns: Discovering data-flow errors in workflows. In Pascal van Eck, Jaap Gordijn, and Roel J. Wieringa, editors, *Advanced Information Systems Engineering, 21st International Conference, CAiSE 2009, Amsterdam, The Netherlands, June 8-12, 2009. Proceedings*, volume 5565 of *Lecture Notes in Computer Science*, pages 425–439. Springer, 2009.
- [47] Fernando Rosa-Velardo and María Martos-Salgado. Multiset rewriting for the verification of depth-bounded processes with name binding. *Inf. Comput.*, 215:68–87, 2012.
- [48] Yoram Hirshfeld. Petri nets and the equivalence problem. In Egon Börger, Yuri Gurevich, and Karl Meinke, editors, *Computer Science Logic, 7th Workshop, CSL '93, Swansea, United Kingdom, September 13-17, 1993, Selected Papers*, volume 832 of *Lecture Notes in Computer Science*, pages 165–174. Springer, 1993.
- [49] Rohit Parikh. On context-free languages. *J. ACM*, 13(4):570–581, 1966.
- [50] Daniela Kühn and Deryk Osthus. A survey on hamilton cycles in directed graphs. *Eur. J. Comb.*, 33(5):750–766, 2012.
- [51] Wojciech Czerwiński, Piotr Hofman, and Georg Zetsche. Unboundedness problems for languages of vector addition systems. *CoRR*, abs/1802.06683, 2018.
- [52] Wojciech Czerwiński, Diego Figueira, and Piotr Hofman. Universality problem for unambiguous VASS. In Igor Konnov and Laura Kovács, editors, *31st International Conference on Concurrency Theory, CONCUR 2020, September 1-4, 2020, Vienna, Austria (Virtual Conference)*, volume 171 of *LIPICs*, pages 36:1–36:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.
- [53] Wojciech Czerwiński and Piotr Hofman. Language inclusion for boundedly-ambiguous vector addition systems is decidable. In Bartek Klin, Sławomir Lasota, and Anca Muscholl, editors, *33rd International Conference on Concurrency Theory, CONCUR 2022, September 12-16, 2022, Warsaw, Poland*, volume 243 of *LIPICs*, pages 16:1–16:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.
- [54] Mohamed Faouzi Atig, Dmitry Chistikov, Piotr Hofman, K. Narayan Kumar, Prakash Saivasan, and Georg Zetsche. Complexity of regular abstractions of one-counter languages. *CoRR*, abs/1602.03419, 2016.
- [55] Stanislav Böhm, Stefan Göller, Simon Halfon, and Piotr Hofman. On büchi one-counter automata. In Heribert Vollmer and Brigitte Vallée, editors, *34th Symposium on Theoretical Aspects of Computer Science, STACS 2017, March 8-11, 2017, Hannover, Germany*, volume 66 of *LIPICs*, pages 14:1–14:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.
- [56] Piotr Hofman and Patrick Totzke. Trace inclusion for one-counter nets revisited. *Theor. Comput. Sci.*, 735:50–63, 2018.
- [57] Dmitry Chistikov, Wojciech Czerwiński, Piotr Hofman, Michał Pilipczuk, and Michael Wehar. Shortest paths in one-counter systems. *Log. Methods Comput. Sci.*, 15(1), 2019.

- [58] Shaull Almagor, Udi Boker, Piotr Hofman, and Patrick Totzke. Parametrized universality problems for one-counter nets. In Igor Konnov and Laura Kovács, editors, *31st International Conference on Concurrency Theory, CONCUR 2020, September 1-4, 2020, Vienna, Austria (Virtual Conference)*, volume 171 of *LIPICs*, pages 47:1–47:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.
- [59] Stanislav Böhm, Stefan Haar, Serge Haddad, Piotr Hofman, and Stefan Schwoon. Active diagnosis with observable quiescence. In *54th IEEE Conference on Decision and Control, CDC 2015, Osaka, Japan, December 15-18, 2015*, pages 1663–1668. IEEE, 2015.
- [60] Patricia Bouyer, Piotr Hofman, Nicolas Markey, Mickael Randour, and Martin Zimmermann. Bounding average-energy games. In Javier Esparza and Andrzej S. Murawski, editors, *Foundations of Software Science and Computation Structures - 20th International Conference, FOSSACS 2017, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017, Uppsala, Sweden, April 22-29, 2017, Proceedings*, volume 10203 of *Lecture Notes in Computer Science*, pages 179–195, 2017.
- [61] Claire David, Piotr Hofman, Filip Murlak, and Michał Pilipczuk. Synthesizing transformations from XML schema mappings. In Nicole Schweikardt, Vassilis Christophides, and Vincent Leroy, editors, *Proc. 17th International Conference on Database Theory (ICDT), Athens, Greece, March 24-28, 2014.*, pages 61–71. OpenProceedings.org, 2014.
- [62] Piotr Hofman and Wim Martens. Separability by short subsequences and subwords. In Marcelo Arenas and Martín Ugarte, editors, *18th International Conference on Database Theory, ICDT 2015, March 23-27, 2015, Brussels, Belgium*, volume 31 of *LIPICs*, pages 230–246. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015.
- [63] Lorenzo Clemente, Piotr Hofman, and Patrick Totzke. Timed basic parallel processes. *CoRR*, abs/1907.01240, 2019.
- [64] Christoph Haase and Piotr Hofman. Tightening the complexity of equivalence problems for commutative grammars. In Nicolas Ollinger and Heribert Vollmer, editors, *33rd Symposium on Theoretical Aspects of Computer Science, STACS 2016, February 17-20, 2016, Orléans, France*, volume 47 of *LIPICs*, pages 41:1–41:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.
- [65] Piotr Hofman, Filip Mazowiecki, and Philip Offtermatt. Fast termination and workflow nets. In Constantin Enea and Akash Lal, editors, *Computer Aided Verification - 35th International Conference, CAV 2023, Paris, France, July 17-22, 2023, Proceedings, Part I*, volume 13964 of *Lecture Notes in Computer Science*, pages 132–155. Springer, 2023.