Autoreferat

1 Imię i nazwisko

Michał Skrzypczak

2 Posiadane dyplomy i stopnie naukowe

- Magisterium z matematyki, *cum laude*, Uniwersytet Warszawski, 2010, tytuł pracy: „O kolorowaniach drzewa Cantora";
- Magisterium z informatyki, *cum laude*, Uniwersytet Warszawski, 2012, tytuł pracy: „O złożoności topologicznej języków definiowanych w logice MSO+U";
- Doktorat z matematyki, specjalność informatyka, *cum laude*, Uniwersytet Warszawski, 2014, tytuł rozprawy: „Descriptive set theoretic methods in automata theory".

3 Informacje o dotychczasowym zatrudnieniu

- 2015 – ..., adiunkt, Uniwersytet Warszawski;
- 2015, staż podoktorski, Liafa (IRIF), Université Denis Diderot – Paris 7;
- 2012, staż 4-miesięczny, LaBRI, Université Bordeaux;
- 2010 – 2014, asystent naukowy, Uniwersytet Warszawski.

4 Osiągnięcie

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 w zakresie sztuki (Dz. U. 2017 r. poz. 1789).

4.1 Tytuł osiągnięcia naukowego

Słabe formy niedeterminizmu w teorii automatów

4.2 Prace naukowe wchodzące w skład tego osiągnięcia

[A] Denis Kuperberg, Michał Skrzypczak.
On determinisation of GFG automata,
In ICALP, Springer-Verlag LNCS, pages 299–310, 2015.

[B] Michał Skrzypczak, Igor Walukiewicz.
Deciding the topological complexity of Büchi languages,
In ICALP, LIPIcs, pages 99:1–99:13, 2016.

[C] Filippo Cavallari, Henryk Michalewski, Michał Skrzypczak.
A characterisation of Π^0_2 regular tree languages,
In MFCS, LIPIcs, pages 56:1–56:14, 2017.

[D] Udi Boker, Orna Kupferman, Michał Skrzypczak.
How deterministic are GFG automata,
In FSTTCS, LIPIcs, pages 18:1–18:14, 2017.

[E] Michał Skrzypczak.
Unambiguous languages exhaust the index hierarchy,
In ICALP, LIPIcs, pages 140:1–140:14, 2018.

[F] Michał Skrzypczak.
Büchi VASS recognise Σ^1_1-complete ω-languages,
In RP, Springer-Verlag LNCS, pages 133–145, 2018.

4.3 Omówienie celu naukowego ww. prac i osiągniętych wyników

4.3.1 Wstęp

Niedeterminizm stanowi jedno z kluczowych pojęć współczesnej informatyki teoretycznej. Wiele fundamentalnych problemów otwartych tej dziedziny, jak chociażby pytanie czy P=NP, koncentruje się na zrozumieniu siły niedeterminizmu. Sytuacja jest nieco prostsza w przypadku teorii automatów, gdzie granice pomiędzy automatami deterministycznymi a niedeterministycznymi są lepiej zrozumiane. Celem prac z mojego osiągnięcia jest właśnie badanie własności typowych form niedeterminizmu w odniesieniu do standardowych modeli automatów.

Z abstrakcyjnego punktu widzenia, automat może być widziany jako skończona maszyna, parsująca daną strukturę i aktualizująca swój wewnętrzny stan w zależności od wczytanych liter z ustalonego alfabetu. Jak się okazuje, dopuszczenie niedeterminizmu (czyli *zgadywania*) w przypadku rozważanego modelu automatów, często znacząco zwiększa siłę wyrazu danego modelu. Ponadto, nawet w sytuacjach, gdy siła wyrazu maszyn niedeterministycznych pozostaje taka sama, automaty niedeterministyczne są często wykładniczo mniejsze od równoważnych im automatów deterministycznych. Rozbieżność ta staje się jeszcze większa, gdy rozważymy modele maszyn alternujących, gdzie oprócz standardowego niedeterminizmu widzianego jako tryb *egzystencjalny*, dopuszczony jest również dualny tryb *uniwersalny*. Interakcję pomiędzy tymi dwoma trybami modeluje się tam za pomocą gry.

Niestety, opisane powyższej korzyści płynące z większej siły wyrazu czy mniejszej liczby stanów automatów niedeterministycznych, nie przychodzą za darmo. Radzenie sobie z niedeterminizmem lub alternacją danej maszyny zwykle powoduje dodatkowe komplikacje w matematycznym wnioskowaniu na temat jej własności. Często oznacza to, że maszyny takie nie mogą być bezpośrednio używane w pewnych zastosowaniach (np. automaty niedeterministyczne nie nadają się do rozwiązywania problemu syntezy). Ponadto, konieczność rozpatrzenia wielu ścieżek wykonania danej maszyny znacznie podnosi złożoność obliczeniową typowych problemów decyzyjnych i konstrukcji. W skrajnych przypadkach prowadzi to do nierozstrzygalności (jak np. dla problemu niepustości automatów alternujących ze stosem).

Biorąc pod uwagę komplikacje związane z niedeterminizmem i alternacją, naturalny wydaje się pomysł poszukiwania modeli pośrednich, gdzie siła tych trybów jest częściowo ograniczona. Najlepiej byłoby przy tym nie wyrugować w pełni jakiejś możliwości *zgadywania*, by móc skorzystać z rozszerzonej siły wyrazu lub zwięzłości oferowanych przez brak pełnego determinizmu. Oznacza to konieczność poszukiwania *złotego środka*: celem jest znalezienie modelu, dla którego koszt radzenia sobie z niedeterminizmem jest możliwie mały, przy jednoczesnym zachowaniu możliwie wielu korzyści płynących ze *zgadywania*. W ramach badań w tym kierunku wyodrębniono następujące pośrednie formy niedeterminizmu (dokładniejszy opis tych form oraz referencje podane są w dalszej części autoreferatu):

(**D**) **automaty deterministyczne**, całkowicie pozbawione możliwości *zgadywania*;

(**U**) **automaty jednoznaczne** (ang. unambiguous), czyli te automaty niedeterministyczne, które spełniają dodatkowy warunek semantyczny, by dla każdego wejścia istniał co najwyżej jeden sposób *zgadywania* (czyli bieg) prowadzący do akceptacji;

(**G**) **automaty *Good-For-Games*** (ozn. GFG), czyli te automaty niedeterministyczne, w których istnieje metoda pozwalająca dokonywać *zgadywania*, w oparciu wyłącznie o już wczytaną część wejścia;

(**N**) **automaty niedeterministyczne**, działające stale w trybie *egzystencjalnym*;

(**A**) **automaty alternujące**, gdzie tryby *egzystencjalny* i *uniwersalny* wchodzą w nieograniczoną interakcję.

Kluczowe zalety płynące z silniejszych form niedeterminizmu są dość jasne: automaty niedeterministyczne mogą *zgadywać* kolejne przejścia; zaś automaty alternujące są w prosty sposób zamknięte ze względu na kombinacje boolowskie (dysjunkcję, koniunkcję i negację). Z drugiej strony, liczne wyniki pokazują, że rezygnacja z pełnej siły niedeterminizmu może przynosić wymierne korzyści w odniesieniu do rozważanych zastosowań. Klasycznym przykładem mogą być automaty jednoznaczne na słowach skończonych: podobnie do automatów niedeterministycznych, są one prosto zamknięte ze względu na operację odwracania słowa wejściowego; a jednocześnie problem ich uniwersalności można rozwiązywać w czasie wielomianowym [SI85], pomimo tego że jest on PSCAPE-zupełny dla ogólnych automatów niedeterministycznych [SM73]. Podobnie, automaty GFG mogą być używane wymiennie z automatami deterministycznymi do rozwiązywania problemu syntezy; a ich symboliczne reprezentacje bywają prostsze od równoważnych automatów deterministycznych [HP06] (Twierdzenie 4.7 poniżej pokazuje dalsze przewagi automatów GFG nad deterministycznymi).

Celem naukowym prowadzonych przeze mnie badań było precyzyjne zrozumienie zależności pomiędzy klasami automatów, odpowiadającymi wymienionym powyżej formom niedeterminizmu. Kładłem przy tym szczególny nacisk na poszukiwanie kompromisów pomiędzy siłą wyrazu, liczbą stanów, warunkiem akceptacji i złożonością odpowiednich problemów obliczeniowych.

Jak wynika bezpośrednio z definicji powyższych form niedeterminizmu, zachodzą pomiędzy nimi oczywiste inkluzje, mówiące że każdy automat z danej klasy może być traktowany jako automat z klasy szerszej: (**D**) $\subseteq$ (**U**),(**G**) $\subseteq$ (**N**) $\subseteq$ (**A**); przy czym klasy (**U**) i (**G**) są *a priori* nieporównywalne. Inkluzje odwrotne, rozumiane literalnie w sensie zawierania się odpowiednich klas automatów, w prosty sposób nie zachodzą dla większości modeli.

Praktycznie każda inkluzja z powyższej listy daje potencjalne miejsce na znalezienie kompromisu, pozwalającego tworzyć efektywniejsze algorytmy operujące na danych automatach. W tym celu należy znaleźć odpowiedzi na fundamentalne pytania dążące do zrozumienia natury takiej inkluzji. Pierwszym z nich jest pytanie o zawieranie się odpowiednich klas języków:

Problem 4.1 *Czy każdy język rozpoznawany automatem z klasy szerszej, daje się też rozpoznawać jakimś automatem z klasy węższej?*

W przypadku, gdy zawieranie się klas języków nie zachodzi, konieczne jest zrozumienie, które języki z szerszej klasy należą też do węższej:

Problem 4.2 *Czy istnieje dokładna charakteryzacja, lub chociaż górne szacowanie złożoności, mówiące które języki rozpoznawane automatami z klasy szerszej dają się również rozpoznawać automatem z klasy węższej?*

Z kolei w przypadku, gdy klasy języków pokrywają się, istotne jest zrozumienie jaki jest koszt transformacji automatów z klasy szerszej do węższej:

Problem 4.3 *Czy daje się oszacować wzrost komplikacji (np. liczby stanów) przy translacji automatu z klasy szerszej, do równoważnego automatu z klasy węższej?*

Ponadto, ze względu na asymetrię pojęcia niedeterminizmu, ważną rolę w badaniu powyższych problemów odgrywają analogiczne pytania dla języków z klasy dualnej, czyli dopełnienia języków rozpoznawanych przez rozważane automaty.

Oczywiście odpowiedzi na te pytania zależą od konkretnego modelu rozważanych maszyn. Rozważa się np. automaty skończone, automaty z licznikami, czy automaty o różnych warunkach akceptacji. Sytuacja zmienia się też w zależności od rozważanych struktur (słów, drzew, itp.). Ze względu na fakt, że wiele pytań upraszcza się dla struktur skończonych, większość badań koncentruje się na słowach i drzewach nieskończonych.

Główne wyniki mojego osiągnięcia naukowego koncentrują się na szukaniu odpowiedzi na powyższe pytania w odniesieniu do wymienionych powyżej modeli automatów. Ze względu na nieco inny zestaw stosowanych technik, opis wyników wchodzących w skład osiągnięcia naukowego został rozbity na dwa nurty.

Pierwszy z omawianych nurtów dotyczy automatów na słowach nieskończonych. Rozważane w tym przypadku warunki akceptacji to warunki Büchiego, co-Büchiego, Rabina, Streetta i parzystości (wraz z jego *słabym* wariantem). Badane w tym nurcie są automaty skończone oraz ich rozszerzenie o *ślepe liczniki.* Prace wchodzące w skład tego nurtu to prace [A], [D] oraz [F]. Rozważane w tym nurcie formy niedeterminizmu to automaty GFG oraz jednoznaczne.

Drugi z nurtów dotyczy drzew nieskończonych. Badana jest tutaj granica pomiędzy automatami jednoznacznymi, niedeterministycznymi oraz alternującymi. Główny nacisk kładziony jest na warunki akceptacji Büchiego oraz słabe warunki parzystości. Prace z tego nurtu to prace [B], [C] oraz [E].

Wyniki tych prac są zaprezentowane w odniesieniu do wcześniejszego stanu wiedzy. Z tego względu cytowane są twierdzenia i hipotezy z innych publikacji, nie wchodzących w skład osiągnięcia naukowego. Twierdzenia pochodzące z prac wchodzących w skład tego osiągnięcia są oznaczone znakiem ★ na marginesie.

4.3.2 Pojęcia wstępne

Ogólne wprowadzenie do używanych pojęć i notacji teorii automatów można znaleźć między innymi w [Tho96]. Odniesienia do deskryptywnej teorii mnogości są oparte o prezentację w [Kec95]. Używana tu notacja i nomenklatura pochodzi z [Skr16].

Przypomnijmy, że *automat niedeterministyczny* na słowach nieskończonych to krotka $\mathcal{A} = \langle \Sigma, Q, q_{\mathrm{I}}, \delta, \lambda \rangle$, gdzie Σ jest *alfabetem* skończonym; Q to skończony zbiór *stanów* automatu; $q_{\mathrm{I}} \in Q$ to stan *początkowy*; $\delta \subseteq Q \times \Sigma \times Q$ to *relacja przejścia* automatu; zaś λ to *warunek akceptacji*. Automat *deterministyczny* to szczególny przypadek automatu niedeterministycznego, w którym $\delta \colon Q \times \Sigma \to Q$ jest zależnością funkcyjną. *Biegiem* automatu na danym słowie nieskończonym $\alpha \in \Sigma^\omega$ jest nieskończone słowo $\rho \in Q^\omega$ etykietowane stanami automatu, takie że $\rho(0) = q_{\mathrm{I}}$ zaś dla $n = 0, 1, \ldots$ kolejne stany $\rho(n)$ i $\rho(n{+}1)$ zgadzają się z literą $\alpha(n)$ i relacją przejścia δ, czyli: $\big(\rho(n), \alpha(n), \rho(n{+}1)\big) \in \delta$.

Warunek akceptacji λ decyduje, które ciągi stanów $\rho \in Q^\omega$ są uważane za *akceptujące*. Warunek *parzystości* to funkcja $\lambda \colon Q \to \mathbb{N}$ przypisująca stanom ich *priorytety*. Nieskończony ciąg stanów spełnia ten warunek, jeśli największy priorytet występujący nieskończenie często jest liczbą parzystą. Mówimy, że warunek parzystości λ ma *indeks* (i, j) dla $i \leqslant j$, jeśli $\mathrm{rg}(\lambda) \subseteq \{i, \ldots, j\}$. *Słaby* warunek parzystości to warunek parzystości, w którym priorytety stanów są niemalejące wzdłuż przejść automatu. Warunki *Büchiego* i dualny (nazywany *co-Büchiego*) można rozumieć jako warunki parzystości odpowiednio indeksu $(1, 2)$ i $(0, 1)$. Warunek *Rabina* to dysjunkcja pewnej rodziny warunków parzystości indeksu $(1, 3)$. Warunek *Streetta* to negacja warunku Rabina.

Automat na nieskończonych drzewach binarnych jest podobny do automatu na słowach oprócz tego, że jego relacja przejścia to podzbiór $Q \times \Sigma \times Q \times Q$ w przypadku niedeterministycznym; lub funkcja $\delta \colon Q \times \Sigma \to Q \times Q$ w przypadku deterministycznym. Bieg takiego automatu na drzewie t to drzewo $\rho \colon \{\mathtt{L}, \mathtt{R}\}^* \to Q$, takie że $\rho(\epsilon) = q_{\mathrm{I}}$ oraz dla każdego wierzchołka $v \in \{\mathtt{L}, \mathtt{R}\}^*$ zachodzi $\big(\rho(v), t(v), \rho(v\mathtt{L}), \rho(v\mathtt{R})\big) \in \delta$. Bieg taki jest akceptujący, jeśli spełnia warunek akceptacji na wszystkich gałęziach nieskończonych drzewa.

Język $\mathrm{L}(\mathcal{A})$ rozpoznawany przez dany automat niedeterministyczny to zbiór tych słów lub drzew nieskończonych, na których istnieje bieg akceptujący. Automaty alternujące różnią się od automatów niedeterministycznych tym, że funkcja przejścia przypisuje stanom i literom pozytywne kombinacje boolowskie kolejnych stanów. Semantykę takich przejść interpretuje się w terminach gry. Słowo lub drzewo nieskończone należy do języka rozpoznawanego przez taki automat, jeśli gracz pozytywny ma strategię wygrywającą w odpowiedniej grze. Język (czyli zbiór słów lub drzew) nazywamy *regularnym*, jeśli daje się rozpoznawać niedeterministycznym (równoważnie alternującym) automatem parzystości.

Automat niedeterministyczny nazywamy *jednoznacznym*, jeśli na każdym wejściu (słowie lub drzewie) posiada on co najwyżej jeden bieg akceptujący.

4.3.3 Niedeterminizm w przypadku słów – automaty GFG

Prace [A] i [D] koncentrują się na przypadku automatów na słowach nieskończonych. Celem tych prac jest zrozumienie różnic pomiędzy automatami deterministycznymi a automatami GFG.

Definicja 4.4 *Automat niedeterministyczny $\mathcal{A}$ na słowach nieskończonych jest nazywany* GFG *(ang. Good-For-Games), jeśli istnieje funkcja $\sigma \colon \Sigma^* \to Q$, taka, że dla każdego słowa nieskończonego akceptowanego przez automat $\alpha \in \mathrm{L}(\mathcal{A}) \subseteq \Sigma^\omega$, ciąg stanów $\rho \in Q^\omega$ uzyskany jako $\rho(n) \stackrel{def}{=} \sigma(\alpha\restriction_n)$ dla $n = 0, 1, \ldots$ jest biegiem akceptującym $\mathcal{A}$ na α.*

Oznacza to, że istnieje sposób (*podpowiedź*), pozwalający wybierać przejścia automatu GFG w oparciu jedynie o już wczytaną część słowa wejściowego. Powyższe wymaganie mówi, że ilekroć wczytywane słowo należy do języka rozpoznawanego przez ten automat (czyli daje się przez niego zaakceptować w sposób niedeterministyczny), to stany wybrane w oparciu o powyższą *podpowiedź* stanowią bieg akceptujący. Innymi słowy jest możliwe, by w każdym momencie czytania słowa wejściowego wybierać kolejne przejście, bez znajomości dalszej części słowa wejściowego, i mimo to zaakceptować wszystkie słowa, które w ogóle daje się zaakceptować.

Klasa automatów GFG została wprowadzona w pracy [HP06]. Motywacja stojąca za tą definicją jest taka, że automat GFG (w przeciwieństwie do ogólnego automatu niedeterministycznego) może być stosowany bezpośrednio w kontekście gier, na przykład w rozstrzyganiu problemu syntezy. Oznacza to, że znalezienie automatów GFG, które są prostsze niż równoważne im automaty deterministyczne, dawałoby szansę na szybsze algorytmy dla tego problemu. Niezależne badania nad pojęciem GFG, pod nazwą automatów *history deterministic*, były prowadzone w odniesieniu do warunków ograniczoności i funkcji kosztu [CL10, Col13].

Szczególnym przypadkiem automatów GFG są takie, gdzie daje się znaleźć równoważny automat deterministyczny wewnątrz struktury danego automatu:

Definicja 4.5 *Niedeterministyczny automat $\mathcal{A}$ nazywamy* DBP *(ang. determinisable by pruning), jeśli istnieje automat $\mathcal{A}'$ powstały z $\mathcal{A}$ przez usuwanie stanów i przejść, taki że $\mathcal{A}'$ jest deterministyczny, a języki $\mathcal{A}$ i $\mathcal{A}'$ są równe.*

Ze względu na fakt, że już deterministyczne automaty parzystości rozpoznają wszystkie języki regularne słów nieskończonych, odpowiedź na Problem 4.1 jest w sposób trywialny prawdziwa, zarówno dla automatów GFG jak i DBP.

Punktem wyjścia do rozważań z pracy [A] była wcześniejsza praca [BKKS13] pokazująca następujące dwa twierdzenia, odpowiadające na odpowiednie wersje problemu inkluzji dla rozważanych klas automatów:

Twierdzenie 4.6 ([BKKS13])

a) Klasa automatów GFG pokrywa się z klasą automatów Good-For-Trees[1]*.*

b) Istnieją automaty GFG z warunkami Büchiego i co-Büchiego, które nie są DBP.

Pierwszy z wyników wskazuje na związek pomiędzy automatami GFG na słowach nieskończonych, a pewną klasą języków drzew (klasa ta może być widziana jako pewne zawężenie języków rozpoznawanych przez automaty deterministyczne). W tym kontekście, wyniki dotyczące np. determinizacji automatów GFG mogą być widziane jako wyniki dotyczące determinizacji szczególnych przypadków automatów niedeterministycznych na drzewach nieskończonych.

Drugi z powyższych wyników sygnalizuje, że automaty GFG mogą mieć istotnie niedeterministyczną strukturę. Wynik ten jednak nie rozstrzygnął, czy automaty GFG mogą być istotnie prostsze od deterministycznych – dla podanych tam przykładów istnieją równie proste automaty deterministyczne. Innymi słowy, Problem 4.3 pozostał otwarty w przypadku automatów GFG. To właśnie pytanie było punktem wyjścia w pracy [A], której główne wyniki dotyczące determinizacji są następujące:

[1] Automaty Good-For-Trees to automaty na słowach nieskończonych, które mogą być użyte do rozpoznawania języków drzew nieskończonych postaci $\mathrm{Path}(L)$ – język taki zawiera wszystkie drzewa częściowe, których wszystkie nieskończone ścieżki należą do danego języka słów $L \subseteq \Sigma^\omega$.

★ **Twierdzenie 4.7** ([A])

a) (Twierdzenie 1.) Istnieją automaty GFG z warunkiem co-Büchiego o $2n+1$ *stanach, takie że każdy równoważny im automat deterministyczny ma przynajmniej* $\frac{2^n}{2n+1}$ *stanów.*

b) (Twierdzenie 8.) Istnieje algorytm, który wczytuje automat GFG z warunkiem Büchiego o n *stanach i zwraca równoważny mu automat deterministyczny o* $O(n^2)$ *stanach.*

Dowód Punktu a) tego twierdzenia przebiega przez konstrukcję rodziny automatów GFG. o ile fakt, że automaty te istotnie są GFG jest dość prosty, o tyle wykazanie dolnego ograniczenia na rozmiar ich determinizacji wymaga argumentu w stylu pompowania. Co ciekawe, argument nie przebiega przez wskazanie bezpośredniego schematu (w stylu pętli w jakimś grafie), którego rozpatrzenie prowadzi do poszukiwanej sprzeczności. Zamiast tego, użyty jest argument pompowania granicznego: tworzona jest pewna rodzina biegów o określonych własnościach, a następnie dzięki topologicznej zwartości odpowiedniego zbioru biegów, wybierany jest bieg *graniczny*, spełniający jednocześnie wszystkie te własności.

Dowód Punktu b) powyższego twierdzenia opiera się na konkretnym algorytmie wczytującym automat GFG $\mathcal{A}$ i przeprowadzającym na nim modyfikacje. Punktem wyjścia jest rozważenie gry sprawdzającej, czy $\mathcal{A}$ istotnie jest GFG (jest to tzw. *gra GFG*). Dzięki założeniu na temat $\mathcal{A}$, wiemy że gracz pozytywny ma strategię wygrywającą w tej grze. Następnie należy rozpatrzyć pewną konkretną strategię tego gracza, która optymalizuje tzw. *sygnatury parzystości* [SE89, Wal96]. Dobre własności kombinatoryczne tej strategii umożliwiają przeprowadzenie indukcyjnego uproszczania jej struktury, aż do otrzymania małego automatu deterministycznego dla języka $\mathrm{L}(\mathcal{A})$.

Ponieważ warunek Büchiego odpowiada indeksowi parzystości $(1,2)$ (analogicznie warunek co-Büchiego odpowiada indeksowi $(0,1)$), to wyniki Twierdzenia 4.7 dają pełną klasyfikację wzrostu liczby stanów przy determinizacji automatów GFG z warunkiem parzystości: wzrost jest wielomianowy dla warunków indeksu $(1,2)$ i niższych; a wykładniczy dla warunków indeksu $(0,1)$ i wyższych. Ponadto, Punkt a) Twierdzenia 4.7 pokazuje, że automaty GFG mogą być istotnie mniejsze od równoważnych im automatów niedeterministycznych. Oznacza to, że faktycznie w pewnych sytuacjach zastosowanie automatów GFG może prowadzić do wykładniczo szybszego rozwiązania problemu syntezy, aniżeli opierając się na automatach deterministycznych.

Kontynuacja badań nad różnicami pomiędzy automatami deterministycznymi a GFG była prowadzona w pracy [D]. Przyjęta tam miara podobieństwa, zamiast liczby stanów, oparta była na pojęciu *typowalności*. Poniższa definicja prezentuje to pojęcie w odniesieniu do automatów GFG:

Definicja 4.8 *Rozważmy dwa typy warunków akceptacji* λ *i* λ'*. Mówimy, że automaty GFG* λ *są* λ'-typowalne *jeśli: ilekroć* $\mathcal{A}$ *jest automatem GFG o warunku* λ*, takim że język* $\mathrm{L}(\mathcal{A})$ *daje się rozpoznawać deterministycznym automatem o warunku* λ'*, to istnieje automat GFG* $\mathcal{A}'$ *o tej samej strukturze co* $\mathcal{A}$*, ale z warunkiem akceptacji* λ'*.*

Pojęcie to zostało wprowadzone w pracy [KPB94] w odniesieniu do automatów deterministycznych. Z kolei w pracy [KMM06] autorzy wykazują przykłady braku własności λ'-typowalności dla automatów niedeterministycznych. Celem pracy [D] było zbadanie w jakim zakresie pojęcie to przenosi się na automaty GFG, czyli na ile są one podobne do automatów deterministycznych. Zgodnie z definicją, pojęcie bycia λ-typowalnym może być widziane

<table>
<tr><th>Dop. w
z</th><th>W</th><th>C</th><th>B</th><th>P</th><th>R</th><th>S</th></tr>
<tr><td>Słaby</td><td colspan="6" rowspan="2">Wielomianowy</td></tr>
<tr><td>Büchiego</td></tr>
<tr><td>Co-Büchiego</td><td rowspan="4"></td><td rowspan="3"></td><td colspan="3" rowspan="3">Wykł.</td><td rowspan="3">?</td></tr>
<tr><td>Parzystości</td></tr>
<tr><td>Rabina</td></tr>
<tr><td>Streetta</td><td colspan="5"></td></tr>
</table>

Tablica 1: Wzrost liczby stanów przy dopełnianiu automatów GFG. Użyte są oznaczenia na warunki akceptacji: W – słaby parzystości; C – co-Büchiego; B – Büchiego; P – parzystości; R – Rabina i S – Streetta.

jako instancja Problemu 4.3, gdzie jako komplikację rozumiemy złożoność warunku akceptacji automatu, bez zmiany jego struktury.

Główne wyniki [D] dotyczące typowalności automatów GFG są przedstawione poniżej. Wyniki te korzystają z pojęcia *ścisłości* automatu GFG (ang. tightness), mówiącego że automat nie zawiera zbędnych przejść z punktu widzenia pewnej funkcji σ zaświadczającej temu, że jest on GFG.

★ **Twierdzenie 4.9** ([D])

a) (Twierdzenie 10.) Ścisłe automaty GFG z warunkiem Streetta są co-Büchi-typowalne.

b) (Twierdzenie 11.) Ścisłe automaty GFG z warunkiem Rabina są Büchi-typowalne.

b) (Przykłady 11., 12.) Założenie ścisłości w obu powyższych twierdzeniach jest konieczne.

c) (Wniosek 16.) Ścisłe automaty GFG z warunkami Streetta i Rabina są typowalne dla słabego warunku parzystości.

Ponadto, praca ta przyniosła kolejne rozszerzenia, oparte o wyniki z pracy [A], dotyczące determinizacji i dopełniania automatów GFG:

★ **Twierdzenie 4.10** ([D])

a) (Wniosek 14.) Rozważmy automat GFG $\mathcal{A}$ z warunkiem Rabina i n stanami. Jeśli $\mathrm{L}(\mathcal{A})$ *daje się rozpoznawać deterministycznym automatem Büchiego, to istnieje deterministyczny automat Büchiego rozpoznający* $\mathrm{L}(\mathcal{A})$ *o $O(n^2)$ stanach.*

b) (Twierdzenie 17.) Automaty GFG ze słabym warunkiem parzystości są DBP.

c) (Twierdzenie 20.) Dopełnienie automatów GFG ze zmianą warunków akceptacji jest podsumowane w Tabeli 1 (patrz Tabela 2 w pracy [D]).

Wreszcie, Sekcja 3 pracy [D] rozważa pojęcie typowalności dla automatów jednoznacznych: definicja typowalności jest taka sama jak w Definicji 4.8, z tym że wymagamy by automat $\mathcal{A}$ był jednoznaczny (a niekoniecznie GFG), zaś $\mathcal{A}'$ może być dowolnym automatem niedeterministycznym. Uzyskane w tym zakresie wyniki są następujące:

★ **Twierdzenie 4.11** ([D])

a) *(Przykłady 7. i 8.) Jednoznaczne automaty parzystości nie są ani Büchi, ani co-Büchi typowalne.*

b) *(Stwierdzenie 9.) Jednoznaczne automaty parzystości które są GFG, są też DBP.*

Przedstawione wyniki pracy [D] można podsumować następująco. Po pierwsze, automaty GFG mają podobne własności typowalności co automaty deterministyczne. Po drugie, pod względem dopełniania automaty GFG znajdują się ściśle pomiędzy automatami deterministycznymi a niedeterministycznymi. Wreszcie Punkt a) Twierdzenia 4.11 pokazuje, że z perspektywy typowalności automaty jednoznaczne bardziej przypominają niedeterministyczne niż deterministyczne. Z kolei Punkt b) Twierdzenia 4.11 pokazuje, że wymaganie jednoznaczności trywializuje automaty GFG.

4.3.4 Niedeterminizm w przypadku słów – automaty ze ślepymi licznikami

Jednym z najprostszych sposobów wzmocnienia siły wyrazu automatów skończonych jest dodanie *ślepych liczników*. Automat taki, nazywany automatem BMC (ang. blind multi-counter), oprócz zbioru stanów Q, posiada też skończony zbiór *liczników* C o wartościach w liczbach naturalnych. Początkowo wszystkie liczniki przyjmują wartość 0. Relacja przejścia takiego automatu to skończony zbiór δ, zawierający *przejścia*. Przejście $(q, a, \tau, q') \in \delta$ oznacza, że automat może ze stanu $q \in Q$ wczytać literę $a \in \Sigma$ i przejść do stanu $q' \in Q$, zmieniając przy tym wartości liczników zgodnie z podanym wektorem $\tau \in \mathbb{Z}^C$ – jeżeli oznaczałoby to, że któryś licznik musi przyjąć wartość ujemną, to przejście takie nie jest możliwe. Oprócz powyższego ograniczenia, wartości liczników nie wpływają na działanie automatu, stąd nazwa mówiąca o *ślepych licznikach*. Warunek akceptacji λ takiego automatu jest zadany wyłącznie w oparciu o ciąg odwiedzonych stanów.

Pojęcie automatów BMC jest motywowane modelami VASS (ang. vector addition systems with states) oraz powiązanymi z nimi sieciami Petriego. Ze względu na wrodzony niedeterminizm tych modeli, również automaty BMC zwykle rozpatruje się w wariancie niedeterministycznym. Fakt, że automaty takie nie mogą *explicite* testować wartości swoich liczników powoduje, że problem niepustości jest dla tego modelu rozstrzygalny. Dodatkowo, ze względu na dostępny w tym modelu niedeterminizm, nietrudno pokazać, że siła wyrazu tych automatów nie zależy od tego czy warunek akceptacji to warunek Büchiego, Streetta, Rabina czy parzystości. Z tego względu często mówimy po prostu o niedeterministycznych automatach BMC, bez specyfikowania warunku.

Praca [FS14] pokazuje, że w ogólności niedeterministyczne automaty BMC są silniejsze od swoich deterministycznych wariantów, czyli stanowi odpowiedź na Problem 4.1 w odniesieniu do tej klasy:

Twierdzenie 4.12 ([FS14]) *Istnieje język rozpoznawany przez niedeterministyczny automat BMC, którego nie rozpoznaje żaden deterministyczny automat BMC z warunkiem parzystości*[2].

Rozumowanie podane w pracy [FS14] jest oparte o argument topologiczny: zaprezentowano tam przykład języka o złożoności topologicznej $\mathbf{\Sigma}^0_3$, która wykracza poza złożoność topologiczną deterministycznych maszyn z warunkiem parzystości. Przykład ten w ograniczonym

[2] W istocie język ten nie jest rozpoznawany nawet przez deterministyczne maszyny Turinga z warunkiem parzystości.

stopniu korzysta z niedeterminizmu, sprawiało to wrażenie, że trudno jest podać istotnie bardziej skomplikowany język w obrębie rozważanego modelu. Sugerowało to, że klasa $\mathbf{\Sigma}^0_3$ stanowi jednocześnie górne ograniczenie złożoności topologicznej niedeterministycznych automatów BMC. W takim przypadku byłoby niewykluczone, że istnieje model maszyn deterministycznych (albo chociaż jednoznacznych), z relatywnie prostym warunkiem akceptacji (np. w klasie $\mathbf{\Sigma}^0_3$), które są w stanie rozpoznawać wszystkie języki rozpoznawane przez niedeterministyczne automaty BMC. Pomimo badań, modelu takiego nie udało się znaleźć.

Odpowiedź na pytanie o istnienie takiego modelu przyniosła praca [F], pokazująca że niedeterminizm w przypadku automatów BMC jest w pewnym sensie inherentny:

★ **Twierdzenie 4.13** ([F]) *Istnieje język rozpoznawany przez niedeterministyczny automat BMC o jednym ślepym liczniku, którego złożoność topologiczna to $\mathbf{\Sigma}^1_1$ (w szczególności język ten jest nieborelowski). Języka tego nie rozpoznaje ani żadna maszyna deterministyczna, ani jednoznaczna o borelowskim (a tym bardziej $\mathbf{\Sigma}^0_3$) warunku akceptacji.*

Jakkolwiek praca [F] nie wspomina o modelach GFG, podany przykład również ten model wyklucza, ze względu na następujący wniosek:

Wniosek 4.14 *Język z Twierdzenia 4.13 nie jest rozpoznawany przez żadną maszynę GFG o borelowskim warunku akceptacji.*

Szkic dowodu. Załóżmy przeciwnie i niech $\mathcal{A}$ będzie maszyną jak w treści wniosku. Załóżmy, że δ to (co najwyżej przeliczalny) zbiór przejść tej maszyny. Funkcja[3] $\sigma\colon \Sigma^+ \to \delta$ świadcząca o tym, że $\mathcal{A}$ jest GFG, zadaje ciągłą transformację $\widehat{\sigma}\colon \Sigma^\omega \to \delta^\omega$. W związku z tym wykres tej transformacji $\mathrm{Graph}(\widehat{\sigma}) \subseteq \big(\Sigma \times \delta\big)^\omega$ jest zbiorem domkniętym, o sekcjach mocy co najwyżej 1. Język $\mathrm{L}(\mathcal{A})$ jest rzutem na współrzędną Σ^ω przecięcia zbioru $\mathrm{Graph}(\widehat{\sigma})$ z borelowskim zbiorem biegów akceptujących. A zatem, analogicznie jak w Twierdzeniu 2 z [F], język $\mathrm{L}(\mathcal{A})$ jest borelowski, co zaprzecza Twierdzeniu 4.13. ■

Wyniki te oznaczają, że nawet w przypadku jednego ślepego licznika, automaty BMC wykazują się pełną siłą niedeterminizmu, a modele deterministyczne, jednoznaczne czy GFG są istotnie słabsze.

4.3.5 Niedeterminizm w przypadku drzew – klasy z dealternacją

W przypadku automatów parzystości na słowach nieskończonych, wszystkie formy niedeterminizmu mają tę samą siłę wyrazu. Własność ta przestaje być prawdą, gdy rozważane modele to drzewa nieskończone. Po pierwsze, automaty deterministyczne mają istotnie słabszą siłę wyrazu od niedeterministycznych. Po drugie, jakkolwiek siła wyrazu automatów niedeterministycznych i alternujących z warunkiem parzystości jest taka sama, to transformacja od automatu niedeterministycznego do alternującego może w niekontrolowany sposób zwiększyć jego indeks. Wyjątek od tej zasady stanowi klasa automatów Büchiego, które definiują podklasę języków regularnych drzew o szczególnie silnych własnościach:

Twierdzenie 4.15 ([MS95], [Rab70]) *Jeżeli język drzew nieskończonych L jest rozpoznawany alternującym automatem Büchiego, to L jest też rozpoznawany niedeterministycznym automatem Büchiego – innymi słowy automaty Büchiego dopuszczają* dealternację.

[3] Z przyczyn technicznych należy tu rozważyć podpowiedź przejść automatu, zamiast stanów jak w Definicji 4.4. Spowodowane jest to faktem, że sam ciąg stanów może nie określać wartości liczników w trakcie biegu.

Ponadto, L daje się rozpoznawać alternującym automatem o słabym warunku parzystości (ozn. weak-ATA*), wtedy i tylko wtedy, gdy zarówno L jak i jego dopełnienie L^c są rozpoznawane niedeterministycznymi automatami Büchiego.*

Powyższe wyniki można interpretować następująco: warunek Büchiego na drzewach nieskończonych jest w pewnym sensie stabilny ze względu na dokładanie alternacji; natomiast języki weak-ATA stanowią klasę o symetrycznie ograniczonym niedeterminizmie (z punktu widzenia silnych warunków parzystości). Własności te pozwoliły przedstawić efektywną charakteryzację języków weak-ATA w obrębie klasy języków Büchiego, dając rozwiązanie Problemu 4.2 w tym przypadku:

Twierdzenie 4.16 ([CKLV13], alternatywny dowód w [Skr14a, Rozdział 2]) *Istnieje algorytm wczytujący niedeterministyczny automat Büchiego $\mathcal{A}$ dla drzew nieskończonych i rozstrzygający czy język* $\mathrm{L}(\mathcal{A})$ *daje się rozpoznawać jakimś automatem weak-ATA. W przypadku odpowiedzi pozytywnej, algorytm jest w stanie stworzyć taki weak-ATA.*

Jakkolwiek twierdzenie to charakteryzuje klasę języków weak-ATA, to nie dostarcza ono nowych informacji na temat ich złożoności w obrębie języków Büchiego. W szczególności, nie wynika z tych wyników odpowiedź na następującą hipotezę odniesioną do języków Büchiego:

Hipoteza 4.17 ([Sku93]) *Jeśli język regularny drzew nieskończonych jest borelowski, to daje się on rozpoznawać automatem weak-ATA.*

Hipoteza ta, podobnie jak argument użyty w Twierdzeniu 4.13, bazuje na intuicji, że języki borelowskie powinny odpowiadać ograniczonym formom niedeterminizmu. Stanowi ona odwrócenie znanego wyniku:

Twierdzenie 4.18 ([DM07], dowód obecny *de facto* w [Mos91]) *Każdy język weak-ATA jest borelowski.*

Główny wynik pracy [B] dostarcza pozytywnej odpowiedzi na powyższą hipotezę, w przypadku języków Büchiego:

★ **Twierdzenie 4.19** ([B]) *Dla danego niedeterministycznego automatu Büchiego $\mathcal{A}$ na drzewach nieskończonych, następujące warunki są równoważne:*

1. *język* $\mathrm{L}(\mathcal{A})$ *jest rozpoznawany przez automat weak-ATA,*
2. *język* $\mathrm{L}(\mathcal{A})$ *jest borelowski.*

Dodatkowo, problem rozstrzygania czy któryś z powyższych przypadków zachodzi jest EXPTIME-zupełny.

Struktura dowodu powyższego twierdzenia jest podobna do struktury dowodu Twierdzenia 2 z mojego doktoratu [Skr14a]: idea opiera się na odpowiednio zaprojektowanej grze, która pozwala w pewnym sensie wyrugować niedeterminizm wejściowego automatu. Celem gracza negatywnego w tej grze jest wykazać, że istnieją drzewa należące do danego języka, na których odpowiedni wariant sygnatury parzystości może być dowolnie duży. Dzięki odpowiednio dobranemu warunkowi wygrywania tej gry nie dość, że otrzymany algorytm ma optymalną złożoność czasową, to jeszcze wiedza wynikająca ze struktury tej gry pozwala stwierdzić, że języki Büchiego, które nie są weak-ATA muszą być $\mathbf{\Sigma}^1_1$-zupełne (w szczególności nieborelowskie).

Powyższe konstrukcje sugerują, że pewne problemy decyzyjne dla drzew nieskończonych mogą się upraszczać w przypadku warunków akceptacji dopuszczających dealternację. Znane przykłady takich warunków to, oprócz warunku Büchiego, również warunki *reachability* (czyli słaby indeks parzystości $(1, 2)$) oraz *safety* (słaby indeks $(0, 1)$). W tych dwóch przypadkach również potwierdzona jest powyższa intuicja, jak pokazuje poniższe twierdzenie:

Twierdzenie 4.20 ([Wal02, Sekcja 6], [Cav18, Sekcja 3.2] lub [JL02]) *Następujące warunki są równoważne dla języka regularnego drzew nieskończonych L:*

1. *L jest zbiorem domkniętym, czyli $\mathbf{\Pi}^0_1$ (odp.* otwartym*, czyli $\mathbf{\Sigma}^0_1$),*
2. *L jest rozpoznawany automatem alternującym safety (odp. reachability),*
3. *L jest rozpoznawany automatem niedeterministycznym safety (odp. reachability).*

Dodatkowo, problem rozstrzygania, czy któryś z powyższych przypadków zachodzi jest EXPTIME-zupełny.

Analogiczna intuicja dotycząca dealternacji okazała się kluczowa w uzyskaniu wyników z pracy [C]. Punktem wyjścia w tych badaniach była próba zrozumienia w sensie Problemu 4.2, które języki regularne drzew należą do klasy złożoności topologicznej $\mathbf{\Delta}^0_2 = \mathbf{\Sigma}^0_2 \cap \mathbf{\Pi}^0_2$ (wyniki [FM14] zawierają istotny błąd, patrz [BCPS18, Sekcja 14]). W trakcie badania tego problemu okazało się, że może być łatwiej pracować z niesymetryczną klasą $\mathbf{\Pi}^0_2$. Dalsza analiza kombinatoryczna problemu doprowadziła do wykazania nowego twierdzenia o dealternacji:

★ **Twierdzenie 4.21** ([C, Twierdzenie 13.]) *Jeżeli język drzew nieskończonych L jest rozpoznawany automatem alternującym o słabym indeksie parzystości* $(1, 3)$*, to L jest też rozpoznawany automatem niedeterministycznym o słabym indeksie parzystości* $(1, 3)$*.*

Powyższy wynik o dealternacji, w połączeniu z ideami topologicznymi z Twierdzenia 4.19 (patrz Stwierdzenie 11 w Sekcji 5 z [B]), pozwolił skonstruować grę, która radzi sobie z niedeterminizmem danego automatu. Doprowadziło to do wykazania następującego twierdzenia:

★ **Twierdzenie 4.22** ([C, Twierdzenia 2. i 13.]) *Następujące warunki są równoważne dla języka regularnego drzew nieskończonych:*

1. *L należy do klasy $\mathbf{\Sigma}^0_2$ złożoności topologicznej,*
2. *L jest rozpoznawany automatem alternującym o słabym indeksie* $(1, 3)$*,*
3. *L jest rozpoznawany automatem niedeterministycznym o słabym indeksie* $(1, 3)$*.*

Dodatkowo, istnieje algorytm, który rozstrzyga czy powyższe warunki zachodzą.

Klasa języków charakteryzowanych przez to twierdzenie jest relatywnie mała. Jednakże, jest to obecnie najszersza klasa dla której znana jest efektywna charakteryzacja, dopuszczająca na wejściu dowolny język regularny – wyniki Twierdzeń 4.16 i 4.19 są ograniczone do automatów Büchiego.

4.3.6 Niedeterminizm w przypadku drzew – języki jednoznaczne

Omówione do tej pory wyniki dotyczące drzew koncentrowały się na automatach deterministycznych (patrz Punkt a) Twierdzenia 4.6 i wyniki dotyczące automatów GFG), oraz niedeterministycznych i alternujących (Sekcja 4.3.5). W niniejszej sekcji zostają umówione wyniki z pracy [E] dotyczące automatów jednoznacznych[4].

Prosto jest podać przykłady pokazujące, że automaty deterministyczne na drzewach nieskończonych są istotnie słabsze od automatów niedeterministycznych. Zupełnie inaczej wygląda Problem 4.1 odniesiony do porównywania automatów jednoznacznych i niedeterministycznych. Pierwszy przykład języka regularnego drzew, który nie daje się rozpoznawać jednoznacznym automatem parzystości (czyli jest *niejednoznaczny*) podano w pracy [NW96] (wyniki rozszerzone w [CLNW10]). Podany tam przykład, to język drzew nad alfabetem $\{a, b\}$, które zawierają choć jedno wystąpienie litery a. Prostota tego języka nasuwa wniosek, że właściwie dowolna forma istotnego niedeterminizmu, musi prowadzić do języka niejednoznacznego. Z drugiej strony, okazuje się, że trudno jest udowodnić, że dany język istotnie jest niejednoznaczny. Na ten moment znane są jedynie dwa istotnie niezależne przykłady takich języków: język z pracy [NW96] oraz język drzew cienkich z pracy [BS13]. Dodatkowo, nie jest na ten moment znana żadna efektywna charakteryzacja (czyli odpowiedź na Problem 4.2) klasy języków jednoznacznych w obrębie wszystkich języków regularnych drzew; jedynie praca [BS13] przynosi pewne częściowe wyniki w tym kierunku.

Wymienione wyżej wyniki przedstawiają dość tajemniczy obraz sytuacji: z jednej strony wydaje się, że bardzo trudno zapewnić, żeby dany język był jednoznaczny; z drugiej strony znamy niewiele istotnie różnych przykładów języków niejednoznacznych. Jedną z hipotez rozważanych w oparciu o ten stan wiedzy było, że może języki jednoznaczne nie odbiegają znacząco od języków deterministycznych tj. również należą do klasy złożoności topologicznej $\mathbf{\Pi}^1_1$.

Pierwszy postęp w tym kierunku badań nastąpił w pracy [Hum12], gdzie wskazano przykład języka jednoznacznego, którego złożoność topologiczna wykracza poza klasę $\mathbf{\Pi}^1_1$. Późniejsze wyniki doprowadziły do następującego twierdzenia:

Twierdzenie 4.23 ([DFH15]) *Istnieją języki jednoznaczne o rosnącej złożoności topologicznej w obrębie drugiego poziomu hierarchii indeksu automatów alternujących.*

Istotnym elementem zaproponowanej konstrukcji jest fakt, że rozpatrywane tam języki są w istocie *podwójnie jednoznaczne*: zarówno dany język L jak i jego dopełnienie L^{c} są jednoznaczne.

Uzyskane w pracy [DFH15] wyniki, z jednej strony pokazują, że języki podwójnie jednoznaczne są zauważalnie bardziej złożone od deterministycznych, ale jednocześnie unaoczniają jak trudno może być znaleźć języki jednoznaczne poza drugim poziomem hierarchii indeksu automatów alternujących. Doprowadziło to do sformułowania następującej hipotezy (jest ona konsekwencją Hipotezy 5.27 w [Skr14a] w połączeniu z Twierdzeniem 26 z pracy [ISB16]):

Hipoteza 4.24 *Każdy język podwójnie jednoznaczny daje się rozpoznawać automatem alternującym indeksu* $(1, 3)$.

Hipoteza ta wciąż pozostaje otwarta, w szczególności nie udało się znaleźć żadnych nowych konstrukcji języków podwójnie jednoznacznych. Jednakże, w toku dalszych badań nad złożonością języków jednoznacznych, wykazano następujące twierdzenie. Występująca tu klasa

[4]Pojęcie automatów GFG dla drzew nie zostało nigdzie *explicite* wprowadzone. Jednak wydaje się, że przyjmując najbardziej naturalną definicję, otrzymywałoby się automaty GFG na słowach. W związku z tym nie rozważam tego modelu.

$\mathrm{Comp}(i,j)$ to, motywowana μ-rachunkiem, klasa języków dopuszczających zagnieżdżenie warunków indeksu (i,j) oraz $(i{+}1,j{+}1)$.

Twierdzenie 4.25 ([Skr14a, Rozdział 1], [MS16] oraz [MS18])

a) Jeżeli dany automat jednoznaczny jest indeksu postaci $(i,2j)$, to rozpoznawany przez niego język należy do klasy $\mathrm{Comp}(i{+}1,2j)$.

b) Analogiczny kolaps nie zachodzi dla automatów jednoznacznych o słabych warunkach parzystości.

Jedną z konsekwencji tego twierdzenia jest następujący wniosek:

Wniosek 4.26 *Każdy jednoznaczny automat Büchiego rozpoznaje język rozpoznawany przez weak-ATA.*

Ze względu na Twierdzenie 4.18, wniosek ten stanowi wzmocnienie wcześniejszego twierdzenia, uzyskanego metodami czysto topologicznymi:

Twierdzenie 4.27 ([FS09]) *Każdy jednoznaczny automat Büchiego rozpoznaje język borelowski.*

Warto przy tym zauważyć, że Wniosek 4.26 można niezależnie wydedukować z powyższego Twierdzenia 4.27, korzystając dodatkowo z Twierdzenia 4.19, mówiącego że borelowski język Büchiego musi być weak-ATA. Tym niemniej, konstrukcja podana w Twierdzeniu 4.25 jest bardziej bezpośrednia, dzięki czemu otrzymany automat ma wielomianowo-wiele stanów względem oryginalnego automatu (automat uzyskany z Twierdzenia 4.19 może być wykładniczo większy).

W momencie udowodnienia Punktu a) Twierdzenia 4.25 nie było jasne, dla jak wielu indeksów $(i,2j)$ wynik ten jest istotnie nietrywialny – było wciąż możliwe, że wszystkie języki jednoznaczne dają się rozpoznawać automatami ograniczonego indeksu. Jak się jednak okazuje tak nie jest:

★ **Twierdzenie 4.28** ([E]) *Istnieją języki jednoznaczne, leżące dowolnie wysoko w hierarchii indeksu automatów alternujących.*

Istotą konstrukcji jest nowa gra, pozwalająca porównywać sygnatury gier parzystości. Okazuje się, że by w jednoznaczny sposób rozpoznawać trudne języki regularne wystarczy umożliwić automatowi zgadnięcie *optymalnych* strategii ze względu na sygnatury parzystości. Można to widzieć jako analogię do dowodu Punktu b) Twierdzenia 4.7, gdzie należało zacząć od optymalnej w sensie sygnatur parzystości strategii w grze GFG, by zapewnić dobre własności kombinatoryczne badanego obiektu.

Twierdzenie 4.28 mówi, że języki jednoznaczne mogą być dowolnie *złożone* wśród wszystkich języków regularnych drzew. Oznacza to, że wcześniejsze intuicje mówiące o ich ograniczonej możliwości wykorzystania niedeterminizmu były błędne. Z drugiej strony, duża część technicznego ciężaru w [E] poświęcona jest zapewnieniu, by wszelkie wybory podejmowane niedeterministycznie przez konstruowany automat mogły być wykonane w sposób jednoznaczny. Innymi słowy, potrzeba wiele pracy, by zapewnić sobie, że dany rodzaj złożoności w sensie indeksu, będzie rozpoznawany w trybie jednoznacznym. Jednym ze skutków tych komplikacji jest fakt, że nie wiadomo czy da się zapewnić, by konstruowane języki były podwójnie jednoznaczne. Między innymi z tego powodu Hipoteza 4.24 pozostaje wciąż otwarta.

4.3.7 Podsumowanie

Wyniki wchodzące w skład mojego osiągnięcia naukowego doprowadziły do rozszerzenia stanu wiedzy dotyczącego pośrednich form niedeterminizmu. Udało się uzyskać postępy w obrębie rozważanych modeli automatów, zarówno w odniesieniu do nieskończonych słów jak i drzew.

Prace [A] i [D] dostarczyły dość kompletnego obrazu na temat automatów GFG, w odniesieniu do automatów deterministycznych, jednoznacznych i niedeterministycznych. Ze względu na związki z automatami Good-For-Trees, rezultaty te można odnieść do pewnych wariantów automatów deterministycznych na drzewach. Wyniki [F] dostarczają zrozumienia siły niedeterminizmu dla jednego z najprostszych modeli automatów z licznikami: niedeterminizm jest tu istotnie silniejszy od jakichkolwiek modeli maszyn jednoznacznych czy GFG.

Prace [B] i [C] dostarczyły nowych rozwiązań Problemu 4.2, opartych o metody growe i powiązane twierdzenia dealternacyjne. Wyniki [B] wskazują na podobną analogię do tej wynikającej z pracy [F]: języki borelowskie odpowiadają niepełnej sile niedeterminizmu, zaś języki o złożoności $\mathbf{\Sigma}^1_1$ odzwierciedlają jego pełną siłę. Wreszcie praca [E] dostarcza odpowiedzi na długo otwarty wariant Problemu 4.2: jak skomplikowane mogą być języki jednoznaczne w obrębie wszystkich języków regularnych drzew nieskończonych.

Metody dowodowe tych rezultatów opierają się na narzędziach kombinatoryki, topologii i teorii gier. Szczególnie obecne są tam gry z warunkami ω-regularnymi używane do charakteryzowania pewnych pojęć ([A], [B], [C], [E]). Unaoczniona jest też rola sygnatur parzystości w wyborze *optymalnego* świadka ([A], [E]) lub szacowaniu złożoności języka ([B]). Istotną rolę odgrywa też pojęcie topologicznej trudności rozważanego języka ([B], [C], [E], [F] oraz *implicite* w [D]).

Jakkolwiek wiele pytań dotyczących niedeterminizmu wciąż czeka na rozwiązanie, to przedstawione osiągnięcie doprowadziło do ostatecznego zamknięcia kilku ważnych, badanych uprzednio problemów otwartych: Twierdzenie 4.7 daje precyzyjne szacowania kosztu determinizacji automatów GFG; Twierdzenie 4.13 ustala stopień niedeterminizmu automatów BMC z jednym ślepym licznikiem; wreszcie Twierdzenie 4.28 pozwala zrozumieć złożoność języków jednoznacznych w obrębie wszystkich języków regularnych drzew.

5 Omówienie pozostałych osiągnięć naukowo–badawczych

Wyniki wchodzące w skład mojego osiągnięcia naukowego wpasowują się w kontekst wcześniejszych badań w tym kierunku. W szczególności, następujące prace mojego współautorstwa, jakkolwiek nie wchodzą w skład mojego osiągnięcia, to są z nim powiązane i zostały już wspomniane powyżej:

[BKKS13] Udi Boker, Denis Kuperberg, Orna Kupferman, and Michał Skrzypczak. Nondeterminism in the presence of a diverse or unknown future. In *ICALP (2)*, pages 89–100, 2013.

Praca pokazuje, że klasa automatów GFG jest istotnie szersza od klasy DBP. Dodatkowo, pokazane jest, że automat parzystości jest GFG wtedy i tylko wtedy, gdy jest Good-For-Trees.

[BS13] Marcin Bilkowski and Michał Skrzypczak. Unambiguity and uniformization problems on infinite trees. In *CSL*, volume 23 of *LIPIcs*, pages 81–100, 2013.

Praca wprowadza nową hipotezę dotyczącą możliwości zdefiniowania w MSO funkcji wyboru na tzw. drzewach cienkich. Przedstawione są inne, równoważne sformułowania tej hipotezy. Podana jest też, oparta o zaproponowaną hipotezę, efektywna charakteryzacja języków podwójnie jednoznacznych na pełnych, nieskończonych drzewach binarnych. Ponadto zaproponowany jest istotnie nowy przykład niejednoznacznego języka regularnego.

[FS14] Olivier Finkel and Michał Skrzypczak. On the topological complexity of ω-languages of non-deterministic Petri nets. *Information Processing Letters*, 114(5):229–233, 2014.

Głównym wynikiem tej pracy jest cytowane powyżej Twierdzenie 4.12.

[MS16] Henryk Michalewski and Michał Skrzypczak. Unambiguous Büchi is weak. In *DLT*, pages 319–331, 2016.

Głównym wynikiem tej pracy jest cytowany powyżej Punkt a) Twierdzenia 4.25.

[MS18] Henryk Michalewski and Michał Skrzypczak. On the strength of unambiguous tree automata. *International Journal of Foundations of Computer Science*, 29(5):911–933, 2018.

Jest to czasopismowa wersja [MS16], rozszerzająca podane wyniki o Punkt b) Twierdzenia 4.25 oraz nową procedurę separacji w stylu [AS05], dla niedeterministycznych automatów ze słabymi warunkami parzystości.

Dodatkowo, wiele z wyników omawianych w dalszej części tego dokumentu znalazło się z moim doktoracie, którego wersja książkowa została opublikowana w 2016 roku:

[Skr16] Michał Skrzypczak. *Descriptive Set Theoretic Methods in Automata Theory – Decidability and Topological Complexity*, volume 9802 of *Lecture Notes in Computer Science*. Springer, 2016.

5.1 Własności miarowe i kategoryjne języków regularnych

Jednym z żywych tematów moich badań była analiza własności motywowanych topologią i teorią miary w odniesieniu do języków regularnych słów i drzew nieskończonych. Prace badające tę tematykę, to:

[BNR$^+$10] Mikołaj Bojańczyk, Damian Niwiński, Alexander Rabinovich, Adam Radziwończyk-Syta, and Michał Skrzypczak. On the Borel complexity of MSO definable sets of branches. *Fundamenta Informaticae*, 98(4):337–349, 2010.

Celem tej pracy jest zbadanie, jaką złożoność topologiczną mają zbiory gałęzi pełnego drzewa binarnego, które mogą być zdefiniowane w oparciu o ustalone parametry monadyczne przez formułę MSO (lub równoważnie niedeterministyczny automat parzystości). Okazuje się, że zbiory te, to dokładnie boolowskie kombinacje zbiorów $\mathbf{\Sigma}^0_2$, czyli zbiory o złożoności topologicznej odpowiadającej warunkom parzystości.

[Skr13] Michał Skrzypczak. Topological extension of parity automata. *Information and Computation*, 228:16–27, 2013.

Praca ta rozbudowuje wyniki [BNR+10] o dokładniejszą analizę zależności pomiędzy warunkami parzystości a boolowskimi kombinacjami zbiorów $\mathbf{\Sigma}^0_2$. Ostateczny wynik to charakteryzacja zbiorów z pierwszych trzech poziomów hierarchii borelowskiej (aż do $\mathbf{\Sigma}^0_3$) w oparciu o odpowiednio dobrane warianty warunków parzystości.

[GMMS14] Tomasz Gogacz, Henryk Michalewski, Matteo Mio, and Michał Skrzypczak. Measure properties of game tree languages. In *MFCS*, pages 303–314, 2014.

Praca pokazuje, że wszystkie języki regularne drzew nieskończonych są uniwersalnie mierzalne. Kluczową częścią dowodu jest wskazanie na zależności pomiędzy językami $W_{i,j}$, a klasą tzw. $\mathcal{R}$-zbiorów wprowadzoną przez Kołmogorowa [Kol28].

[GMMS17] Tomasz Gogacz, Henryk Michalewski, Matteo Mio, and Michał Skrzypczak. Measure properties of regular sets of trees. *Information and Computation*, 256:108–130, 2017.

Jest to pełna czasopismowa wersja [GMMS14]. Oprócz uszczegółowienia wszystkich dowodów z [GMMS14], praca ta wykazuje dodatkowo, że stratyfikacja języków $W_{i,j}$, w oparciu o odpowiednio zdefiniowane sygnatury parzystości, jest ciągła ze względu na miarę. W połączeniu z mierzalnością, pozwala to w pełni wyeliminować potrzebę stosowania aksjomatu Martina z wyników dotyczących semantyki growej dla probabilistycznego μ-rachunku [Mio12].

[MSM18] Matteo Mio, Michał Skrzypczak, and Henryk Michalewski. Monadic second order logic with measure and category quantifiers. *Logical Methods in Computer Science*, 14(2):1–29, 2018.

Praca wprowadza i bada pewne dodatkowe kwantyfikatory wyrażające *generyczność* danego zbioru lub gałęzi. Rozważane pojęcia *generyczności* są oparte o miarę (zbiór pełnej miary) i kategorię (dopełnienie zbioru pierwszej kategorii Baire'a).

5.2 Automaty growe i gry rozgałęziające

Kolejny temat moich badań dotyczy pojęcia automatów growych oraz gier rozgałęziających. Automat growy może być widziany jako samo-dualizacja automatu deterministycznego na drzewach nieskończonych: każde przejście tego automatu jest albo uniwersalne (i wysyła dwa stany do dwojga dzieci danego wierzchołka), albo egzystencjalne (i niedeterministycznie zgaduje, w którym kierunku wysłać następny stan). W ramach następującej serii prac udało nam się rozwiązać wszystkie problemy indeksu dla tej klasy automatów, dodatkowo przedstawiając efektywną charakteryzację rodziny języków rozpoznawanych w tym trybie:

[FMS13] Alessandro Facchini, Filip Murlak, and Michał Skrzypczak. Rabin-Mostowski index problem: A step beyond deterministic automata. In *LICS*, pages 499–508, 2013.

[FMS15] Alessandro Facchini, Filip Murlak, and Michał Skrzypczak. On the weak index problem for game automata. In *WoLLIC*, pages 93–108, 2015.

[FMS16] Alessandro Facchini, Filip Murlak, and Michał Skrzypczak. Index problems for game automata. *ACM Transactions on Computational Logic*, 17(4):24:1–24:38, 2016.

Częściowo powiązane z tymi wynikami są badania nad tzw. grami rozgałęziającymi, gdzie pełen zapis pojedynczej rozgrywki ma kształt nieskończonego drzewa. Ze względu na fakt niepełnej informacji, jaką posiadają gracze w takiej grze, gry te nie muszą być zdeterminowane. W ramach poniższej pracy badaliśmy ich poziom determinacji w zależności od komplikacji warunku wygrywania, oraz dopuszczonej klasy strategii. Dodatkowo analizowaliśmy rozstrzygalność problemu obliczania wartości takiej gry.

[PS16] Marcin Przybyłko and Michał Skrzypczak. On the complexity of branching games with regular conditions. In *MFCS*, pages 78:1–78:14, 2016.

5.3 Rozstrzygalność MSO

Poniższe dwie prace stanowią dość kompletną analizę zależności pomiędzy logiką MSO a automatami w odniesieniu do klasy tzw. drzew cienkich. Drzewo cienkie to takie częściowe drzewo binarne, które ma jedynie przeliczalnie wiele gałęzi nieskończonych. Okazuje się, że nie dość, że teoria MSO takich drzew jest rozstrzygalna, to stanowią one dość naturalny krok pośredni pomiędzy nieskończonymi słowami a drzewami.

[BIS13] Mikołaj Bojańczyk, Tomasz Idziaszek, and Michał Skrzypczak. Regular languages of thin trees. In *STACS*, volume 20 of *LIPIcs*, pages 562–573, 2013.

[ISB16] Tomasz Idziaszek, Michał Skrzypczak, and Mikołaj Bojańczyk. Regular languages of thin trees. *Theory of Computing Systems*, 58(4):614–663, 2016.

Praca stanowi rozszerzoną, czasopismową wersję [BIS13].

W ramach odrębnego nurtu badawczego, kontynuując wyniki [KM16], badaliśmy siłę aksjomatyczną potrzebną do wykazania twierdzenia Büchiego [Büc62], mówiącego o rozstrzygalności MSO na słowach nieskończonych. Główne wyniki tej pracy pokazują, że rozstrzygalność MSO na słowach nieskończonych jest zasadniczo równoważna zasadzie indukcji dla Σ^0_2-formuł.

[KMPS16] Leszek Aleksander Kołodziejczyk, Henryk Michalewski, Pierre Pradic, and Michał Skrzypczak. The logical strength of Büchi's decidability theorem. In *CSL*, pages 36:1–36:16, 2016.

5.4 Ilościowe rozszerzenia pojęcia regularności

Ostatnia grupa prac koncentruje się na modelach rozszerzających języki regularne o jakieś własności ilościowe. Większość tych wyników dotyczy rozszerzeń mówiących o ograniczoności występowania pewnych zjawisk: studiowany jest kwantyfikator U (wprowadzony w [Boj04]) oraz automaty ωB, ωS i ωBS (studiowane w [BC06]).

[HST10] Szczepan Hummel, Michał Skrzypczak, and Szymon Toruńczyk. On the topological complexity of MSO+U and related automata models. In *MFCS*, pages 429–440, 2010.

Praca bada siłę wyrazu logiki MSO rozszerzonej o tzw. kwantyfikator nieograniczoności U. Okazuje się, że tak rozszerzona logika (ozn. MSO+U) może definiować języki nieborelowskie. Wyklucza to możliwość chwytania tej logiki przez jakąkolwiek klasę borelowskich automatów niedeterministycznych. Dodatkowo, praca podaje dokładne szacowania złożoności topologicznej dla automatów niedeterministycznych z warunkami ωB, ωS i ωBS; oraz pokazuje języki złożone topologicznie, które daje się rozpoznawać automatami alternującymi z warunkiem ωBS.

[HS12] Szczepan Hummel and Michał Skrzypczak. The topological complexity of MSO+U and related automata models. *Fundamenta Informaticae*, 119(1):87–111, 2012.

Jest to rozszerzona, czasopismowa wersja [HST10]. W pracy tej udało się rozszerzyć konstrukcję nieborelowskiego języka definiowalnego w MSO+U do całej rodziny języków, sięgających na wszystkie skończone poziomy hierarchii rzutowej. Pozwoliło to wykluczyć możliwość chwytania MSO+U przez jakiekolwiek borelowskie automaty alternujące.

[BGMS14] Mikołaj Bojańczyk, Tomasz Gogacz, Henryk Michalewski, and Michał Skrzypczak. On the decidability of MSO+U on infinite trees. In *ICALP (2)*, pages 50–61, 2014.

Praca podaje dowód, że przy założeniu aksjomatu Gödla [Göd39] o konstruowalności uniwersum (ozn. V=L), logika MSO+U jest nierozstrzygalna na drzewach nieskończonych. Dowód tego faktu jest oparty o języki trudne topologicznie z [HS12] oraz metody [She75] użyte do wykazania nierozstrzygalności MSO($\mathbb{R}$). Jakiś czas po opublikowaniu tej pracy, znaleziono bezpośredni dowód nierozstrzygalności MSO+U na słowach nieskończonych [BPT16].

[Skr14b] Michał Skrzypczak. Separation property for wB- and wS-regular languages. *Logical Methods in Computer Science*, 10(1):1–20, 2014.

Praca bada własność separacji i regularności dla języków słów nieskończonych rozpoznawanych przez automaty z warunkami ωB i ωS. Główny wynik mówi, że dwa rozłączne języki rozpoznawane automatami typu ωB (odpowiednio ωS) dają się odseparować od siebie językiem regularnym.

[FHKS15] Nathanaël Fijalkow, Florian Horn, Denis Kuperberg, and Michał Skrzypczak. Trading bounds for memory in games with counters. In *ICALP (2)*, pages 197–208, 2015.

Praca analizuje własność determinacji ze skończoną pamięcią (ang. finite memory determinacy) dla rodziny gier parzystości z warunkami ograniczoności B. Główne wyniki mówią, że taka determinacja w ogólności nie zachodzi, natomiast ma ona miejsce na arenach pochodzących od drzew cienkich.

Innym ilościowym rozszerzeniem pojęcia regularności jakie było przeze mnie badane są automaty probabilistyczne. Następująca krótka notatka podaje bardzo prosty przykład nieregularnego języka słów skończonych, który daje się rozpoznawać przez takie automaty.

[FS15] Nathanaël Fijalkow and Michał Skrzypczak. Irregular behaviours for probabilistic automata. In *RP*, pages 33–36, 2015.

Literatura

[AS05] André Arnold and Luigi Santocanale. Ambiguous classes in μ-calculi hierarchies. *Theoretical Computer Science*, 333(1–2):265–296, 2005.

[BC06] Mikołaj Bojańczyk and Thomas Colcombet. Bounds in ω-regularity. In *LICS*, pages 285–296, 2006.

[BCPS18] Mikołaj Bojańczyk, Filippo Cavallari, Thomas Place, and Michał Skrzypczak. Regular tree languages in low levels of Wadge hierarchy. *CoRR*, abs/1806.02041, 2018. Submitted to a journal.

[BGMS14] Mikołaj Bojańczyk, Tomasz Gogacz, Henryk Michalewski, and Michał Skrzypczak. On the decidability of MSO+U on infinite trees. In *ICALP (2)*, pages 50–61, 2014.

[BIS13] Mikołaj Bojańczyk, Tomasz Idziaszek, and Michał Skrzypczak. Regular languages of thin trees. In *STACS*, volume 20 of *LIPIcs*, pages 562–573, 2013.

[BKKS13] Udi Boker, Denis Kuperberg, Orna Kupferman, and Michał Skrzypczak. Nondeterminism in the presence of a diverse or unknown future. In *ICALP (2)*, pages 89–100, 2013.

[BNR^{+}10] Mikołaj Bojańczyk, Damian Niwiński, Alexander Rabinovich, Adam Radziwończyk-Syta, and Michał Skrzypczak. On the Borel complexity of MSO definable sets of branches. *Fundamenta Informaticae*, 98(4):337–349, 2010.

[Boj04] Mikołaj Bojańczyk. A bounding quantifier. In *CSL*, pages 41–55, 2004.

[BPT16] Mikołaj Bojańczyk, Paweł Parys, and Szymon Toruńczyk. The MSO+U theory of (N, $<$) is undecidable. In *STACS*, pages 21:1–21:8, 2016.

[BS13] Marcin Bilkowski and Michał Skrzypczak. Unambiguity and uniformization problems on infinite trees. In *CSL*, volume 23 of *LIPIcs*, pages 81–100, 2013.

[Büc62] Julius Richard Büchi. On a decision method in restricted second-order arithmetic. In *Proc. 1960 Int. Congr. for Logic, Methodology and Philosophy of Science*, pages 1–11, 1962.

[Cav18] Filippo Cavallari. *Regular tree languages in the first two levels of the Borel hierarchy*. PhD thesis, HEC Lausanne (UNIL) and University of Turin, June 2018.

[CKLV13] Thomas Colcombet, Denis Kuperberg, Christof Löding, and Michael Vanden Boom. Deciding the weak definability of Büchi definable tree languages. In *CSL*, pages 215–230, 2013.

[CL10] Thomas Colcombet and Christof Löding. Regular cost functions over finite trees. In *LICS*, pages 70–79, 2010.

[CLNW10] Arnaud Carayol, Christof Löding, Damian Niwiński, and Igor Walukiewicz. Choice functions and well-orderings over the infinite binary tree. *Central European Journal of Mathematics*, 8:662–682, 2010.

[Col13] Thomas Colcombet. Fonctions régulières de coût. Habilitation thesis, Université Paris Diderot—Paris 7, 2013.

[DFH15] Jacques Duparc, Kevin Fournier, and Szczepan Hummel. On unambiguous regular tree languages of index (0, 2). In *CSL*, pages 534–548, 2015.

[DM07] Jacques Duparc and Filip Murlak. On the topological complexity of weakly recognizable tree languages. In *FCT*, pages 261–273, 2007.

[FHKS15] Nathanaël Fijalkow, Florian Horn, Denis Kuperberg, and Michał Skrzypczak. Trading bounds for memory in games with counters. In *ICALP (2)*, pages 197–208, 2015.

[FM14] Alessandro Facchini and Henryk Michalewski. Deciding the Borel complexity of regular tree languages. In *CiE*, pages 163–172, 2014.

[FMS13] Alessandro Facchini, Filip Murlak, and Michał Skrzypczak. Rabin-Mostowski index problem: A step beyond deterministic automata. In *LICS*, pages 499–508, 2013.

[FMS15] Alessandro Facchini, Filip Murlak, and Michał Skrzypczak. On the weak index problem for game automata. In *WoLLIC*, pages 93–108, 2015.

[FMS16] Alessandro Facchini, Filip Murlak, and Michał Skrzypczak. Index problems for game automata. *ACM Transactions on Computational Logic*, 17(4):24:1–24:38, 2016.

[FS09] Olivier Finkel and Pierre Simonnet. On recognizable tree languages beyond the Borel hierarchy. *Fundamemta Informaticae*, 95(2–3):287–303, 2009.

[FS14] Olivier Finkel and Michał Skrzypczak. On the topological complexity of ω-languages of non-deterministic Petri nets. *Information Processing Letters*, 114(5):229–233, 2014.

[FS15] Nathanaël Fijalkow and Michał Skrzypczak. Irregular behaviours for probabilistic automata. In *RP*, pages 33–36, 2015.

[GMMS14] Tomasz Gogacz, Henryk Michalewski, Matteo Mio, and Michał Skrzypczak. Measure properties of game tree languages. In *MFCS*, pages 303–314, 2014.

[GMMS17] Tomasz Gogacz, Henryk Michalewski, Matteo Mio, and Michał Skrzypczak. Measure properties of regular sets of trees. *Information and Computation*, 256:108–130, 2017.

[Göd39] Kurt Gödel. Consistency-Proof for the Generalized Continuum-Hypothesis. *Proceedings of the National Academy of Sciences of the United States of America*, 25(4):220–224, 1939.

[HP06] Thomas A. Henzinger and Nir Piterman. Solving games without determinization. In *CSL*, LNCS, pages 395–410. Springer, 2006.

[HS12] Szczepan Hummel and Michał Skrzypczak. The topological complexity of MSO+U and related automata models. *Fundamenta Informaticae*, 119(1):87–111, 2012.

[HST10] Szczepan Hummel, Michał Skrzypczak, and Szymon Toruńczyk. On the topological complexity of MSO+U and related automata models. In *MFCS*, pages 429–440, 2010.

[Hum12] Szczepan Hummel. Unambiguous tree languages are topologically harder than deterministic ones. In *GandALF*, pages 247–260, 2012.

[ISB16] Tomasz Idziaszek, Michał Skrzypczak, and Mikołaj Bojańczyk. Regular languages of thin trees. *Theory of Computing Systems*, 58(4):614–663, 2016.

[JL02] David Janin and Giacomo Lenzi. On the logical definability of topologically closed recognizable languages of infinite trees. *Computers and Artificial Intelligence*, 21(3), 2002.

[Kec95] Alexander Kechris. *Classical descriptive set theory.* Springer-Verlag, New York, 1995.

[KM16] Leszek Aleksander Kołodziejczyk and Henryk Michalewski. How unprovable is Rabin's decidability theorem? In *LICS*, pages 788–797, 2016.

[KMM06] Orna Kupferman, Gila Morgenstern, and Aniello Murano. Typeness for ω-regular automata. *IJFCS*, 17(4):869–884, 2006.

[KMPS16] Leszek Aleksander Kołodziejczyk, Henryk Michalewski, Pierre Pradic, and Michał Skrzypczak. The logical strength of Büchi's decidability theorem. In *CSL*, pages 36:1–36:16, 2016.

[Kol28] Andrey Nikolaevich Kolmogorov. Operations sur des ensembles (in Russian, summary in French). *Matematicheskii Sbornik*, 35:415–422, 1928.

[KPB94] Sriram C. Krishnan, Anuj Puri, and Robert K. Brayton. Deterministic ω-automata vis-a-vis deterministic Büchi automata. In *Algorithms and Computations*, volume 834 of *LNCS*, pages 378–386. Springer, 1994.

[Mio12] Matteo Mio. On the equivalence of game and denotational semantics for the probabilistic mu-calculus. *Logical Methods in Computer Science*, 8(2), 2012.

[Mos91] Andrzej W. Mostowski. Hierarchies of weak automata and weak monadic formulas. *Theoretical Computer Science*, 83(2):323–335, 1991.

[MS95] David E. Muller and Paul E. Schupp. Simulating alternating tree automata by nondeterministic automata: New results and new proofs of the theorems of Rabin, McNaughton and Safra. *Theoretical Computer Science*, 141(1&2):69–107, 1995.

[MS16] Henryk Michalewski and Michał Skrzypczak. Unambiguous Büchi is weak. In *DLT*, pages 319–331, 2016.

[MS18] Henryk Michalewski and Michał Skrzypczak. On the strength of unambiguous tree automata. *International Journal of Foundations of Computer Science*, 29(5):911–933, 2018.

[MSM18] Matteo Mio, Michał Skrzypczak, and Henryk Michalewski. Monadic second order logic with measure and category quantifiers. *Logical Methods in Computer Science*, 14(2):1–29, 2018.

[NW96] Damian Niwiński and Igor Walukiewicz. Ambiguity problem for automata on infinite trees. unpublished, 1996.

[PS16] Marcin Przybyłko and Michał Skrzypczak. On the complexity of branching games with regular conditions. In *MFCS*, pages 78:1–78:14, 2016.

[Rab70] Michael Oser Rabin. Weakly definable relations and special automata. In *Proceedings of the Symposium on Mathematical Logic and Foundations of Set Theory*, pages 1–23. North-Holland, 1970.

[SE89] Robert S. Streett and E. Allen Emerson. An automata theoretic decision procedure for the propositional mu-calculus. *Information and Computation*, 81(3):249–264, 1989.

[She75] Saharon Shelah. The monadic theory of order. *The Annals of Mathematics*, 102(3):379–419, 1975.

[SI85] Richard Edwin Stearns and Harry B. Hunt III. On the equivalence and containment problems for unambiguous regular expressions, regular grammars and finite automata. *SIAM Journal on Computing*, 14(3):598–611, 1985.

[Skr13] Michał Skrzypczak. Topological extension of parity automata. *Information and Computation*, 228:16–27, 2013.

[Skr14a] Michał Skrzypczak. *Descriptive set theoretic methods in automata theory*. PhD thesis, University of Warsaw, 2014.

[Skr14b] Michał Skrzypczak. Separation property for wB- and wS-regular languages. *Logical Methods in Computer Science*, 10(1):1–20, 2014.

[Skr16] Michał Skrzypczak. *Descriptive Set Theoretic Methods in Automata Theory – Decidability and Topological Complexity*, volume 9802 of *Lecture Notes in Computer Science*. Springer, 2016.

[Sku93] Jerzy Skurczyński. The Borel hierarchy is infinite in the class of regular sets of trees. *Theoretical Computer Science*, 112(2):413–418, 1993.

[SM73] Larry Stockmeyer and Albert R. Meyer. Word problems requiring exponential time. In *ACM Symposium on Theory of Computing*, pages 1–9, 1973.

[Tho96] Wolfgang Thomas. Languages, automata, and logic. In *Handbook of Formal Languages*, pages 389–455. Springer, 1996.

[Wal96] Igor Walukiewicz. Pushdown processes: Games and model checking. In Rajeev Alur and Thomas A. Henzinger, editors, *Computer Aided Verification*, pages 62–74. Springer, 1996.

[Wal02] Igor Walukiewicz. Deciding low levels of tree-automata hierarchy. In *WoLLIC*, volume 67, pages 61–75, 2002.

....
(podpis kandydata)