

Autoreferat

24 września 2024

I. IMIĘ I NAZWISKO

Jacek Sroka

II. POSIADANE DYPLOMY, STOPNIE NAUKOWE LUB ARTYSTYCZNE – Z PODANIEM PODMIOTU NADAJĄCEGO STOPIEŃ, ROKU ICH UZYSKANIA ORAZ TYTUŁU ROZPRAWY DOKTORSKIEJ

- Stopień doktora nauk matematycznych (specjalność informatyka) w 2009 na Wydziale Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego, tytuł rozprawy doktorskiej: *Models and Languages for Specification of Collection-oriented Scientific Workflows*.
- Tytuł magistra z informatyki w 2003 na Wydziale Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego, praca magisterska: *Uniwersytecki System Obsługi Studiów. Moduł akademiki*.
- Tytuł licencjata z matematyki w 2002 na Wydziale Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego, praca licencjacka: *Analiza algorytmów genetycznych w świetle teorii schematów*.

III. INFORMACJA O DOTYCHCZASOWYM ZATRUDNIENIU W JEDNOSTKACH NAUKOWYCH LUB ARTYSTYCZNYCH

- 2009–obecnie adiunkt w Instytucie Informatyki na Wydziale Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego.
- 2018–2022 szef zespołu tworzącego system Wsparcia Organów Wyborczych w Krajowym Biurze Wyborczym (obsługa wyborów: Wybory Prezydenta Rzeczypospolitej Polskiej, Wybory do Sejmu i Senatu Rzeczypospolitej Polskiej, Wybory do Parlamentu Europejskiego, Wybory Samorządowe); ekspert oraz programista w Krajowym Biurze Wyborczym.
- 2007–2009 asystent w Instytucie Informatyki na Wydziale Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego.
- 2003–2007 studia doktoranckie w Instytucie Informatyki na Wydziale Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego.
- 1998–2003 jednoczesne studia z informatyki i matematyki na Wydziale Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego.

IV. 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.)

Tytuł: Przepływy prac: zastosowania w bioinformatyce oraz modelowanie przy pomocy sieci Petriego.

Publikacje składające się na osiągnięcie naukowe

- [1] Kamil Kedzia, Wojtek Ptak, Jacek Sroka, Andrzej M. Kierzek: Simulation of multicellular populations with Petri nets and genome scale intracellular networks. **Sci. Comput. Program.** 157: 3-16 (2018), <https://doi.org/10.1016/j.scico.2017.10.002>.
Jest to rozszerzona wersja pracy konferencyjnej [A.3], która została wyróżniona nagrodą za **najlepszą pracę** na Conference on Applications and Theory of Petri Nets and Concurrency.
- [2] Jacek Sroka, Jan Hidders: On Determining the AND-OR Hierarchy in Workflow Nets, **Fundamenta Informaticae**, Volume 151 (1), 2017, Pages 95-123, <https://doi.org/10.3233/FI-2017-1599>.
- [3] Jacek Sroka, Piotr Chrzastowski-Wachtel, Jan Hidders: On Generating Hierarchical Workflow Nets and their Extensions and Verifying Hierarchicality, **Fundamenta Informaticae**, Volume 141 (4), 2015, Pages 367-398, <https://doi.org/10.3233/FI-2015-1280>.
- [4] Jacek Sroka, Jan Hidders: On generating *-sound nets with substitution, **Information Systems**, Volume 40, 2014, Pages 32-46, ISSN 0306-4379, <https://doi.org/10.1016/j.is.2013.10.004>.
- [5] Jacek Sroka, Łukasz Bieniasz-Krzywiec, Szymon Gwóźdź, Dariusz Leniowski, Jakub Łącki, Mateusz Markowski, Claudio Avignone-Rossa, Michael E Bushell, Johnjoe McFadden and Andrzej M Kierzek: ACORN: A grid computing system for constraint based modeling and visualization of the genome scale metabolic reaction networks via a web interface. **BMC Bioinformatics**, 12, 196 (2011). <https://doi.org/10.1186/1471-2105-12-196>.
Praca otrzymała w czasopiśmie status *Highly accessed*.
- [6] Jacek Sroka, Jan Hidders, Paolo Missier, Carole Goble. (2010). A formal semantics for the Taverna 2 workflow model. **Journal of Computer and System Sciences**, 76, 490-508, <https://doi.org/10.1016/j.jcss.2009.11.009>.

Omówienie wyników

A. Wprowadzenie

Osiągnięcie naukowe składa się z grupy artykułów związanych z modelowaniem przepływów pracy (ang. workflow modelling). Moje zainteresowanie tematem zaczęło się od zastosowań w bioinformatyce, gdzie metody informatyczne są używane do modelowania i symulacji złożonych systemów biologicznych. Najpierw pomogłem wprowadzić metody formalne do dobrze ugruntowanych projektów, a następnie stworzyłem nowe systemy, które przyczyniły się do rozwoju biologii. Zastosowałem również techniki Big Data, aby umożliwić realizację tych działań na większą skalę. Zainspirowany

praktycznymi problemami wniosłem wkład w dziedzinie modelowania przepływów pracy za pomocą sieci Petriego oraz analizy ich właściwości.

W [6] zdefiniowaliśmy formalną semantykę dla systemu Taverna 2, narzędzia szeroko stosowanego w praktyce do modelowania i wykonywania przepływów prac w biologii. Była to wspólna praca z autorami Taverny, która była niezbędna do umożliwienia rozumowania, czy dwie specyfikacje przepływu prac w Tavernie są równoważne oraz czy dana właściwość może być wyrażona w języku przepływu prac Taverna 2. Następnie, w [5] stworzyliśmy system ACORN, który pozwala na symulację metabolizmu w rekonstrukcjach sieci metabolicznych komórki w skali genomu. W tym artykule nie stosujemy przepływów prac, ale ACORN został później rozszerzony do systemu MUFINS z modelem regulacji genów opartym na sieciach Petriego. MUFINS został opublikowany w jednym z czasopism Nature [A.2]. Artykuł zawierał eksperymentalną walidację prognozy, demonstrując zastosowanie naszego oprogramowania w iteracyjnym cyklu generowania hipotez, ich weryfikacji oraz udoskonalania modelu. Ten kierunek badań kontynuowaliśmy w [1], stosując techniki Big Data, aby umożliwić symulację ogromnych populacji współdziałających komórek, gdzie część regulacyjna każdej komórki była wciąż modelowana jako sieć Petriego. Ze względu na złożoność modelowanych systemów biologicznych użytkownicy potrzebowali narzędzi do "debugowania modeli", które opracowaliśmy w [A.4] i [D.4]. Ze względu na obliczeniową trudność tego problemu zainteresowałem się formalnymi metodami sprawdzania, czy specyfikacja sieci Petriego ma pożądane właściwości oraz jak je projektować, aby te właściwości były zagwarantowane.

Tego typu potrzeby były jedną z motywacji powstania trzech artykułów skoncentrowanych na podklasie sieci Petriego zwanej sieciami przepływu pracy (ang. WF nets) oraz metodach hierarchicznych. W [4] rozwinęliśmy metodę hierarchicznej konstrukcji poprawnych sieci przepływów prac przez podstawianie sieci z wieloma wejściami i wyjściami, co prowadzi do klasy sieci zwanych sieciami AND-OR. Wykazaliśmy, że ta metoda jest poprawna i że wyniki sieci AND-OR uogólniają klasy sieci, które mogą być uzyskane przy pomocy innych podejść hierarchicznych. Dodatkowo, wprowadziliśmy nowe pojęcie poprawności, które jest zachowane przez nasz uogólniony typ podstawienia. Wykazaliśmy, że jest to w pewnym sensie optymalne pojęcie poprawności do generowania poprawnych sieci przez przedstawiony typ podstawienia. Następnie, w [3], wykazaliśmy, że sieci AND-OR uogólniają klasy sieci, które w innych podejściach mogą być generowane jedynie za pomocą niehierarchicznych rozszerzeń. Przy pomocy translacji do sieci AND-OR przeprowadziliśmy formalny dowód poprawności takich rozszerzeń. Następnie, w [2], komplementarnie do podstawienia, zdefiniowaliśmy pojęcie redukcji, gdzie sieć przepływu prac jest przekształcana w mniejszą sieć przez iteracyjne zwijanie niektórych dobrze uformowanych podsieci do pojedynczych węzłów, aż do momentu, gdy nie są możliwe dalsze takie zwijania. Wykazaliśmy, że wynik redukcji podsieci AND-OR jest niezależny od kolejności wyboru redukowanych podsieci. Oznacza to, że pomimo dowolności wyboru zwijanych podsieci, ostateczny

wynik redukcji jest zawsze taki sam z dokładnością do wyboru tożsamości węzłów. Na podstawie tego wyniku opracowaliśmy algorytm działający w czasie wielomianowym, który oblicza ten unikalny wynik redukcji. Na koniec, pokazaliśmy, że ten algorytm może być użyty do weryfikacji, czy sieć przepływu prac jest siecią AND-OR.

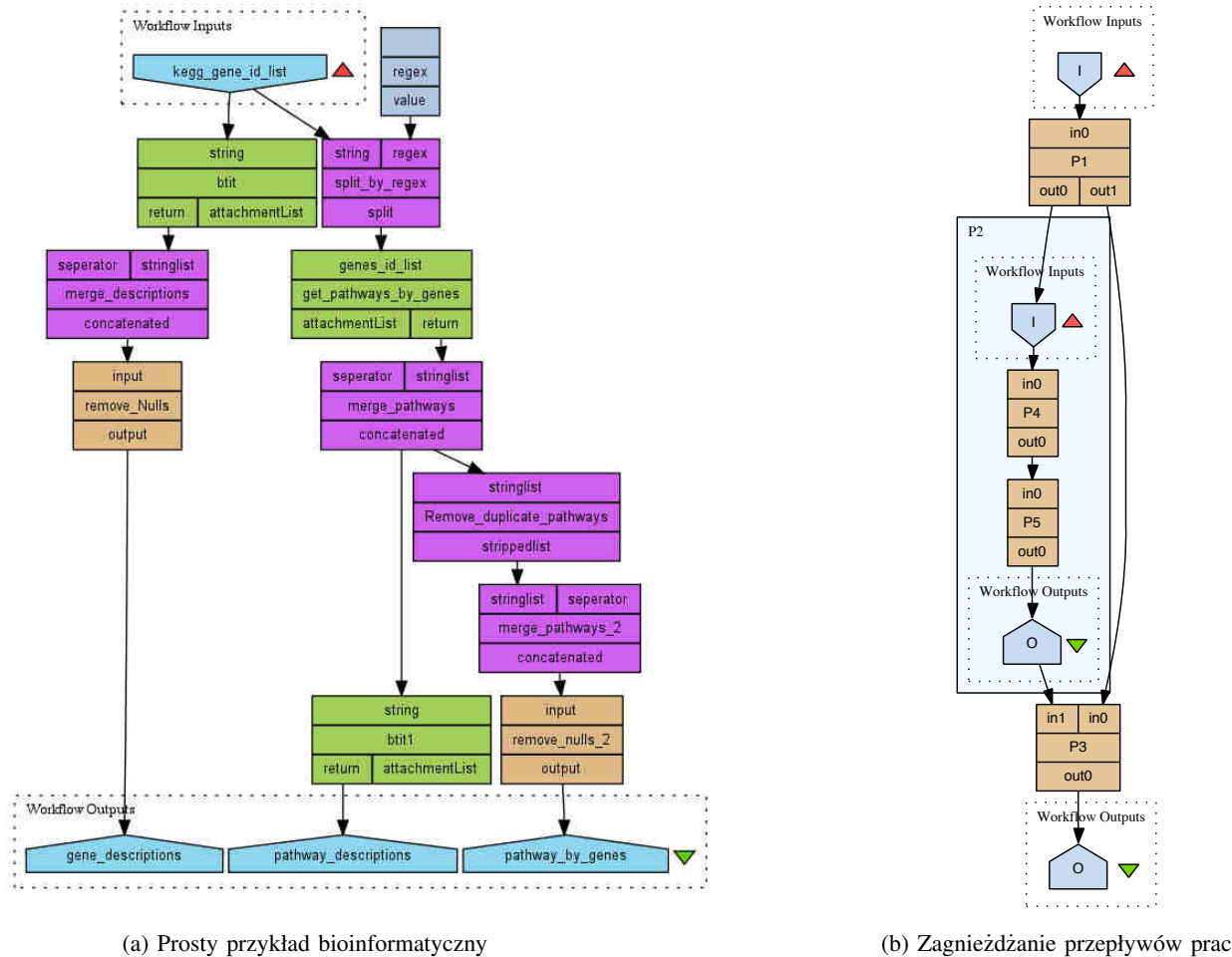
Teraz przedstawię te osiągnięcia bardziej szczegółowo. Dokładne informacje na temat mojego osobistego wkładu oraz wkładu współautorów są zawarte w drugim dokumencie tej aplikacji (*Wykaz osiągnięć naukowych albo artystycznych, stanowiących znaczny wkład w rozwój określonej dyscypliny*).

B. Formalna semantyka systemu Taverna 2

W [E.3] opublikowałem formalny model dla przepływów prac operujących na danych, który był oparty na sieciach Petriego oraz nested relational calculus czyli języku zapytań bazodanowych nad złożonymi obiektami, który odpowiada za obsługę kolekcji elementów danych (w szczególności za iterację) oraz za system typowania. Interesowałem się również analizą modeli stosowanych w systemach przepływów prac używanych przez społeczność e-science, takich jak system Taverna w bioinformatyce. Moja praca została dobrze przyjęta i zostałem zaproszony do współpracy z zespołem Taverny w celu formalnego zdefiniowania semantyki dla systemu Taverna 2.

System Taverna 2 jest następcą systemu Taverna i w tamtym czasie był szeroko stosowany przez społeczności e-science na całym świecie. Jest to zestaw narzędzi zaprojektowany do łączenia rozproszonych usług sieciowych i/lub lokalnych narzędzi w złożone potoki analityczne, które mogą być wykonywane na różnych infrastrukturach obliczeniowych, począwszy od lokalnych komputerów stacjonarnych, przez superkomputery, sieci Grid, aż po środowiska chmurowe. Główne zastosowania Taverny dotyczą e-science, a zwłaszcza bioinformatyki, gdzie przepływy pracy Taverna są zazwyczaj stosowane w obszarach analiz omicznych o wysokiej przepustowości (na przykład, proteomicznych czy transkryptomicznych) lub metod opartych o text mining lub data mining. Taverna 2 zapewnia naukowcom łatwy dostęp do kilku tysięcy różnych narzędzi i zasobów, które są bezpłatnie dostępne z szerokiej gamy instytucji naukowych zajmujących się naukami przyrodniczymi. Po zdefiniowaniu, przepływy pracy stają się wielokrotnie wykonywalnymi definicjami protokołów bioinformatycznych, które można udostępnić, ponownie używać i adaptować, co faktycznie ma miejsce dzięki powstaniu internetowych repozytoriów takich jak www.myexperiment.org.

Na Rysunku 1 przedstawiono przykłady przepływów prac w systemie Taverna 2. Przepływ po lewej stronie przyjmuje dostarczoną przez użytkownika listę genów w formacie oczekiwanej przez bazę danych KEGG i pobiera informacje o szlakach metabolicznych, w które zaangażowane są wprowadzone geny. Używa usługi sieciowej KEGG do pobierania danych oraz kilka dodatkowych skryptów, znanych jako „shims”, które działają jako adaptory między wyjściem jednej usługi a wejściem kolejnej. Przepływ pracy po prawej stronie ma na celu pokazanie możliwości zagnieżdżania przepływów pracy, co ułatwia ich ponowne wykorzystanie, np. użytkownik



Rysunek 1: Przykłady przepływów prac w systemie Taverna 2

może pobrać przepływ prac opublikowany w repozytorium internetowym, takim jak www.myexperiment.org, i użyć go jako elementu bardziej złożonego przepływu prac. Tym samym przepływy prac można konstruować w sposób hierarchiczny, co pomaga radzić sobie ze złożonością, podobnie jak programowanie proceduralne pomaga organizować złożony kod.

Choć graficzna notacja Taverny 2 jest intuicyjna i była z powodzeniem używana przez społeczność e-science, brakowało jej formalnej semantyki, a model przepływu pracy nie był zdefiniowany formalnie. Ponieważ Taverna 2 zawiera kilka cech wysokiego poziomu, które dodają funkcjonalność impliците, użytkownicy mogli skonstruować przepływy pracy, które wyglądają zwięźle i komunikują ideę potoku analitycznego. Dobrze nadają się do użycia w artykule lub prezentacji komunikującej ideę badań, ale wiele złożonych szczegółów, takich jak iteracje po zagnieżdżonych kolekcjach czy mechanizmy rozgałęzień i powtarzania nieudanych prób związanych, np. z awariami usług sieciowych, pozostaje ukrytych. Formalna definicja semantyki była potrzebna, aby umożliwić konstrukcję bardziej złożonych przepływów pracy, na przykład przez połączenie prostszych przepływów opublikowanych przez innych badaczy. Ponadto, znaczenie formalnej definicji modelu dla różnych typów przepływów pracy jest znane, co podkreśla definicja formalnej semantyki dla innych popularnych syste-

mów przepływów pracy, takich jak Kepler [LAB⁺06] i jego prekursor, system Ptolemy [Pto14]. Takie formalnie zdefiniowane modele mogą być używane do wspierania projektowania języków przepływów pracy i ich interpreterów, kompilatorów i optymalizatorów, jak również debuggerów, oraz do wspierania definiowania procedur weryfikacyjnych, podobnych do tych używanych do weryfikacji poprawności złożonych transakcji biznesowych, np. [SBDM⁺24], [ZSF17]. W przypadku Taverna 2, systemu, którego projekt w dużej mierze był napędzany praktyczną potrzebą wspierania automatyzacji procesów dla różnych typów użytkowników, taka formalizacja jest jeszcze ważniejsza, ponieważ zapewnia projektantom, deweloperom oraz użytkownikom przepływów pracy opis zachowania systemu a posteriori, który jest zarówno jednoznaczny, jak i kompletny. Umożliwia to również rozumowanie o właściwościach i równości specyfikacji przepływów pracy oraz badania nad proveniencją, np. [Bel21], [BLR⁺22]). Praca [6] została dobrze przyjęta przez społeczność.

Taverna 2 rozszerzyła model z poprzedniej wersji systemu (dla którego również opracowaliśmy semantykę w [E.1], [E.2] oraz [E.4]) na dwa główne sposoby: (1) poprzez dodanie wsparcia dla przetwarzania potokowego, co z kolei umożliwiło efektywne przetwarzanie strumieni wejściowych o nieokreślonej długości; oraz (2) poprzez wprowadzenie nowych punktów

rozszerzeń, które umożliwiły dodawanie do modelu przepływu pracy nowych operatorów.

Od strony technicznej zdefiniowaliśmy specjalnie dobrany zestaw operatorów pozwalających opisać semantykę Taverna 2 w kategoriach śladów. Formalizacja w kategoriach śladów została wybrana z kilku powodów. Umożliwia badanie równoważności obserwacyjnej przepływów pracy, ponieważ głównym celem przedstawionego formalizmu było zdefiniowanie, kiedy dwie specyfikacje przepływu pracy są równoważne i czy dana właściwość może być wyrażona w języku przepływu pracy Taverna 2, czy też nie. Wymaga to, aby wybrany język miał kompletną semantykę formalną, która rzeczywiście opisuje, kiedy dwa przepływy pracy są obserwacyjnie równoważne.

Ponadto, definicja w kategoriach śladów umożliwia wygodne modelowanie specyficznych cech Taverna 2, takich jak wykonywanie procesorów w ramach potoku wykonawczego oraz użycie stosu wywołań, który definiuje rozszerzalny mechanizm zmieniający zachowanie poszczególnych procesorów, a także system typów, który pozwala na operowanie na rekurencyjnie zagnieżdżonych listach, przy pomocy list comprehensions i iteracji, co w przeciwnym razie wymagałoby znacznych rozszerzeń podstawowych formalizmów, takich jak rachunek π i sieci Petriego. Te cechy są wysoce ekspresywne i niewidoczne w graficznej notacji Taverna 2, a jednocześnie stanowią wyzwanie dla precyzyjnej specyfikacji. Ich formalny opis wykracza poza zakres tego streszczenia, ale przedstawiam ich nieformalny opis, aby podkreślić złożoność systemu. Po dalsze szczegóły odsyłam czytelnika do [6].

Przetwarzanie potokowe pozwala na iterację *implicite* po elementach kolekcji (i iloczynach wielu kolekcji w przypadku wielu wejść) i ma na celu poprawę efektywności wykonania na dużych zbiorach danych. Iteracje po elementach listy mogą być wykonywane równocześnie, jeśli dostępne są wystarczające zasoby do obsługi równoległych wątków wykonawczych. Gdy tak się dzieje, elementy listy wynikowej mogą być produkowane w dowolnej kolejności, ze względu na różne prędkości poszczególnych wątków. Ta dopuszczalna różnica kolejności jest uchwycona w śladach przepływu pracy, poprzez rejestrowanie zdarzeń, które reprezentują indywidualne miejsca listy wyjściowej przypisywane wartości oraz wartości w miejscach wejściowych używane przez daną aktywność.

Stos wywołań pozwala na skonfigurowanie poszczególnych procesorów P w grafie za pomocą stosu warstw wykonawczych, a przetwarzanie, które ma miejsce, gdy wykonanie przepływu pracy dotrze do P , zależy od tej konfiguracji. Każda warstwa w stosie otrzymuje żądania od warstwy powyżej, wykonuje określoną funkcję, która przekształca te żądania, a następnie albo przekazuje je do warstwy poniżej, albo zwraca je z powrotem w górę stosu jako odpowiedź. Taverna 2 oferuje różnorodne wbudowane standardowe warstwy (*Invoke*, *Retry_k*, *Bounce*, *Failover*, *Branch* i *Loop*), ale stos może również zawierać niestandardowe warstwy dostarczone przez użytkownika. Daje to możliwość implementacji dodatkowych operatorów sterujących.

Waga prostej formalnej semantyki oraz praktycznie motywowana potrzeba dostępności metod analizy i weryfikacji zwiększyły moje zainteresowanie formalizmem sieci Petriego,

z którym wiążą się pozostałe prace.

C. Symulacja sieci metabolicznych skali genomu dla komórek

W [5] opracowaliśmy ACORN, otwartoźródłowy (GNU GPL) system gridowy do symulacji sieci zdefiniowanych przy pomocy więzów w interaktywnym środowisku internetowym.

ACORN umożliwia symulację metabolizmu w rekonstrukcjach sieci metabolicznych komórek na skalę genomową. Przykład małej części takiej sieci przedstawiono na Rysunku 2. Jest on wyrażony za pomocą notacji sieci Petriego¹, gdzie elipsy reprezentują substancje, reakcje są przedstawione jako prostokąty, a strzałki określają substraty i produkty reakcji.

Użytkownik ACORNa może analizować sieci reakcji biochemicznych, aby umożliwić przewidywania dotyczące fenotypu² na podstawie katalogu części molekularnych zakodowanych w genomie. System jest modelowany przez zestaw zależności liniowych między obliczanymi przepływami metabolitów przez reakcje. Chociaż układ liniowy odpowiadający modelom na skalę genomową jest niedookreślony, można go użyć do optymalizacji za pomocą programowania liniowego (PL) funkcji celu, która reprezentuje zdolności metaboliczne będące przedmiotem zainteresowania. Funkcja celu jest definiowana jako przepływ przez określoną reakcję lub przepływ w kierunku określonego metabolitu. Optymalizacja PL prowadzi do unikalnej wartości funkcji celu, ale odpowiadający jej rozkład przepływów nie jest unikalny. Zakres każdego indywidualnego przepływu przez reakcje we wszystkich rozkładach przepływów, które są zgodne z maksymalną wartością funkcji celu, można oceniać za pomocą iteracyjnych protokołów PL, takich jak Analiza Zmienności Przepływu (ang. Flux Variability Analysis, FVA) [MS03]. Te podejścia są określane jako oparte na więzach, ponieważ oceniają zdolności metaboliczne w ramach zestawu ograniczeń wyrażonych przez równania bilansowe wyrażające w sposób ilościowy wartości pewnych przepływów (określone na podstawie eksperymentów *in vitro*) oraz informacji (z literatury) o maksymalnej przepustowości i odwracalności reakcji. Pozyskiwanie tych informacji można zainicjować przez analizę sekwencji genomu, a podejście to zostało zastosowane do rekonstrukcji sieci reakcji metabolicznych skali genomu (ang. Genome-Scale Metabolic Network, GSMNs) dla głównych organizmów modelowych, patogenów i mikroorganizmów stosowanych w procesach biotechnologicznych. Dostępna jest również rekonstrukcja metabolizmu człowieka, która jest używana do badania mechanizmów chorób metabolicznych.

Opracowana przez nas gridowa architektura ACORNa ułatwia wykonywanie intensywnych obliczeniowo, iteracyjnych protokołów, takich jak FVA, w interaktywnym środowisku internetowym i może być łatwo skalowana wraz ze wzrostem liczby użytkowników i modeli. Część systemu działająca na serwerze internetowym pozwala na wizualizację wyników numerycznych na mapach ścieżek zdefiniowanych przez użytkownika. ACORN zawiera również zintegrowany edytor

¹Oryginalnie sieci Petriego zaprojektowano do wyrażania reakcji chemicznych, a dopiero później zostały zaadaptowane jako matematyczny język modelowania do specyfikacji systemów rozproszonych.

²W genetyce fenotyp to zbiór obserwowalnych cech lub właściwości organizmu.

towy, zgodnie z zasadą „myśl jak wierzchołek” z systemów typu Pregel [MAB⁺10], które opracowano w przemyśle do przetwarzania grafów Big Data, takich jak sieci społecznościowe na skalę globalną. Symulacje mogą być uruchamiane na klastrach komputerowych, ale ponieważ symulacja komórek QSSPN jest obliczeniowo intensywna, wprowadziliśmy optymalizacje z użyciem specjalnego rodzaju pamięci podręcznej na węzłach, aby przyspieszyć obliczenia strumieni metabolicznych w stanie stacjonarnym. Optymalizacje te minimalizują narzut przetwarzania kosztem wyższych wymagań pamięciowych. Buforujemy jedynie stan istotnych miejsc, dla których wprowadziliśmy liniowy porządek zamiast pełnych ograniczeń dla problemu optymalizacji liniowej. Zredukowało to zapotrzebowanie na rozmiar pamięci podręcznej o kilka rzędów wielkości względem [A.3], co pozwoliło nam przechowywać całą pamięć podręczną i umożliwiło bardziej efektywne wyszukiwanie. Podobne rodzaje optymalizacji badałem dalej w [A.4] w celu umożliwienia weryfikacji dynamicznego zachowania modeli QSSPN.

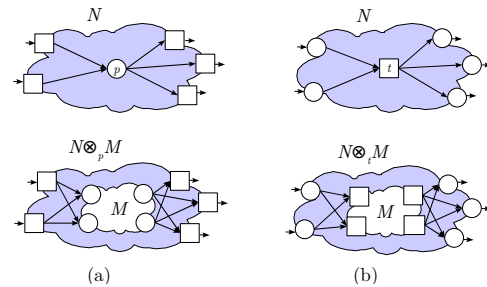
Wymyślona przeze nas architektura została pomyślnie zastosowana do symulacji interakcji między komórkami wątroby przy pomocy cytokiny FGF19 podczas homeostatycznej odpowiedzi na gwałtowny wzrost cholesterolu. To pokazało, że podejście może być użyte do mechanistycznego modelowania wyłaniania się zachowań systemów wielokomórkowych w wyniku interakcji między genomem a środowiskiem.

E. AND-OR sieci i *-soundness

Następnie przedstawiam grupę teoretycznych artykułów, które koncentrują się na opracowywaniu metod konstrukcji poprawnych przepływów pracy oraz weryfikacji poprawności przepływów pracy skonstruowanych w inny sposób. Rozważane przepływy pracy są wyrażane jako sieci Petriego, a ogólnie przyjęte kryteria poprawności jak *soundness*, *k-soundness* i **-soundness* [vHSV03] są wybrane jako punkt wyjścia. Metody, które opracowaliśmy są hierarchiczne i mają na celu uzyskania jak najszerszej klasy sieci.

Oprócz gwarantowanej poprawności, konstruowanie sieci w sposób hierarchiczny daje tę przewagę, że produkuje sieci przepływu z wyraźną i naturalną hierarchiczną strukturą, co znacznie ułatwia zrozumienie specyfikacji. Może to być przydatne w narzędziach do projektowania i analizy oraz pozwala użytkownikowi na przybliżanie i oddalanie konkretnych części sieci poprzez rozwijanie lub zwijanie węzłów zgodnie z hierarchiczną strukturą. Ponadto struktura ta może często być dopasowana do hierarchii organizacji, której dotyczy konkretny przepływ prac lub hierarchii w komórce lub organizmie, który jest modelowany, a tym samym może być powiązana na przykład z poziomami zarządzania w organizacji lub procesami komórkowymi, takimi jak anabolizm, neurotransmisja czy detoksykacja. Co więcej, struktura hierarchiczna może pomóc w eleganckim specyfikowaniu obsługi wyjątków i przywracaniu poprawnego stanu po niespodziewanych sytuacjach [CWBH⁺03], [CW06a]. Bardziej rozbudowaną motywację i opis zalet hierarchicznej metody projektowania sieci można znaleźć w [DRM⁺12], [PGBD12].

Badania te zaowocowały trzema artykułami w czasopiśmie.



Rysunek 4: Ilustracja podstawienia za miejsce i podstawienia za tranzycję

W [4] uogólniliśmy sieci przepływów prac do dwóch nowych klas sieci, zwanych sieciami pWF i sieciami tWF. Nieformalnie takie sieci mają węzły graniczne będące miejscami lub przejściami, odpowiednio, ale nie są ograniczone do pojedynczego węzła wejściowego i wyjściowego, jak to ma miejsce w przypadku klasycznych sieci przepływów prac [vdA98]. Aby odnieść się do wariantu z pojedynczym węzłem wejściowym i wyjściowym, używam przedrostka 1l, na przykład, przy pomocy sieci 1lpWF odnoszę się do klasy wszystkich sieci przepływów prac z pojedynczym miejscem wejściowym i wyjściowym.

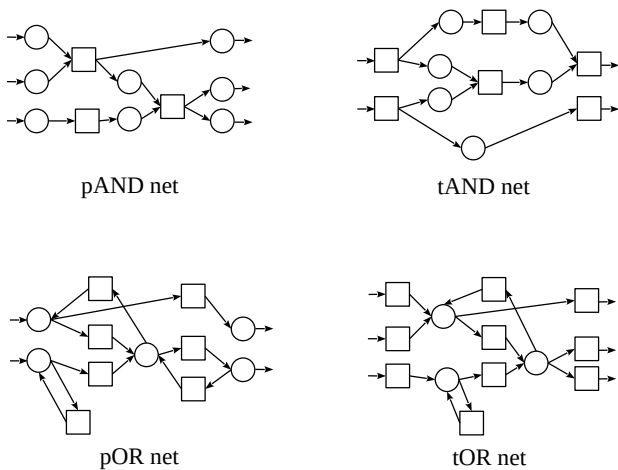
Następnie wprowadzamy *AND-OR* sieci, które są specjalnymi podklasami sieci pWF i sieci tWF. Rozważamy ogólne podstawienia węzłów w istniejących sieciach za pomocą sieci o pasującym typie interfejsu, tj. gdy miejsca i przejścia są zastępowane przez odpowiednio sieci pWF i sieci tWF. Nasze pojęcie podstawienia opiera się na tym wprowadzonym przez Van Hee i in. w [vHSV03], ale jest uogólnione, aby mogło zastępować sieci z wieloma węzłami wejściowymi i wieloma węzłami wyjściowymi.

Wyniki podstawienia miejsca i podstawienia tranzycji są przedstawione na Rysunku 4 (a) i (b), odpowiednio. W artykule badamy również właściwości takich podstawień, jak na przykład asocjatywność.

Następnie rozważamy, które klasy sieci można otrzymać, zaczynając od pewnych podstawowych klas sieci i pozwalając na podstawienia miejsc za pomocą sieci pWF oraz tranzycji za pomocą sieci tWF.

Jako podstawowe sieci, od których rozpoczynamy proces generowania, rozważamy sieci, które nazywamy sieciami pAND, tAND, pOR i tOR, które zilustrowano na Rysunku ??, odpowiednio z węzłami wejściowymi po lewej stronie i węzłami wyjściowymi po prawej stronie. Nieformalnie sieci AND można opisać jako acykliczne sieci przepływów, które składają się wyłącznie z AND splits i AND joins, a sieci OR można opisać jako potencjalnie cykliczne sieci składające się wyłącznie z OR splits i OR joins. Sieci AND i OR są odpowiednio uogólnieniami marked graphs/T-nets i maszyn stanowych/S-nets [DE05], które są ograniczone do dokładnego jednego węzła wejściowego i wyjściowego w podstawianych sieciach.

Formalne definicje są skonstruowane tak żeby generowane sieci były poprawne w sensie *soundness*. Dla zdefiniowanych



Rysunek 5: Przykłady sieci pAND, tAND, pOR i tOR

sieci AND i OR istnieją pewne oczywiste wyniki dotyczące poprawności. Wszystkie sieci pAND i tOR są *-poprawne, a dla sieci tAND i pOR jest to prawda, jeśli są to sieci jedno-wejściowe jedno-wyjściowe. W [4] dostarczamy formalnych dowodów silniejszych właściwości substitution soundness dla wszystkich omawianych tutaj podstawowych klas.

Definition 1 (AND-OR net). Klasę $S(\text{pAND} \cup 11\text{tAND} \cup 11\text{pOR} \cup \text{tOR})$ nazywamy klasą AND-OR sieci.

Pokazujemy również, że jedno-wejściowe jedno-wyjściowe sieci tAND nie są potrzebne, tzn. mogą zostać usunięte z początkowej klasy bez zmiany zbioru sieci, które mogą być wygenerowane.

Jednak jedno-wejściowe jedno-wyjściowe sieci pOR są niezbędne, ponieważ cykl zawierający węzły wejściowe i wyjściowe nie może być uzyskany w inny sposób, na co podajemy kontrprzykład. Oczywiście, sieci pAND i tOR również nie są zbędne, ponieważ pozwalają na wielokrotne węzły wejściowe i wyjściowe.

AND-OR sieci są bardzo podobne do ST-sieci zdefiniowanych w [vHSV03]. W rzeczywistości klasa ST-sieci jest ścisłą podklasą domknięcia ze względu na podstawienia klasy $11\text{tAND} \cup 11\text{pOR}$, które zabrania krawędzi wchodzących dla węzłów wejściowych i krawędzi wychodzących dla węzłów wyjściowych. Jest jasne, że domknięcie ze względu na podstawienia klasy $11\text{tAND} \cup 11\text{pOR}$ jest właściwą podklasą AND-OR sieci, ponieważ zawiera tylko jedno-wejściowe jedno-wyjściowe sieci przepływu. Jednakże, istnieją również jedno-wejściowe jedno-wyjściowe AND-OR sieci, które nie są w domknięciu ze względu na podstawienia klasy $11\text{tAND} \cup 11\text{pOR}$.

Niestety, nie jest prawdą, że *-soundness jest zachowywana przez podstawienia zdefiniowane w tym artykule. Dzieje się tak ze względu na możliwe krawędzie wychodzące z węzłów wyjściowych, na co prezentuję kontrprzykłady. Dlatego wprowadzamy nowe pojęcie poprawności zwane *substitution soundness* (w skrócie sub-soundness) i badamy jego właściwości. Pokazujemy, że wszystkie podstawowe klasy sieci z definicji AND-OR sieci są sub-sound.

Intuicja stojąca z sub-soundness jest taka, że nie powinno mieć znaczenia, że podczas uruchamiania sieci przepływu

usuwamy pozornie gotowe żetony z miejsc wyjściowych. Innymi słowy, jeśli sieć zaczyna z k żetonami w miejscach wejściowych, osiąga markowanie z co najmniej $k' \leq k$ żetonami w każdym miejscu wyjściowym i usuwamy te k' żetonów z każdego miejsca wyjściowego, to sieć nadal może zakończyć z $k - k'$ żetonami w każdym miejscu wyjściowym. Jest to w pewnym sensie niezbędny warunek do konstruowania 1-sound sieci przez podstawienie węzłów w 1-sound sieciach. W szczególności można wykazać, że nie ma słabszego warunku, który jest zachowywany przez podstawienie i implikuje 1-soundness.

W pracy dowodzimy, że sub-soundness jest wystarczające do konstruowania *-sound sieci przez podstawienie i że wszystkie AND-OR sieci są sub-sound. Wreszcie, w [4] pokazujemy, że wszystkie AND-OR sieci są *-sound.

F. Sieci hierarchiczne i rozszerzone sieci hierarchiczne

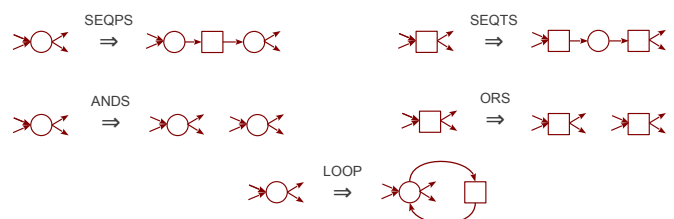
Po zdefiniowaniu AND-OR sieci, w [4] pokazujemy, że nasze podstawienia są nie tylko bardziej ogólne niż te dla sieci ST-nets van Hee i in., ale są również wystarczająco silne, aby wyrazić klasy sieci, które w innych podejściach wymagają rozszerzeń niehierarchicznych.

W tym celu przyjrzelśmy się klasom *sieci hierarchicznych* oraz *rozszerzonych sieci hierarchicznych* z [CWBH⁺03]. Pierwsza z nich jest zdefiniowana za pomocą pięciu prostych i intuicyjnych reguł (patrz Rysunek 6)³. Reguły te odpowiadają następującym konstrukcjom:

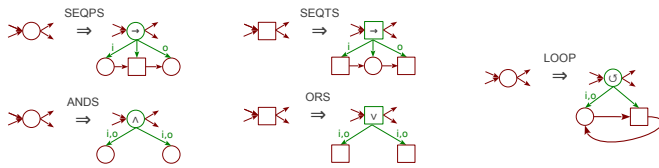
- 1) *Sekwencja*. Miejsce (odpowiednio tranzycja) może zostać rozwinięte w sekwencję przeplatających się miejsc i tranzycji rozpoczynających się i kończących miejscem (odpowiednio tranzycją).
- 2) *Równoległość*. Miejsce może zostać skopiowane wraz z jego krawędziami wejściowymi i wyjściowymi, aby reprezentować dwa równoległe wątki.
- 3) *Wybór*. Tranzycja może zostać skopiowana wraz z jej krawędziami wejściowymi i wyjściowymi, aby reprezentować dwa wykluczające się wybory.
- 4) *Pętla*. Tranzycja może zostać dodana i samozapętlić się z wybranym miejscem, aby umożliwić wielokrotne powtórzenie pętli reprezentowanej przez tę tranzycją.

Te konstrukcje wybrano, bo odpowiadają znanym konstrukcjom programistycznym: sekwencyjnej kompozycji poleceń,

³W kontekście tej podsekcji sieci hierarchiczne odnoszą się do klasy sieci zdefiniowanych przez te konkretne pięć reguł, a nie do sieci skonstruowanych metodami hierarchicznymi.



Rysunek 6: Reguły definiujące sieci hierarchiczne



Rysunek 7: Reguły dla drzewa uszczegółowień

instrukcji wyboru, wywołaniu równoległemu wątków i tworzeniu pętli.

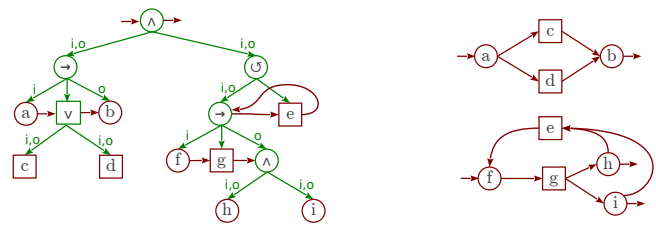
Rozszerzone sieci hierarchiczne z [CWBH⁺03] uzupełniają sieci hierarchiczne o dwa rodzaje dodatkowych niehierarchicznych reguł, pozwalających wprowadzić (1) synchronizację między dwoma równoległe działającymi podprzepływami pracy lub (2) przekazanie wątku z jednego podprzepływu do drugiego.

W [4] przedstawiliśmy pierwszą, w pełni formalną definicję obu klas z niezbędnymi warunkami wstępnymi dla rozszerzonych reguł. Pokazaliśmy, że klasa rozszerzonych sieci hierarchicznych, zdefiniowana poprzez dodanie tych dwóch dodatkowych typów reguł, jest właściwym nadzbiorem sieci hierarchicznych, ale nadal wszystkie takie sieci wykazują właściwość poprawności *-soundness. Osiągnięto to, pokazując, że klasa ta jest właściwym podzbiorem sieci AND-OR.

Rozszerzone reguły mają naturalne i niezbędne warunki wstępne, które gwarantują, że wynik jest nadal poprawną siecią przepływu prac. W przypadku (1), gdzie chcemy zsynchronizować dwie akcje w dwóch podprzepływach pracy, należy zapewnić, że podprzepływy te reprezentują równoległe wątki, które zawsze są wykonywane razem, w przeciwnym razie łatwo może wystąpić zakleszczenie. Dualnie, w przypadku (2), jeśli po dokonaniu wyboru między dwoma podprzepływami pracy chcemy umożliwić transfer między nimi, można to bezpiecznie zrobić, pod warunkiem że w międzyczasie nie było żadnego rozwidlenia wątków. Aby umożliwić formalną definicję tych warunków wstępnych, wprowadzono *drzewo uszczegółowień* (ang. refinement tree), które przedstawia proces generowania sieci hierarchicznej.

Zamiast zastępować istniejący węzeł sieci nowymi węzłami sieci, zastępujemy go węzłem reguły, do którego nowe węzły sieci są podczipione jako jego potomkowie. Zasady generowania drzewa uszczegółowień przedstawiono na rysunku 7. Zielone (wewnętrzne) węzły są węzłami reguł, a zielone (zstępujące) krawędzie przedstawiają, jak przebiega uszczegółowienie. Czerwone węzły (liście) reprezentują wygenerowaną sieć przepływu prac. Zielone (zstępujące) oznaczenia krawędzi definiują, które krawędzie przodka są ponownie połączone z potomkiem w wytworzonej hierarchicznej sieci.

Przykład drzewa uszczegółowień z odpowiadającą mu hierarchiczną siecią przedstawiono na rysunku 8. Węzły sieci są nazwane dla wygody czytelników. Należy zauważyć, że podczas gdy drzewo uszczegółowień definiuje hierarchiczną sieć, dla jednej sieci może istnieć wiele drzew uszczegółowień. W pracy zbadaliśmy tę niejednoznaczność.

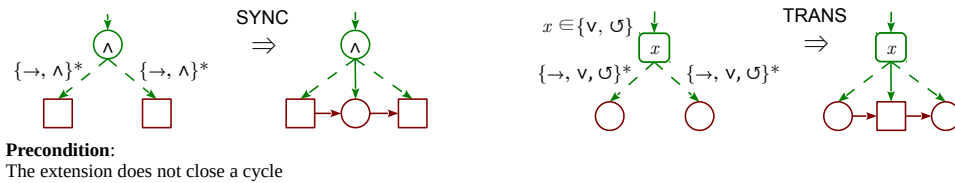


Rysunek 8: Generowanie prostej hierarchicznej sieci przepływu prac z drzewem uszczegółowień

Zbadaliśmy również, jak reguły mogą być stosowane w odwrotnym kierunku, poprzez kontrakcję fragmentów sieci w pojedyncze węzły. Taka kontrakcja stanowi efektywny sposób sprawdzania, czy sieć jest hierarchiczna. Najpierw pokazujemy lemat, który mówi, że dla dowolnych dwóch możliwych kontrakcji, nawet jeśli istnieje konflikt, ponieważ podsieci z kontrakcji mają część wspólną, to istnieje pewna forma konfluencji, że po wykonaniu dowolnego z nich nadal możliwe jest osiągnięcie izomorficznej sieci poprzez wykonanie dodatkowego kroku redukcji. Ten lemat jest następnie używany do pokazania, że aby zweryfikować, czy sieć jest hierarchiczna, wystarczy przeprowadzać kontrakcję jej fragmentów, aż zostanie tylko jeden węzeł i że możliwość osiągnięcia pojedynczego węzła nie zależy od kolejności, w jakiej dokonujemy kontrakcji, tzn. możemy być zachłanni i nie jest konieczne stosowanie backtrackingu.

Hierarchiczne sieci zapewniają wygodny sposób prezentowania złożonych procesów tak, aby struktura procesu, jego równoległość oraz możliwe wybory wzajemnie wykluczających się zachowań były łatwe do zidentyfikowania dla ludzkich czytelników. Proces kontrakcji ułatwia weryfikację, czy sieć jest hierarchiczna, podczas gdy proces uszczegółowienia pozwala na generowanie sieci, w których weryfikacja nie jest wymagana. Jednak wciąż istnieje wiele sieci *-sound, które są przydatne do celów modelowania, ale nie są hierarchiczne. W [CWBH⁺03] metoda generowania hierarchicznego jest rozszerzona o niehierarchiczne reguły, które pozwalają uwzględnić niektóre z takich przypadków. Do kontrolowania warunków wstępnych użyto koncepcji drzewa uszczegółowień.

Rozszerzone, niehierarchiczne reguły SYNC i TRANS przedstawiono na Rysunku 9. Dodanie miejsc i przejść za pomocą tych reguł jest zaznaczone w drzewie uszczegółowień przez dodanie nowych węzłów jako potomków pierwszego wspólnego przodka dwóch węzłów, między którymi są one dodawane. Drzewa uszczegółowień wytworzone w ten sposób nazywane są *rozszerzonymi drzewami uszczegółowień*. Formalnie, aby rozszerzone reguły mogły być zastosowane, musi być spełniony warunek, że pierwszy wspólny przodek w rozszerzonym drzewie uszczegółowień jest typu dozwolonego przez korzeń reguły, tzn. ANDS dla reguły SYNC i ORS lub LOOP dla reguły TRANS. Ponadto, wszystkie węzły na ścieżce od tego wspólnego przodka do rodzeństwa nowego węzła muszą mieć typ dozwolony przez odpowiadającą regułę, tzn. SEQPS, SEQTS i ANDS dla reguły SYNC oraz SEQPS, SEQTS, ORS i LOOP dla reguły TRANS. Dodatkowym wa-



Rysunek 9: Dodatkowe reguły dla rozszerzonych sieci hierarchicznych

runkiem wstępnym dla reguły SYNC jest, aby wprowadzone rozszerzenie nie zamknęło cyklu.

Reguły hierarchiczne i niehierarchiczne mogą być swobodnie mieszane (z zachowaniem warunków wstępnych dla reguł rozszerzonych) podczas konstruowania sieci, tzn. po dodaniu dodatkowych węzłów za pomocą reguł rozszerzonych, te lub wcześniej istniejące węzły mogą być uszczegółowione standardowymi regułami, następnie mogą być ponownie dodane nowe węzły, i znowu dowolne węzły mogą być uszczegółowione, i tak dalej.

Należy zauważyć, że nowe węzły wprowadzone przez SYNC i TRANS nie zastępują żadnych węzłów, więc nie dostają żadnych odziedziczonych krawędzi, jak miało to miejsce w przypadku innych reguł, ale czerwone krawędzie, które są z nimi wprowadzane uczestniczą normalnie w dalszym procesie uszczegółowienia.

Na koniec pracy omawiamy związek między sieciami AND-OR a rozszerzonymi hierarchicznymi sieciami. Pokazujemy, że wszystkie rozszerzone sieci hierarchiczne są sieciami AND-OR, a zatem dowodzimy, że są *-sound. Należy zauważyć, że chociaż rozszerzone sieci hierarchiczne stanowią właściwy podzbiór AND-OR sieci, nadal są interesujące, ze względu na swoją prostotę oraz ponieważ istnieją dla nich pewne unikalne wyniki, takie jak analiza dźwiękowych oznaczeń [CW06b].

G. Wyznaczanie AND-OR hierarchii w sieciach przepływu prac

W ostatnim artykule [2] zbadaliśmy pojęcie *redukcji*, gdzie sieć przepływu prac jest przekształcana w mniejszą sieć poprzez iteracyjne kontrakcje niektórych dobrze uformowanych podsieci do pojedynczych węzłów, aż do momentu, gdy takie kontrakcje nie są już możliwe. Ta redukcja może ujawnić hierarchiczną strukturę sieci przepływu prac i, ponieważ zachowuje niektóre właściwości semantyczne, takie jak soundness, może pomóc w analizie i zrozumieniu, dlaczego sieć przepływu prac jest sound lub nie. Redukcja może być również używana do weryfikacji, czy sieć przepływu prac jest siecią AND-OR.

Pokazaliśmy, że pomimo niedeterminizmu wynikającego z wyboru podsieci, które podlegają kontrakcji, końcowy rezultat procesu redukcji jest zawsze taki sam z dokładnością co do wyboru tożsamości węzłów. Na podstawie tego wyniku, przedstawiliśmy wielomianowy algorytm, który oblicza ten unikalny wynik redukcji. Na koniec pokazaliśmy, jak tego algorytmu można użyć do weryfikacji, czy sieć przepływu prac jest siecią AND-OR.

Algorytm redukcji może być używany nie tylko dla sieci AND-OR, ale także do analizy soundness i struktury innych sieci przepływów prac. Może pomóc użytkownikowi w

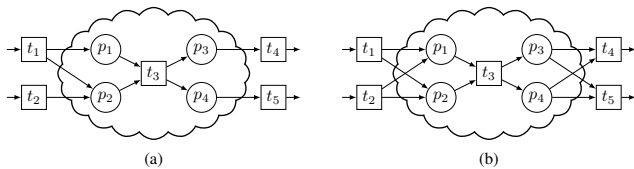
znalezieniu problemów powodujących brak soundness. Konkretniej, jeśli wynik weryfikacji AND-OR jest negatywny, algorytm redukcji zatrzymuje się, nie osiągając jednowęzłowej sieci przepływu prac. Powstała sieć może służyć jako skondensowana wersja oryginalnej sieci i wskazywać użytkownikowi źródło problemu w projekcie. Należy zauważyć, że sieć przepływu prac może nie być siecią AND-OR, ale nadal być *-sound lub sub-sound. Przypuszczamy, że aby zweryfikować *-soundness lub sub-soundness dowolnej sieci, wystarczy zweryfikować *-soundness lub sub-soundness sieci wynikającej z procedury weryfikacji AND-OR. Kontrakcje użyte w naszym algorytmie musiałyby zostać udowodnione jako zachowujące *-soundness lub sub-soundness, podobnie jak na przykład zasady [Mur89] zachowują żywotność i ograniczoność. To dałoby symetryczny i prawdopodobnie podobnie pracochłonny wynik do [4], gdzie pokazaliśmy, że podstawienia sieci AND-OR za pasujące węzły sieci AND-OR zachowuje sub-soundness, z czego wynika, że zachowują również *-soundness. Zredukowana sieć, wynikająca z procedury weryfikacji sieci AND-OR, mogłaby następnie przejść właściwą weryfikację pod względem soundness podobnymi metodami jak w [Ver04], [VBvdA01], [PWW11], [FFK⁺11]. Ponadto ograniczenie rozmiaru weryfikowanej sieci za pomocą metod hierarchicznych może być pomocne dla użytkowników borykających się ze zrozumieniem przyczyn braku soundness sieci przepływu prac. Jest to często występujący problem, nawet przy użyciu automatycznych narzędzi weryfikacyjnych, jak na przykład pokazano w [FF06].

Wynik procedury redukcji może pomóc w lepszym zrozumieniu przepływu pracy. Przykładowo w kontekście QSPN fragmenty sieci regulacji mogłyby być zredukowane bez zmiany zachowania reszty umożliwiając tym samym łatwiejszą analizę i weryfikację. Podobne korzyści omówiono w [Gol14] dla grafów przepływu sterowania (ang. control flow graphs), które są dobrze znaną reprezentacją sekwencyjnej struktury przepływu sterowania programów i mają wiele zastosowań. Redukcje grafów ułatwiają inżynierom oprogramowania analizę przepływu sterowania w dużych grafach reprezentujących funkcje, zestawy funkcji, a nawet całe programy. Ponadto, jako produkt uboczny udanej redukcji sieci przepływu prac, można określić strukturę drzewa uszczegóławiania opisującą zagnieżdżenie fragmentów sieci. Podobnie jak w przypadku podobnych metod [CWBH⁺03], [CW06a], [CGL13], które dotyczą klas sieci przepływów prac będących właściwymi podklasami sieci AND-OR, taka struktura drzewa może być używana do modelowania obszarów odzyskiwania (ang. recovery regions) lub określania poprawnych markowań (ang. sound markings), lub po prostu do lepszego zrozumienia struktury sieci przepływu prac i jej właściwości. To ostatnie

może na przykład pomóc w określeniu, jak części przepływu pracy mogą być najlepiej rozdzielone między niezależne jednostki organizacyjne lub różne serwery w przypadku przepływów pracy reprezentujących obliczenia, np. w przepływach takich jak w Tavernie.

Nieformalnie, procedurę można opisać jako odwrócenie procesu generowania. Oznacza to, że staramy się znaleźć podsieci, które mogły zostać wygenerowane przez podstawienie sieci za węzeł i odwrócenie tego. Proces ten powtarza się, dopóki nie udaje się znaleźć więcej takich podsieci, a jeśli wynikowa sieć jest pojedynczym węzłem, wnioskujemy, że oryginalna sieć jest siecią AND-OR.

W [2] omówiliśmy, dlaczego wynik kontrakcji jest faktycznie siecią Petriego i przedstawiliśmy procedurę redukcji, w której szukamy niepustych podzbiorów węzłów w sieci przepływu prac, takich że odpowiadająca im sieć jest nietrywialną podstawową siecią AND-OR, i dokonujemy ich kontrakcji do jednego nowego węzła. Taka kontrakcja powinna być odwrotnością podstawienia za ten nowy węzeł z powiązanej sieci w tym sensie, że jeśli zastosujemy kontrakcję, a następnie podstawienie, powinniśmy ponownie uzyskać początkową sieć przepływu prac. Jednak zastosowanie tego podstawienia może nie zawsze dać z powrotem początkową sieć. Aby poradzić sobie z tym problemem, wprowadzamy własność *dobrego zagnieżdżenia*, która dotyczy zewnętrznych presetów i postsetów podsieci. Dobre zagnieżdżenie podsieci jest zarówno koniecznym, jak i wystarczającym warunkiem, aby podstawienie odwróciło kontrakcję.



Rysunek 10: Przykład podsieci nie będącej dobrze zagnieżdżoną (a) oraz podsieci dobrze zagnieżdżonej (b)

Na podstawie tej własności zdefiniowaliśmy pojęcie AND-OR redukcji sieci przepływu prac, w dalszej części nazywanej po prostu redukcją sieci przepływu prac, które opisuje wynik redukcji sieci przepływu prac przez stosowanie kontrakcji, aż nie będzie to dalej możliwe.

Głównym wynikiem pracy jest to, że redukcja sieci przepływu prac jest zdefiniowana jednoznacznie co do izomorfizmu, tzn. wszystkie wyniki są identyczne, jeśli nie brać pod uwagę tożsamości węzłów. Dowód tego twierdzenia opiera się na wykazaniu, że jest tak dla pojedynczego kroku, a następnie wykazaniu, że z tego można wywnioskować, że cała procedura redukcji posiada tę własność.

Formalny dowód dla jednego kroku jest dość złożony, dlatego w [2] przedstawiliśmy jego zwięzłą wersję, a szczegółową wersję z wszystkimi detalami przedstawiliśmy w [SH17]. Podstawowa idea dowodu dla pojedynczego kroku polega nieformalnie na tym, że jeśli istnieją dwie podsieci, które można poddać kontrakcji, to (1) możemy wykonać ich kontrakcję w dowolnej kolejności, tzn. po wykonaniu kontrakcji jednej sieci, drugą nadal można poddać kontrakcji, oraz (2) niezależnie

od kolejności kontrakcji uzyskujemy ten sam wynik co do izomorfizmu. Jest to w oczywisty sposób prawdziwe, jeśli sieci są rozdzielne w kontekście relacji przepływu, ale staje się bardziej skomplikowane, jeśli są one połączone przez pewne krawędzie lub nawet zachodzą na siebie. Dla tych ostatnich możliwości wyróżniamy cztery przypadki, które są zilustrowane na Rysunku 11. Na tym rysunku używamy chmur S_1 i S_2 do wskazania podsieci oraz zaokrąglonych prostokątów n_1 i n_2 do reprezentowania węzłów, które mogą być zarówno miejscami, jak i tranzycjami. Wyróżniamy następujące przypadki:

Na końcu pracy [2] przedstawiamy algorytm pozwalający obliczać wynik redukcji oraz analizę jego poprawności i złożoności. Pokazujemy również, jak można go użyć do weryfikacji, czy sieć przepływu prac jest siecią AND-OR. Gdyby redukcja była obliczana w naiwny sposób, przez iterację po wszystkich podsieciach w celu sprawdzenia, czy którąś z nich można poddać kontrakcji, algorytm byłby wykładniczy względem rozmiaru wejściowej sieci przepływu prac. Pokazujemy, że można to zrobić w bardziej wyrafinowany sposób, otrzymując w wyniku algorytm o złożoności wielomianowej. Dodatkowo, pokazujemy, jak ten algorytm użyć do weryfikacji, czy dana sieć przepływu prac jest siecią AND-OR.

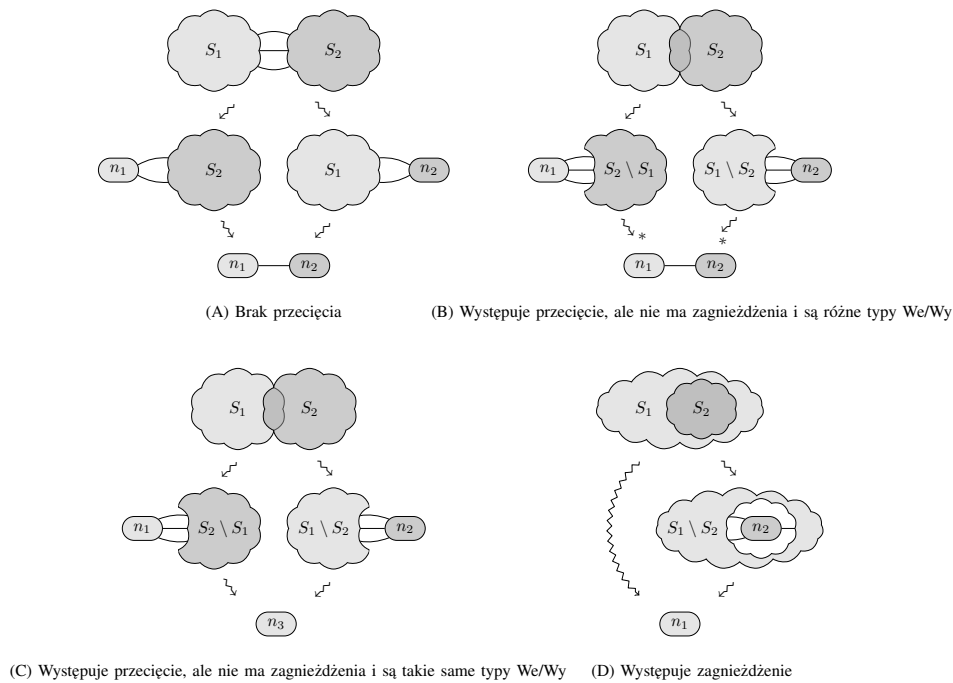
Algorytm opiera się na twierdzeniu o jednoznaczności co do izomorfizmu redukcji sieci przepływu prac, co pozwala na procedowanie bez backtrackingu. Zasadniczo powtarzając proces poszukiwania nietrywialnych podsieci, które można poddać kontrakcji i wykonując ją aż nie będzie to możliwe. Efektywność tego algorytmu jest dodatkowo poprawiona przez algorytm identyfikacji takich podsieci w czasie wielomianowym. W rezultacie cały algorytm redukcji działa w czasie wielomianowym.

V. INFORMACJA O WYKAZYWANIU SIĘ ISTOTNĄ AKTYWNOŚCIĄ NAUKOWĄ ALBO ARTYSTYCZNĄ REALIZOWANĄ W WIĘCEJ NIŻ JEDNEJ UCZELNI, INSTYTUCJI NAUKOWEJ LUB INSTYTUCJI KULTURY, W SZCZEGÓLNOŚCI ZAGRANICZNEJ

W tej sekcji najpierw przedstawiam szybki przegląd moich pozostałych publikacji. Publikacje są zorganizowane według dziedzin. W podsekcjach od ?? do ?? prezentuję artykuły opublikowane po uzyskaniu doktoratu. Te opublikowane wcześniej znajdują się w podsekcji E. Sekcja kończy się wykazem moich wizyt na zagranicznych uniwersytetach.

A. Bioinformatics

- [A.1] Jacek Sroka, Lukasz Krupa, Andrzej M. Kierzek, Jerzy Tyszkiewicz: CalcTav - integration of a spreadsheet and Taverna workbench. **Bioinformatics** 27(18): 2618-2619 (2011)
- [A.2] Wu Huihai, von Kamp Axel, Leoncik Vytautas, Mori Wataru, Sahin Nilgun, Gevorgyan Albert, Linley Catherine, Grabowski Marek, Mannan Ahmad A., Stoy Nicholas, Stewart Graham R., Ward Lara T., Lewis David J. M., Sroka Jacek, Matsuno Hiroshi, Klamt Steffen, Westerhoff Hans V., McFadden Johnjoe, Plant



Rysunek 11: Cztery przypadki istotne dla dowodu

Nicholas J., and Kierzek Andrzej M.: MUFINS: multi-formalism interaction network simulator. **NPJ Systems Biology and Applications** vol. 2 16032. 17 Nov. 2016, doi:10.1038/npjbsa.2016.32

- [A.3] Wojciech Ptak, Andrzej M. Kierzek, Jacek Sroka: AB-QSSPN: Integration of Agent-Based Simulation of Cellular Populations with Quasi-Steady State Simulation of Genome Scale Intracellular Networks. **Petri Nets** 2016: 113-122
- [A.4] Marek Grabowski, Grzegorz Bokota, Jacek Sroka, Andrzej M. Kierzek: Verification of Dynamic Behaviour in Qualitative Molecular Networks Describing Gene Regulation, Signalling and Whole-cell Metabolism. **Fundamenta Informaticae** 160(1-2): 199-219 (2018)
- [A.5] Grzegorz Bokota, Jacek Sroka, Subhadip Basu, Nirmal Das, Pawel Trzaskoma, Yana Yushkevich, Agnieszka Grabowska, Adriana Magalska, and Dariusz Plewczynski: PartSeg: a tool for quantitative feature extraction from 3D microscopy images for dummies. **BMC Bioinformatics**, 22(1):72, 2021.
- [A.6] Soumyendu Sekhar Bandyopadhyay, Anup Kumar Halder, Piyali Chatterjee, Jacek Sroka, Mita Nasipuri, and Subhadip Basu: Analysis of large-scale human protein sequences using an efficient spark-based DBSCAN algorithm. In Debotosh Bhattacharjee, Dipak Kumar Kole, Nilanjan Dey, Subhadip Basu, and Dariusz Plewczynski, editors, **Proceedings of International Conference on Frontiers in Computing and Systems**, pages 601–609, Singapore, 2021. Springer Singapore.
- [A.7] Paulina Szymczak, Marcin Możejko, Tomasz Grzegorzek, Radosław Jurczak, Marta Bauer, Damian Neubauer, Karol Sikora, Michał Michalski, Jacek Sroka, Piotr Setny, Wojciech Kamysz, and Ewa Szczurek: Discovering hi-

ghly potent antimicrobial peptides with deep generative model HydrAMP. **Nature Communications**, 14(1):1453, <https://doi.org/10.1038/s41467-023-36994-z>, 2023.

- [A.8] Agnieszka Motyka, Aleksandra Bartnik, Aleksandra Żurko, Marta Giziewska, Paweł Gora, and Jacek Sroka: Using cellular automata to compare sars-cov-2 infectiousness in different pois and under different conditions. In Subhadip Basu, Dipak Kumar Kole, Arnab Kumar Maji, Dariusz Plewczynski, and Debotosh Bhattacharjee, editors, **Proceedings of International Conference on Frontiers in Computing and Systems**, pages 331–339, Singapore, 2023. Springer Nature Singapore.
- [A.9] Tomasz Pawłowski, Grzegorz Bokota, Georgia Lazarou, Andrzej M. Kierzek, and Jacek Sroka: Emulation of quantitative systems pharmacology models to accelerate virtual population inference in immuno-oncology. **Methods**, 223:118–126, 2024.

W [A.1] zintegrowaliśmy system Taverna Workbench z arkuszem kalkulacyjnym, aby użytkownicy zaznajomieni z arkuszami mogli używać usług sieciowych dostępnych w Tavernie i jednocześnie przetwarzać wyniki za pomocą formuł i wizualizacji w arkuszu.

W [A.2] opracowaliśmy narzędzie pozwalające na symulację modeli komórek składających się z części metabolicznej i części regulacyjnej, która jest wyrażona za pomocą sieci Petriego i reprezentuje enzymatyczną regulację metabolizmu. Takie hybrydowe podejście jest skuteczne w symulacjach dynamicznych zachowań, takich jak homeostaza kwasów żółciowych w ludzkich komórkach wątroby (hepatocytach).

[A.3] jest wcześniejszą wersją [1], która została opublikowana na konferencji.

W [A.4] opracowaliśmy zoptymalizowane narzędzie do wyczerpującej analizy przestrzeni stanów modelu QSSPN

do pewnej głębokości. Takie narzędzie może być używane podczas projektowania modeli. W przeciwieństwie do podejścia Monte Carlo używanego wcześniej, który nie dawał jednak pewności, możemy udowodnić, że pewne zachowania nie mogą wystąpić w modelu w określonej liczbie kroków. Pozwala to na odrzucenie modeli, które nie są w stanie odtworzyć obserwowanych eksperymentalnie zachowań, a także na weryfikację, że nierealistyczne biologicznie zachowania nie mogą wystąpić w symulacji. Pokazujemy również przykład, w jaki sposób te funkcje poprawiają identyfikację problemów w dużych modelach sieci.

W [A.5] stworzyliśmy PartSeg — narzędzie do bioobrazowania, które może być używane do przetwarzania obrazów 3D mikroskopii o wysokiej rozdzielczości, a w szczególności do segmentacji struktur w jądrze komórkowym. Problem nie jest prosty ze względu na gęste upakowanie i znaczące nakładanie się sygnałów. Jednocześnie inne dostępne narzędzia do segmentacji oferują stromą krzywą uczenia dla nowych użytkowników bez doświadczenia informatycznego, czego nie można powiedzieć o PartSegu. Jest to szczególnie widoczne przy hurtowym przetwarzaniu zestawów obrazów, które w innych narzędziach wymaga użycia jakiejś formy notacji programistycznej.

W [A.6] rozproszyliśmy algorytm klastrowania oparty na gęstości DBSCAN. Rozproszona implementacja została użyta do przetwarzania dużych danych z informacjami o sekwencjach białkowych człowieka przez badanie ich reprezentacji za pomocą 3-shingli. Wyniki eksperymentalne na tym zbiorze danych pokazują dobre przyspieszenie w skutek rozproszenia obliczeń i uzyskanie klastrów wysokiej jakości.

[A.7] koncentruje się na znalezieniu peptydów przeciwdrobnoustrojowych, które mogą złagodzić globalne zagrożenie zdrowia związane z opornością na środki przeciwdrobnoustrojowe. W artykule opracowaliśmy warunkowy autoenkoder wariacyjny (HydrAMP), który uczy się niskowymiarowej, ciągłej reprezentacji peptydów i uwzględnia ich właściwości przeciwdrobnoustrojowe. Model oddziela wyuczoną reprezentację peptydu od jego warunków przeciwdrobnoustrojowych, a jego kreatywnością można sterować poprzez parametry. Skutkuje to pierwszym modelem, który jest bezpośrednio optymalizowany dla różnych zadań, w tym nieograniczonego generowania analogów. Dodatkowa procedura preselekcji oparta na rankingu wygenerowanych peptydów i symulacjach dynamiki molekularnej zwiększa wskaźnik walidacji eksperymentalnej. Eksperymenty *in vitro* na pięciu szczepach bakteryjnych potwierdzają wysoką aktywność dziewięciu peptydów wygenerowanych jako analogi klinicznie istotnych prototypów, a także sześciu analogów nieaktywnego peptydu. HydrAMP umożliwia generowanie różnorodnych i skutecznych peptydów, stanowiąc krok w kierunku rozwiązania kryzysu oporności na środki przeciwdrobnoustrojowe.

W [A.8] zastosowaliśmy symulacje automatów komórkowych do porównania zakaźności Sars-Cov-2 w różnych miejscach zainteresowania, takich jak sklepy czy autobusy komunikacji publicznej oraz przy uwzględnieniu różnych warunków.

W [A.9] przedstawiliśmy nowatorskie podejście do inferencji populacji wirtualnej w immunoonkologii (IO) przy użyciu emulacji modelu ilościowej farmakologii systemowej (ang.

Quantitative systems pharmacology, QSP) i funkcji celu opartej na statystyce Kolmogorowa-Smirnowa w celu maksymalizacji zgodności symulowanych i obserwowanych klinicznych rozkładów wielkości guza. Modele QSP są coraz częściej stosowane do odkrywania celów i doboru dawek w IO. Typowe zastosowanie obejmuje wirtualne badanie kliniczne, symulację wirtualnej populacji setek instancji modelu z wejściami modelu odzwierciedlającymi indywidualną zmienność. Struktura modelu i początkowa parametryzacja opierają się na literaturze opisującej podstawową biologię, jednak kalibracja wirtualnej populacji za pomocą istniejących danych klinicznych jest często wymagana do tworzenia instancji modelu specyficznych dla populacji nowotworów i pacjentów. Ponieważ porównanie wirtualnego badania z wynikami klinicznymi wymaga setek dużych, nieliniowych ewaluacji modeli, inferencja wirtualnej populacji jest kosztowna obliczeniowo, często stając się wąskim gardłem. Próbujemy przestrzeń parametrów modelu QSP IO, aby zebrać zestaw profili czasowych wzrostu guza. Oceniamy wydajność kilku podejść uczenia maszynowego w interpolacji tych profili czasowych i tworzymy model zastępczy, który oblicza profile wzrostu guza szybciej niż oryginalny model i pozwala na badanie dziesiątek milionów wirtualnych pacjentów. Używamy modelu zastępczego do inferencji populacji wirtualnej maksymalizującej zgodność z wykresem wodospadowym badania klinicznego pembrolizumab. Uważamy, że nasze podejście ma zastosowanie nie tylko w QSP IO, ale także gdy populacje wirtualne muszą być wnioskowane dla kosztownych obliczeniowo modeli mechanistycznych.

B. Social choice theory

- [B.1] Tomasz P. Michalak, Talal Rahwan, Jacek Sroka, Andrew James Dowell, Michael J. Wooldridge, Peter McBurney, Nicholas R. Jennings: On representing coalitional games with externalities. **EC** 2009: 11-20
- [B.2] Talal Rahwan, Tomasz P. Michalak, Madalina Croitoru, Jacek Sroka, Nicholas R. Jennings: A Network Flow Approach to Coalitional Games. **ECAI** 2010: 1017-1018
- [B.3] Tomasz P. Michalak, Jacek Sroka, Talal Rahwan, Michael Wooldridge, Peter McBurney, Nicholas R. Jennings: A distributed algorithm for anytime coalition structure generation. **AAMAS** 2010: 1007-1014
- [B.4] Patrick Doherty, Tomasz P. Michalak, Jacek Sroka, Andrzej Szalas: Contextual Coalitional Games. **ICLA** 2011: 65-78
- [B.5] Talal Rahwan, Tomasz P. Michalak, Edith Elkind, Piotr Faliszewski, Jacek Sroka, Michael Wooldridge, Nicholas R. Jennings: Constrained Coalition Formation. **AAAI** 2011

Te artykuły dotyczą problemu generowania struktur koalicyjnych (ang. Coalition structure generation, CSG). Celem jest podział zbioru agentów na koalicje, tak aby zmaksymalizować całkowity zysk wszystkich koalicji.

W [B.1] rozważaliśmy problem reprezentowania gier koalicyjnych w systemach wielo-agentowych z zewnętrznymi zależnościami, czyli tam, gdzie wydajność jednej koalicji może zależeć od innych współistniejących koalicji. Proponujemy szereg nowych reprezentacji opartych na nowym pojęciu zewnętrznych zależności (ang. externalities). Pokazujemy,

że nowe reprezentacje są w pełni ekspresywne i dla wielu klas gier bardziej zwarte niż konwencjonalna reprezentacja Partition Function Game (PFG). Proponujemy również szereg podejść do rozwiązania problemu generowania struktur koalicyjnych w systemach z zewnętrznymi zależnościami. Pokazujemy, że jeśli zewnętrzne zależności charakteryzują się różnym stopniem regularności, nowe reprezentacje pozwalają na adaptację algorytmów generowania struktur koalicyjnych, które pierwotnie zostały zaprojektowane dla dziedzin bez zewnętrznych zależności, tak aby mogły być używane, gdy zewnętrzne zależności są obecne.

W [B.2] zaproponowaliśmy nowe podejście do reprezentowania gier koalicyjnych, zwane Coalition-Flow Network (CF-NET), które opiera się na uogólnieniu literatury dotyczącej przepływu w sieci. Konkretnie, ta reprezentacja opiera się na naszym spostrzeżeniu, że proces formowania koalicji można postrzegać jako problem kierowania przepływu przez sieć, w której każda krawędź ma określone ograniczenie na pojemność.

W [B.3] opracowaliśmy pierwszy zdecentralizowany algorytm do optymalnego rozwiązania problemu CSG. W naszym algorytmie niezbędne obliczenia są rozdzielane między agentów, zamiast być wykonywane centralnie przez jednego agenta. Algorytm zawiera również interesujące techniki filtrowania wejścia i zmniejszania obciążenia komunikacyjnego między agentami.

W [B.4] zaproponowaliśmy nowe podejście do modelowania scenariuszy, w których gry koalicyjne są kontekstualizowane za pomocą wyrażeń logicznych reprezentujących środowisko i dodatkowe części stanu, a rozkłady prawdopodobieństwa są umieszczone w przestrzeni kontekstów w celu modelowania stochastycznej natury scenariuszy. Definiujemy również i pokazujemy, jak obliczać oczekiwaną wartość Shapleya w takich grach w sposób efektywny obliczeniowo. Użyteczność podejścia ilustrujemy na przykładzie zastosowania autonomicznych robotów do wsparcia w nagłych wypadkach.

Wreszcie, w [B.5] zauważyliśmy, że w wielu rzeczywistych zastosowaniach istnieją ograniczenia dotyczące możliwych koalicji, na przykład, niektórzy agenci mogą nie móc być razem lub struktura koalicyjna musi się składać z koalicji o tej samej wielkości. W artykule przedstawiamy pierwsze usystematyzowane badanie formowania koalicji w tych sytuacjach (ang. Constrained Coalition Formation, CCF). Proponujemy ogólny framework dla tego problemu i identyfikujemy ważną klasę ustawień CCF, gdzie ograniczenia określają, które grupy agentów powinny/nie powinny współpracować. Na koniec opracowujemy algorytm do generowania optymalnej (maksymalizującej dobrobyt) ograniczonej struktury koalicyjnej.

C. Bazy danych

- [C.1] Marek Grabowski, Jan Hidders, Jacek Sroka: Representing MapReduce Optimisations in the Nested Relational Calculus. **BNCOD** 2013: 175-188
- [C.2] Jacek Sroka, Adrian Panasiuk, Krzysztof Stencel, Jerzy Tyszkiewicz: Translating Relational Queries into Spreadsheets. **IEEE Trans. Knowl. Data Eng.** 27(8): 2291-2303 (2015)

- [C.3] Jerzy Sikora, Jacek Sroka, Jerzy Tyszkiewicz: Programming Communication with the User in Multiplatform Spreadsheet Applications. **STAF Workshops** 2016: 356-371
- [C.4] Jacek Sroka, Marek Rogala, Michal Adamczyk, Jan Hidders: A Datalog Engine for Iterative Graph Algorithms on Large Clusters. **DSDIS** 2015: 113-114
- [C.5] Jerzy Sikora, Jacek Sroka, Jerzy Tyszkiewicz: Spreadsheet as a Multi-platform Mobile Application. **MOBILESoft** 2015: 140-141
- [C.6] Jacek Sroka, Artur Lesniewski, Mirosław Kowaluk, Krzysztof Stencel, Jerzy Tyszkiewicz: Towards minimal algorithms for big data analytics with spreadsheets. **BeyondMR@SIGMOD** 2017: 1:1-1:4
- [C.7] Jacek Sroka and Jerzy Tyszkiewicz: Aggregating over Dominated Points by Sorting, Scanning, Zip and Flat Maps. In Inge Li Gørtz, Martin Farach-Colton, Simon J. Puglisi, and Grzegorz Herman, editors, *31st Annual European Symposium on Algorithms (ESA 2023)*, volume 274 of **Leibniz International Proceedings in Informatics (LIPIcs)**, pages 96:1–96:13, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

W [C.1] zastosowaliśmy model MapReduce do przetwarzania danych hierarchicznych z zagnieżdżonymi kolekcjami, takich jak dane przechowywane w formatach JSON lub XML, ale o ograniczonej głębokości zagnieżdżenia, jak to zwykle ma miejsce w zagnieżdżonym modelu relacyjnym. Systemy analizy danych z tamtego okresu często opierały się na formalizmach ad-hoc do reprezentowania planów wykonania zapytań i optymalizacji ich wykonania. Argumentowaliśmy, że Nested Relational Calculus (NRC) stanowi ogólny, elegancki i efektywny sposób opisu i badania tych optymalizacji.

W [C.2] pokazaliśmy, że formuły arkusza kalkulacyjnego mogą być używane do wyrażania algebry relacji i zapytań SQL. Arkusze kalkulacyjne są jednymi z najczęściej używanych aplikacji do przechowywania i analizy danych. Wykazaliśmy, że arkusz kalkulacyjny może pełnić rolę silnika relacyjnej bazy danych, bez użycia makr czy wbudowanych języków programowania, jedynie za pomocą formuł. Osiąga się to poprzez implementację wszystkich operatorów algebry relacji za pomocą funkcji arkusza kalkulacyjnego. Zaimplementowaliśmy również praktyczne konstrukcje SQL, które wykraczają poza algebrę jak: sortowanie, domknięcie przechodnie SQL:1999 i prawdziwe wartości NULL, które dokładnie odzwierciedlają trójwartościową logikę SQL. W tym artykule oferujemy również arkusz kalkulacyjny do sortowania o złożoności $O(n \log^2 n)$, używający niestalej liczby wierszy oraz, co zaskakujące, wyszukiwanie w głąb (DFS) i przeszukiwanie wszerek (BFS) w grafach.

W [C.3] i [C.5] idziemy o krok dalej i pokazujemy, że w arkuszu kalkulacyjnym można opracować proste aplikacje do zbierania i walidacji danych. Raportujemy nasze doświadczenia z tworzeniem interfejsu użytkownika wieloplatformowej aplikacji arkuszowej dla archeologów pracujących w terenie, którzy potrzebują, aby dane z wprowadzanych grafów były sprawdzane pod kątem zawierania cykli.

W [C.4] pokazaliśmy jak użyć język Datalog rozszerzonego o agregację jako języka zapytań do przetwarzania dużych

grafów, np. w sieci web lub sieciach społecznościowych. Opracowaliśmy rozszerzenie do Apache Spark, który jest najpopularniejszym systemem do przetwarzania dużych danych na klastrach komputerów, z możliwością wykonywania zapytań Datalog. Takie podejście umożliwia wyrażanie algorytmów grafowych w dobrze zbadanym deklaratywnym języku zapytań i wykonywanie ich na istniejącej i dojrzałej infrastrukturze do przetwarzania rozproszonego. Jednocześnie dane przetwarzane za pomocą zapytań Datalog są w pełni zintegrowane z mechanizmem buforowania Spark, a zapytania Datalog mogą być częścią większych algorytmów iteracyjnych.

W [C.6] opracowaliśmy idealnie zrównoważony rozproszony algorytm do specyfikacji obliczeń MapReduce dla dużej liczby zapytań zakresowych. Zapytania zakresowe są popularną klasą zapytań do baz danych, ale jako motywujący przykład dla tych badań pokazaliśmy, że nasze algorytmy mogą być używane do rozproszenia obliczeń zdefiniowanych w formie arkusza kalkulacyjnego na zbiorach danych przekraczających wymiary siatki arkusza kalkulacyjnego o rzędy wielkości. Może to zapewnić dostępny dla mas interfejs użytkownika dla chmur obliczeniowych, ponieważ duże zbiory danych są często przechowywane jako pliki CSV, a formuły arkusza kalkulacyjnego są najpopularniejszą formą analizy danych. Prace te kontynuujemy w [C.7], gdzie ustalamy, że agregacja prefiksowa wystarcza do wyrażenia agregacji nad zdominowanymi punktami w większej liczbie wymiarów, mimo że ta ostatnia jest daleko idącym uogólnieniem pierwszej. Wiele problemów, które wiadomo, że można wyrazić za pomocą agregacji nad zdominowanymi punktami, staje się również możliwych do wyrażenia za pomocą agregacji prefiksowej. Opieramy się na małym zestawie operacji prymitywnych, które gwarantują łatwy transfer do różnych architektur rozproszonych i pewne pożądane właściwości implementacji.

D. Inne publikacje

- [D.1] Magdalena Dukielska, Jacek Sroka: JavaSpaces NetBeans: a Linda workbench for distributed programming course. **ITiCSE** 2010: 23-27
- [D.2] Jacek Sroka, Piotr Włodarczyk and Łukasz Krupa, Jan Hidders DFL designer — collection-oriented scientific workflows with Petri nets and nested relational calculus. **WANDS** 2010, Indianapolis, USA
- [D.3] Jacek Sroka, Piotr Chrzastowski-Wachtel, Jan Hidders: On Generating *-Sound Nets with Substitution. **ACSD** 2011: 3-12
- [D.4] Krzysztof Lecki, Jerzy Tyszkiewicz, Jacek Sroka: Structural Induction as a Method to Distribute the Generation of a Trace Language Representation for Complex Systems. **ATAED@Petri Nets/ACSD** 2017: 55-70

W [D.1] opracowaliśmy IDE do wspomagania zajęć z programowania równoległego. Wspiera ono programowanie w technologii JavaSpaces, która implementuje zasady Lindy w Javie. JavaSpaces NetBeans (JSN) jest narzędziem dydaktycznym do praktycznych zadań podczas kursów programowania rozproszonego. Ukrywa zaawansowane aspekty konfiguracji JavaSpaces i pozwala studentom skupić się na koordynacji między procesami. Ważnym elementem JSN jest rozproszony

debugger, który pomaga analizować typowe problemy, takie jak zakleszczenie czy zagłodzenie, i sprawia, że zajęcia są bardziej interesujące.

W [D.2] stworzyliśmy DFL definger, który jest narzędziem do definiowania collection-oriented scientific workflows opierających się na notacji DFL, która z kolei opiera się na uznanych formalizmach z baz danych oraz modelowania przepływów prac to znaczy sieciach Petriego oraz nested relational calculus.

[D.3] jest wcześniejszą wersją [4], która została opublikowana na konferencji.

W [D.4] stworzyliśmy metodę rozproszenia generacji wszystkich sekwencji zdarzeń dla zadanej sieci Petriego, które nie przekraczają określonego limitu długości. Kluczową zaletą tej metody jest możliwość rozproszenia procesu generowania i zastosowanie technik przetwarzania Big Data. Sieć jest dekomponowana na funkcjonalne fragmenty, z których każdy jest niezależnie symulowany w celu wygenerowania zestawu swoich sekwencji. Wyniki są następnie łączone, aby uzyskać sekwencje dla całej sieci. Proces ten może być wykonany za pomocą indukcji strukturalnej i ułatwia ponowne użycie wyników częściowych. Aby ograniczyć ilość danych przesyłanych podczas obliczeń, co jest czynnikiem krytycznym, reprezentujemy te sekwencje w zwartej formie śladów Mazurkiewicza, które łączą wszystkie sekwencje różniące się zamianą kolejności niezależnych zdarzeń. W związku z tym pojedynczy ślad odpowiada wszystkim rozproszonym przebiegom sieci, gdzie zadania, które mogą być wykonywane równocześnie, są zapisywane we wszystkich możliwych kolejnościach. W artykule przedstawiamy konstruktywny algorytm do synchronizacji śladów. Nasze narzędzie może być użyteczne w process miningu, a w szczególności w sprawdzaniu zgodności (ang. conformance checking).

E. Publications preceding PhD

- [E.1] Jacek Sroka, Jan Hidders: Towards a Formal Semantics for the Process Model of the Taverna Workbench. Part I. **Fundamenta Informaticae** 92(3): 279-299 (2009)
- [E.2] Jacek Sroka, Jan Hidders: Towards a Formal Semantics for the Process Model of the Taverna Workbench. Part II. **Fundamenta Informaticae** 92(4): 373-396 (2009)
- [E.3] Jan Hidders, Natalia Kwasnikowska, Jacek Sroka, Jerzy Tyszkiewicz, Jan Van den Bussche: DFL: A dataflow language based on Petri nets and nested relational calculus. **Information Systems** 33(3): 261-284 (2008)
- [E.4] Jan Hidders, Jacek Sroka: Towards a Calculus for Collection-Oriented Scientific Workflows with Side Effects. **OTM Conferences (1)** 2008: 374-391
- [E.5] Jan Hidders, Natalia Kwasnikowska, Jacek Sroka, Jerzy Tyszkiewicz, Jan Van den Bussche: A Formal Model of Dataflow Repositories. **DILS** 2007: 105-121
- [E.6] Jacek Sroka, Grzegorz Kaczor, Jerzy Tyszkiewicz, Andrzej M. Kierzek: XQTav: an XQuery processor for Taverna environment. **Bioinformatics** 22(10): 1280-1281 (2006)
- [E.7] Jan Hidders, Natalia Kwasnikowska, Jacek Sroka, Jerzy Tyszkiewicz, Jan Van den Bussche: Petri Net + Nested

Relational Calculus = Dataflow. **OTM Conferences (1)** 2005: 220-237

W [E.1] i [E.1] oraz w pracy konferencyjnej [E.4] zbadaliśmy formalną semantykę Taverna 1. System został znacząco przebudowany w Taverna 2, co wymagało zdefiniowania w [6] jego semantyki w inny sposób.

W [E.3] oraz w pracy konferencyjnej [E.7] połączyliśmy sieci Petriego i Nested Relational Calculus w DFL, który jest językiem do specyfikacji przepływów danych, to znaczy przepływów prac skoncentrowanych na przetwarzaniu zagnieźdzonych kolekcji ustrukturyzowanych danych.

W [E.6] stworzyliśmy narzędzie hybrydowe umożliwiające wykorzystanie procesorów Taverna 1 wraz z językiem skryptowym opartym na XQuery i SQL.

W [E.5] opracowaliśmy formalny model danych dla repozytoriów przepływów danych, czyli baz danych zawierających przepływy danych i ich różne uruchomienia. Nasz model zawierał staranne sformalizowanie takich funkcji jak złożona manipulacja danymi, wywołania zewnętrznych usług, podprzepływy danych oraz iteracja wartości wynikowych.

F. Wizyty na innych uczelniach

Od 2018-02-01 do 2018-02-22 odwiedziłem National Institute of Technical Teacher Training and Research w Kalkucie, Indie, gdzie pracowałem jako **niezależny badacz** i organizowałem prezentacje moich wyników.

W sierpniu-wrześniu 2004 odwiedziłem Limburgs Universitair Centrum, Belgia, w ramach **wizyty badawczej**.

Liczne, inne wizyty są wymienione w dokumencie *Wykaz osiągnięć naukowych albo artystycznych, stanowiących znaczny wkład w rozwój określonej dyscypliny*.

W latach 2018-2022 byłem szefem zespołu tworzącego system Wsparcia Organów Wyborczych w Krajowym Biurze Wyborczym i doradzałem Krajowemu Biuru Wyborczemu w kwestiach związanych z cyberbezpieczeństwem oraz wyborami online. Współpracowałem z badaczami specjalizującymi się w tych dziedzinach, analizowałem istniejące systemy i inicjatywy w innych krajach oraz brałem udział w międzynarodowych spotkaniach, na przykład organizowanych przez Parlament Europejski i NATO.

VI. INFORMACJA O OSIĄGNIĘCIACH DYDAKTYCZNYCH, ORGANIZACYJNYCH ORAZ POPULARYZUJĄCYCH NAUKĘ LUB SZTUKĘ

A. Dydaktyka

Wypromowałem ponad 50 prac magisterskich oraz ponad 100 licencjatów.

Przygotowałem i prowadziłem następujące wykłady na Wydziale Matematyki, Informatyki i Mechaniki:

- Uczenie maszynowe w dużej skali,
- Przetwarzanie dużych danych i programowanie na klastrach,
- Języki i narzędzia programowania III,
- Programowanie obiektowe,
- Zaawansowane bazy danych 2: grafy, strumienie, logika,
- Java Enterprise Edition,

- Tworzenie aplikacji wielowarstwowych.

Przygotowałem i prowadziłem następujące zajęcia/laboratoria na Wydziale Matematyki, Informatyki i Mechaniki:

- Zespołowy projekt programistyczny,
- Przetwarzanie dużych danych i programowanie na klastrach,
- Uczenie maszynowe w dużej skali,
- Statystyczne uczenie maszynowe,
- Bazy danych,
- Statystyczna analiza danych,
- Programowanie obiektowe,
- Zaawansowane bazy danych 2: grafy, strumienie, logika,
- Java Enterprise Edition,
- Tworzenie aplikacji wielowarstwowych,
- Nowoczesne języki programowania obiektowego i platformy programistyczne na przykładzie Scala i NetBeans,
- Informatyka,
- Tworzenie systemów informacyjnych w Javie,
- Inżynieria oprogramowania,
- Programowanie aspektowe,
- XML i nowoczesne techniki zarządzania treścią.

Sukcesy moich studentów:

- Wypromowałem pracę licencjacką „System obsługi parkingów”, która uzyskała drugie miejsce w ogólnopolskim konkursie na najlepszą pracę licencjacką/magisterską związaną z technologiami Java za rok 2010.
- Byłem mentorem (oficjalny opiekun) zespołu studentów Uniwersytetu Warszawskiego, który zakwalifikował się do finałów prestiżowego konkursu Imagine Cup 2008.
- Moja studentka Magdalena Dukielska zdobyła w 2008 roku grant o wartości 11 500 USD od firmy Sun Microsystems na stworzenie, pod moim nadzorem, środowiska programistycznego wspomagającego naukę programowania rozproszonego.

Współorganizowałem wykłady PhD Open na Wydziale Matematyki, Informatyki i Mechaniki.

W 2010 byłem współautorem książki Informatyka Gospodarcza, Tom III, będącej podręcznikiem na wydziałach zarządzania.

B. Wykłady, seminaria i zaproszone tutoriale

- International Conference on Frontiers in Computing and Systems (COMSYS-2020) od 2020-01-13 do 2020-01-15, Indie, **zaproszony tutorial**;
- Uniwersytet Hasselt, od 2019-03-31 do 2019-04-06, Belgia, **seria wykładów zaproszonych 12h**;
- School of Computing at Newcastle University, od 2017-06-25 do 2017-06-28, Wielka Brytania, **zaproszone seminarium**;
- Vrije Universiteit Brussel, od 2017-02-19 do 2017-02-24, Belgia, **seria wykładów zaproszonych 6h**;

C. Osiągnięcia organizacyjne: warsztaty SWEET i BeyondMR

Współorganizowałem serię warsztatów na czołowej konferencji bazodanowej SIGMOD. Dwie edycje ACM SIGMOD

Workshop on Scalable Workflow Execution Engines and Technologies (SWEET) odbyły się razem z konferencją SIGMOD w 2012 [C.1] i 2013 [C.2]. W 2015 roku SWEET połączył się z warsztatami Algorithms and Systems for MapReduce and Beyond (BeyondMR), które w tym roku odbywały się wraz z konferencją EDBT. Następnie połączone warsztaty przeniosły się na SIGMOD, gdzie odbyły się trzy kolejne udane edycje w 2016 [C.3], 2017 [C.4] i 2018 [C.5]. BeyondMR współorganizowałem z prof. Jeffreyem D. Ullmanem.

W wyniku pozytywnego przyjęcia BeyondMR opublikowaliśmy raporty w SIGMOD Record [C.7]-[C.9] i zorganizowaliśmy dwa numery specjalne w czasopismach *Fundamenta Informaticae* [C.10] i *Future Generation Comp. Syst.* [C.11].

- [C.1] Jan Hidders, Paolo Missier, Jacek Sroka: Proceedings of the 1st ACM SIGMOD Workshop on Scalable Workflow Execution Engines and Technologies, **SWEET@SIGMOD** 2012, Scottsdale, AZ, USA, May 20, 2012. ACM 2012, ISBN 978-1-4503-1876-1
- [C.2] Jan Hidders, Paolo Missier, Jacek Sroka: Proceedings of the 2nd ACM SIGMOD Workshop on Scalable Workflow Execution Engines and Technologies, **SWEET@SIGMOD** 2013, New York, New York, USA, June 23, 2013. ACM 2013, ISBN 978-1-4503-2349-9
- [C.3] Foto N. Afrati, Jacek Sroka, Jan Hidders: Proceedings of the 3rd ACM SIGMOD Workshop on Algorithms and Systems for MapReduce and Beyond, **BeyondMR@SIGMOD** 2016, San Francisco, CA, USA, July 1, 2016. ACM 2016, ISBN 978-1-4503-4311-4
- [C.4] Foto N. Afrati, Jacek Sroka: Proceedings of the 4th ACM SIGMOD Workshop on Algorithms and Systems for MapReduce and Beyond, **BeyondMR@SIGMOD** 2017, Chicago, IL, USA, May 19, 2017. ACM 2017, ISBN 978-1-4503-5019-8
- [C.5] Foto N. Afrati, Jacek Sroka, Ke Yi, Jan Hidders: Proceedings of the 5th ACM SIGMOD Workshop on Algorithms and Systems for MapReduce and Beyond, **BeyondMR@SIGMOD** 2018, Houston, TX, USA, June 15, 2018. ACM 2018
- [C.6] Jacek Sroka, Jan Hidders, Paolo Missier: Report from the second workshop on scalable workflow enactment engines and technology (SWEET'13). **SIGMOD Record** 42(4): 73-77 (2013)
- [C.7] Jan Hidders, Jacek Sroka, Paolo Missier: Report from the first workshop on scalable workflow enactment engines and technology (SWEET'12). **SIGMOD Record** 41(4): 60-64 (2012)
- [C.8] Foto N. Afrati, Jan Hidders, Christopher Ré, Jacek Sroka, Jeffrey D. Ullman: Report from the third workshop on Algorithms and Systems for MapReduce and Beyond (BeyondMR'16). **SIGMOD Record** 46(2): 43-48
- [C.9] Foto N. Afrati, Jan Hidders, Paraschos Koutris, Jacek Sroka, Jeffrey D. Ullman: Report from the Fourth Workshop on Algorithms and Systems for MapReduce and Beyond (BeyondMR '17). **SIGMOD Record** 46(4): 44-48
- [C.10] Jan Hidders, Paolo Missier, Jacek Sroka, Jan Van den Bussche: Preface. **Fundamenta Informaticae** 128(3)

[C.11] Jan Hidders, Paolo Missier, Jacek Sroka: Recent advances in Scalable Workflow Enactment Engines and Technologies. **Future Generation Comp. Syst.** 46: 1-2 (2015)

VII. OPRÓCZ KWESTII WYMIENIONYCH W PKT. 1-6, WNIOSKODAWCA MOŻE Podać INNE INFORMACJE, WAŻNE Z JEGO PUNKTU WIDZENIA, DOTYCZĄCE JEGO KARIERY ZAWODOWEJ

A. Praktyczne doświadczenie w ważnych projektach

W latach 2018-2022 pracowałem jako szef zespołu rozwijającego system Wsparcia Organów Wyborczych w Krajowym Biurze Wyborczym Rzeczypospolitej Polskiej. System, który opracowałem wraz z moim zespołem, był z powodzeniem stosowany przy organizacji wszystkich głównych wyborów w Polsce, w tym wyborów Prezydenta Rzeczypospolitej Polskiej, wyborów parlamentarnych, wyborów do Parlamentu Europejskiego oraz wyborów samorządowych. Nadzorowałem wdrożenie systemu i doradzałem Krajowemu Biuru Wyborczemu oraz Państwowej Komisji Wyborczej. Osobiście opracowałem również część systemu.

W latach 2023 i 2024 nadzorowałem tworzenie internetowego systemu informacyjnego dla Wydziału Matematyki, Informatyki i Mechaniki na Uniwersytecie Warszawskim.

Te obowiązki wydłużyły przygotowanie mojego wniosku habilitacyjnego.

B. Nagrody

- Nagroda Rektora UW za osiągnięcia naukowe 2017
- Nagroda za najlepszą pracę na międzynarodowej konferencji Conference on Applications and Theory of Petri Nets and Concurrency 2016 za pracę [A.3]
- *Highly accessed* status dla pracy [5]
- Laureat konkursu na stypendia dla najlepszych młodych doktorów w ramach projektu *Nowoczesny Uniwersytet — kompleksowy program wsparcia dla doktorantów i kadry dydaktycznej Uniwersytetu Warszawskiego* współfinansowanego z Europejskiego Funduszu Społecznego w ramach Programu Operacyjnego Kapitał Ludzki (listopad 2011). Mój wniosek został oceniony na 60 z 60 możliwych punktów.
- Nagroda JM Rektor UW indywidualna za osiągnięcia naukowe w listopadzie 2009.

LITERATURA

- [Bel21] Khalid Belhajjame. On the anonymization of workflow provenance without compromising the transparency of lineage. *J. Data and Information Quality*, 14(1), dec 2021.
- [BLR⁺22] Fahima Amin Bhuyan, Shiyong Lu, Robert Reynolds, Jia Zhang, and Ishtiaq Ahmed. A security framework for scientific workflow provenance access control policies. *IEEE Transactions on Services Computing*, 15(1):97–109, 2022.
- [CGL13] Piotr Chrzastowski-Wachtel, Paweł Golab, and Bartosz Lewinski. Sound recoveries of structural workflows with synchronization. In *Proceedings of the 22nd International Workshop on Concurrency, Specification and Programming, Warsaw, Poland*, pages 73–87, 2013.
- [CW06a] Piotr Chrzastowski-Wachtel. Determining sound markings in structured nets. *Fundam. Inf.*, 72(1-3):65–79, April 2006.
- [CW06b] Piotr Chrzastowski-Wachtel. Determining sound markings in structured nets. *Fundam. Inf.*, 72(1-3):65–79, April 2006.

- [CWBH⁺03] Piotr Chrzastowski-Wachtel, Boualem Benatallah, Rachid Hamadi, Milton O'Dell, and Adi Susanto. A top-down Petri net-based approach for dynamic workflow modeling. In *Proceedings of BPM'03*, pages 336–353. Springer-Verlag, 2003.
- [DE05] Jorg Desel and Javier Esparza. *Free Choice Petri Nets (Cambridge Tracts in Theoretical Computer Science)*. Cambridge University Press, New York, NY, USA, 2005.
- [DRM⁺12] Marlon Dumas, Marcello La Rosa, Jan Mendling, Raul Mäesalu, Hajo A. Reijers, and Nataliia Semenenko. Understanding business process models: The costs and benefits of structuredness. In Jolita Ralyté, Xavier Franch, Sjaak Brinkkemper, and Stanislaw Wrycza, editors, *Advanced Information Systems Engineering - 24th International Conference, CAiSE 2012, Gdansk, Poland, June 25-29, 2012. Proceedings*, volume 7328 of *Lecture Notes in Computer Science*, pages 31–46. Springer, 2012.
- [FFK⁺11] Dirk Fahland, Cédric Favre, Jana Koehler, Niels Lohmann, Hagen Völzer, and Karsten Wolf. Analysis on demand: Instantaneous soundness checking of industrial business process models. *Data Knowl. Eng.*, 70(5):448–466, May 2011.
- [FF06] Ch. Flender and Th. Freytag. Visualizing in the soundness of workflow nets. In *Bericht 267, Tagungsband des 13. Workshops Algorithmen und Werkzeuge für Petri-Netze, AWPN'06, FBI-HH-B-267/06*, pages 47–53, Sep. 06.
- [Gol14] Robert Gold. Reductions of control flow graphs. *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 8(3):427–434, 2014.
- [LAB⁺06] Bertram Ludäscher, Ilkay Altintas, Chad Berkley, Dan Higgins, Efrat Jaeger, Matthew Jones, Edward A. Lee, Jing Tao, and Yang Zhao. Scientific workflow management and the kepler system. *Concurrency and Computation: Practice and Experience*, 18(10):1039–1065, 2006.
- [MAB⁺10] Grzegorz Malewicz, Matthew H. Austern, Aart J.C Bik, James C. Dehnert, Ilan Horn, Naty Leiser, and Grzegorz Czajkowski. Pregel: a system for large-scale graph processing. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, SIGMOD '10*, page 135–146, New York, NY, USA, 2010. Association for Computing Machinery.
- [MS03] Radhakrishnan Mahadevan and Christophe H Schilling. The effects of alternate optimal solutions in constraint-based genome-scale metabolic models. *Metabolic engineering*, 5(4):264–276, 2003.
- [Mur89] T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, Apr 1989.
- [PGBD12] Artem Polyvyanyy, Luciano García-Bañuelos, and Marlon Dumas. Structuring acyclic process models. *Information Systems*, 37(6):518 – 538, 2012.
- [Pto14] Claudius Ptolemaeus, editor. *System Design, Modeling, and Simulation using Ptolemy II*. Ptolemy.org, 2014.
- [PWW11] Artem Polyvyanyy, Matthias Weidlich, and Mathias Weske. Connectivity of workflow nets: the foundations of stepwise verification. *Acta Inf.*, 48(4):213–242, 2011.
- [SBDM⁺24] Florian Schintke, Khalid Belhajjame, Ninon De Mecquenem, David Frantz, Vanessa Guarino, Marcus Hilbrich, Fabian Lehmann, Paolo Missier, Rebecca Sattler, Jan Sparka, Daniel Speckhard, Hermann Stolte, Anh Vu, and Ulf Leser. Validity constraints for data analysis workflows. *Future Generation Computer Systems*, 157, 03 2024.
- [SH17] Jacek Sroka and Jan Hidders. Finding AND-OR hierarchies in workflow nets. *CoRR*, abs/1701.02199, 2017.
- [VBvdA01] H. M. W. Verbeek, T. Basten, and W. M. P. van der Aalst. Diagnosing workflow processes using woflan. *The Computer Journal*, 44(4):246–279, 2001.
- [vdA98] Wil M. P. van der Aalst. The application of Petri nets to workflow management. *Journal of Circuits, Systems, and Computers*, 8(1):21–66, 1998.
- [Ver04] Henricus M. W. Verbeek. *Verification of WF-nets*. PhD thesis, Eindhoven University of Technology, Eindhoven, The Netherlands, 2004. PhD thesis.
- [vHSV03] Kees M. van Hee, Natalia Sidorova, and Marc Voorhoeve. Soundness and separability of workflow nets in the stepwise refinement approach. In Wil M. P. van der Aalst and Eike Best, editors, *ICATPN*, volume 2679 of *Lecture Notes in Computer Science*, pages 337–356. Springer, 2003.
- [ZSF17] Zhao Zhang, Evan R. Sparks, and Michael J. Franklin. Diagnosing machine learning pipelines with fine-grained lineage.

In *Proceedings of the 26th International Symposium on High-Performance Parallel and Distributed Computing, HPDC '17*, page 143–153, New York, NY, USA, 2017. Association for Computing Machinery.



PODPIS ZAUFANY

JACEK
SROKA

24.09.2024 18:34:17 [GMT+2]

Dokument podpisany elektronicznie
podpisem zaufanym