Michał Horodelski

Wyznaczanie reprezentatywnego fragmentu współdziałania systemu obiektów

Rozprawa doktorska

Promotor: **prof. dr hab. Józef Winkowski**

Promotor pomocniczy: **dr Piotr Filipkowski**
Szkoła Główna Handlowa w Warszawie

Instytut Podstaw Informatyki
Polskiej Akademii Nauk
Warszawa, 2019

Przedmowa

Pragnę podziękować mojej rodzinie za wsparcie i wyrozumiałość w okresie tworzenia rozprawy doktorskiej, mojej żonie Marcie oraz synowi Grzegorzowi, którzy wspierali mnie wytrwale do ostatniego dnia tworzenia rozprawy.

Dziękuję mojemu promotorowi Profesorowi Józefowi Winkowskiemu za zainteresowanie mnie tematyką opisu działania systemów współbieżnych, za czas poświęcony na dyskusje i rozważania nad pracą, za cierpliwość oraz cenne uwagi do pracy.

Dziękuję także promotorowi pomocniczemu, Doktorowi Piotrowi Filipkowskiemu za inspirację do zastosowań opracowanych metod rozprawy, za poświęcony czas na rozważania nad treściami rozprawy, za owocne dyskusje często do późnego wieczoru oraz za cenne wskazówki podnoszące jakość pracy.

Dziękuję także Profesorowi Andrzejowi Janickiemu za inspirację i zainicjowanie mojego procesu doktoryzowania oraz za zdobyte kompetencje podczas pracy w założonej przez Profesora Katedrze Technologii Społeczeństwa Informacyjnego na Wydziale Nauk Przyrodniczych (teraz Wydziale Matematyki, Informatyki i Architektury Krajobrazu) Katolickiego Uniwersytetu Lubelskiego Jana Pawła II w Lublinie.

Chciałbym także podziękować Zespołowi Teorii Systemów Rozproszonych i Obliczeniowych IPIPAN w Warszawie za zaproszenia na Seminaria oraz za cenne uwagi i wskazówki.

Spis treści

Streszczenie **9**

Abstract **11**

Wstęp **13**

1 O pewnych systemach obiektów i ich działaniu **15**
1.1 Modele systemów, ich zachowania i związana z tym problematyka . . . 15
1.2 Związek wybranego modelu z innymi znanymi modelami 20
1.2.1 Sieci z ograniczeniami czasowymi dla łuków i żetonów 20
1.2.2 Sieci z ograniczeniami czasowymi do przejść ze stoperami 22
1.2.3 Czasowy system tranzycyjny 27

2 Reprezentatywny fragment rozgałęzionego procesu **27**
2.1 Strukturalny aspekt zachowania 28
2.2 Aspekt czasowy zachowania 29
2.2.1 Funkcja przyporządkowująca zdarzeniom momenty wykonania w symbolicznych rozgałęzionych procesach 32
2.3 Symboliczne rozwinięcie dla PSwPN 32
2.4 Prefiks zupełny rozwinięcia TPN 33
2.4.1 Równoważność przyczynowej przyszłości rozgałęzionych procesów (równoważność stanów) 34
2.4.2 Zdarzenie odcięcia 38
2.4.3 Reprezentatywny fragment 38

3 Reprezentatywny fragment grafu stref **39**
3.1 Dyskretyzacja zachowania 39
3.1.1 TTS jako model zachowania dla TPN 39
3.1.2 Dyskretyzacja TTS bazująca na strefach 40
3.1.3 Aproksymacja strefy 46
3.2 Weryfikacja modelowa własności temporalnych zachowania 48
3.2.1 Syntaktyka logiki TCTL dla TPN 48
3.2.2 Semantyka logiki TPN-TCTL 49
3.2.3 Weryfikacja własności systemów TPN w praktyce 49

4 Proponowane rozwiązanie - reprezentatywny fragment **52**
4.1 Powiązania pomiędzy modelami zachowania RPC i TTS 53
4.2 Algorytm konstruowania reprezentatywnego rozgałęzionego procesu czasowego (modelu RPC) 53
4.3 Algorytm konstruowania reprezentatywnego czasowego systemu tranzycyjnego (modelu TTS) 63
4.4 Dodatkowy warunek momentu stopu w algorytmie konstruowania modelu TTS 70

4.5 Logika TdPN-TCTL(TTS) do opisu własnosci zachowań modelu systemu 71

5 Ocena jakości proponowanego rozwiązania 72
5.1 Algorytmy weryfikacji własności modelu systemu 72
5.1.1 Algorytm weryfikacji własność $EF_I\varphi$ 73
5.1.2 Algorytm weryfikacji własność $EG_I\varphi$ 74
5.1.3 Algorytm weryfikacji własności $AF_I\varphi$ i $AG_I\varphi$ 76
5.1.4 Algorytm weryfikacji własności ograniczonej reakcji 76
5.2 Analiza algorytmu generującego graf stref (TTS) 78
5.3 Analiza algorytmów weryfikacji własności $EF_I\varphi$, $EG_I\varphi$, $AF_I\varphi$, $AG_I\varphi$. 79
5.4 Analiza algorytmów weryfikacji własności ograniczonej reakcji 80

6 Badania symulacyjne i weryfikacja własności wybranego modelu systemu c-TdPN 81
6.1 Systemy z synchronizacją dostępu do danych z dodatkowymi ograniczeniami 83
6.2 Analiza wybranych własności opisanych w TdPN-TCTL 84
6.2.1 Wariant nr 1 modelu systemu 84
6.2.2 Wariant nr 2 modelu systemu 85
6.2.3 Wariant nr 3 modelu systemu 85
6.2.4 Wariant nr 4 modelu systemu 86
6.3 Analiza przydatności miejsc odczytu 86
6.4 Analiza porównawcza reprezentatywnych przestrzeni stanów (TAPAAL) 88
6.5 Badanie eksperymentalne własności $EF_I\varphi$ algorytmu konstruowania modelu TTS z dodatkowym warunkiem momentu stopu 91
6.6 Wpływ wybranej aproksymacji na weryfikację własności modelu systemu 93
6.7 Analiza wpływu wzrostu parametrów modelu systemu na środowisko symulacyjne do konstruowania reprezentatywnego TTS 96

Zakończenie 97

Literatura 100

Spis rysunków 104

Spis tabel 105

Załączniki 106

A Systemy transakcyjne i łańcuch bloków 106
A.1 Opis modelu systemu 106
A.2 Analiza wybranych własności opisanych w TdPN-TCTL 108
A.3 Analiza porównawcza reprezentatywnych przestrzeni stanów (TAPAAL) 109

B Procedury Państwowego Ratownictwa Medycznego **110**
B.1 Opis modelu systemu 110
B.2 Określenie parametrów i zainicjowanie modelu systemu 113
B.3 Reprezentatywny fragment rozgałęzionego procesu czasowego 114
B.4 Analiza stanów zapisanych w reprezentatywnym fragmencie do zadanego momentu 118

Streszczenie

Niniejsza rozprawa dotyczy metod opisu i analizy działania systemów współbieżnych. Wybranym modelem systemu jest czasowo-kontekstowa sieć Petriego. W tym modelu obserwowane są momenty pojawienia się i usuwania żetonów w miejscach, z których prowadzą łuki do tranzycji. Specyfikacja modelu polega na założeniu, że żetony są dostępne tylko w przyporządkowanych łukom przedziałach czasu. Dla systemów reprezentowanych przez takie modele zdefiniowano ich zachowania i zbadano własności tych zachowań.

Rozprawa uzupełnia poszukiwania reprezentatywnych fragmentów zachowania modeli w niej rozważanych (c-TdPN). W szczególności przedstawia ona rozwiązanie problemu dla takich modeli opisywanych w logikach temporalnych czasu rzeczywistego.

Problem którego dotyczy rozprawa polega po pierwsze na znalezieniu tych własności modeli c-TdPN, przy których istnieje skończony odcinek początkowy rozwinięcia modelu i dotyczące tego odcinka ograniczenia momentów wykonania tranzycji takie, które wyznaczają wszystkie rozgałęzione procesy czasowe modelowanego systemu.

Po drugie na opracowaniu algorytmu rozstrzygania czy dany model systemu ma takie własności i wyznaczania odpowiedniego odcinka początkowego i dotyczących go ograniczeń, jeżeli takie istnieją.

W celu rozwiązania problemu postawionego w rozprawie wybrano charakterystyczne dla podobnych modeli sposoby opisu ich zachowania przez rozgałęziony proces czasowy i przez czasowy system tranzycyjny. Dla obu takich modeli, reprezentujących semantykę innego czasowego rozszerzenia sieci Petriego przedstawiono znane i powszechnie stosowane sposoby wyznaczania reprezentatywnego fragmentu.

Dla czasowego systemu tranzycyjnego zbadano metody weryfikacji jego własności opisanych w logikach modalnych czasu rzeczywistego (TCTL). Na tej podstawie opracowano sposób wyznaczania reprezentatywnego fragmentu dla sieci c-TdPN dla badanych opisów zachowania oraz algorytmy weryfikacji jego własności reprezentowanych przez formuły logiki TCTL. Te prace miały na celu wykazanie, że przeprowadzone w rozprawie rozwiązanie jest poprawne i użyteczne w zastosowaniach.

Abstract

This dissertation concerns the methods of description and analysis of the operation of concurrent systems. The selected model of the system is the contextual timed Petri net. In this model, the moments of appearing and removing tokens are observed in places from which arcs lead to the transition. The model specification is based on the assumption that tokens are only available in arcs of time intervals. For systems represented by such models, their behaviours were defined and the properties of these behaviours were examined.

The dissertation complements the search for representative fragments of behaviour of the models considered in it (c-TdPN). In particular, it presents a solution to the problem for such models described in real-time temporal logics.

The problem which the dissertation relates to is first of all finding those properties of the c-TdPN models, where there is a finite initial segment of the model development and the limit of the moment of execution of the transitions that determine all branching time processes of the modelled system.

Secondly, the development of an algorithm for determining whether a given system model has such properties and determining the appropriate initial section and restrictions that apply to it, if any.

In order to solve the problem set in the thesis, characteristic for similar models of ways of describing their behavior by a branched time process and by a temporary transitional system were chosen. For both such models representing the semantics of another temporal extension of the Petri network, known and widely used methods for determining a representative fragment were presented.

For the temporary transit system, the methods of verification of its properties described in real-time modal logic (TCTL) were investigated. On this basis, a method for determining a representative fragment for the c-TdPN network for the studied behavioral descriptions and algorithms for verification of its properties represented by the TCTL logic formulas were developed. This work was aimed at demonstrating that the solution carried out at the hearing was correct and useful in applications.

Wstęp

Niniejsza rozprawa dotyczy metod opisu i analizy działania systemów współbieżnych. Wybranym modelem systemu jest czasowo-kontekstowa sieć Petriego. W tym modelu obserwowane są momenty pojawienia się i usuwania żetonów w miejscach, z których prowadzą łuki do tranzycji. Specyfikacja modelu polega na założeniu, że żetony są dostępne tylko w przyporządkowanych łukom przedziałach czasu. Dla systemów reprezentowanych przez takie modele zdefiniowano ich zachowania i zbadano własności tych zachowań.

Autor nie znalazł w literaturze satysfakcjonującego rozwiązania problemu wyznaczania reprezentatywnego fragmentu zachowania modelu systemu pozwalającego na weryfikację własności z zależnościami czasowymi. Rozprawa uzupełnia zatem poszukiwania reprezentatywnych fragmentów zachowania modeli w niej rozważanych (c-TdPN). W szczególności przedstawia ona rozwiązanie problemu dla takich modeli opisywanych w logikach temporalnych czasu rzeczywistego.

W celu rozwiązania problemu autor wybrał charakterystyczne dla podobnych modeli sposoby opisu ich zachowania przez rozgałęziony proces czasowy i przez czasowy system tranzycyjny. Dla obu takich modeli, reprezentujących semantykę innego czasowego rozszerzenia sieci Petriego (Merlin [35, 36]) przedstawiono znane i powszechnie stosowane sposoby wyznaczania reprezentatywnego fragmentu. Głównymi pracami w których poruszono te zagadnienia są [32, 46] o tzw. symbolicznym procesie czasowym oraz [9] o czasowym systemie tranzycyjnym po dyskretyzacji przy pomocy stref.

Dla czasowego systemu tranzycyjnego zbadano metody weryfikacji jego własności opisanych w logikach modalnych czasu rzeczywistego (TCTL). Jest to rozwiązanie opisane przede wszystkim w pracy [9]. Na tej podstawie autor opracował sposób wyznaczania reprezentatywnego fragmentu dla sieci c-TdPN dla badanych opisów zachowania oraz algorytmy weryfikacji jego własności reprezentowanych przez formuły logiki TCTL. Te prace miały na celu wykazanie, że przeprowadzone w rozprawie rozwiązanie jest poprawne i użyteczne w zastosowaniach.

Niniejsza rozprawa pozwala także na uzyskanie odpowiedzi na szereg innych pytań:

- jakie opisy zachowania należy rozważać dla modeli c-TdPN?
- jakie założenia należy przyjąć, aby badać interesujące własności?
- jakie własności można badać dla modelu c-TdPN?
- jaki jest koszt algorytmów je weryfikujących?
- jaka jest jakość proponowanego rozwiązania w porównaniu do znanego narzędzia TAPAAL [13, 29] ?

Rozprawa składa się z sześciu rozdziałów, wstępu i zakończenia. W pierwszym rozdziale przedstawiono opis modelu rozważanego w rozprawie, problem który ma być rozwiązany, tezę i pytania badawcze, oraz związek wybranego modelu z innymi znanymi modelami. W drugim rozdziale opisano znane rozwiązanie problemu wyznaczania

reprezentatywnego fragmentu zachowania dla innego czasowego modelu sieci Petriego. W rozdziale trzecim przedstawiono znaną metodę dyskretyzacji przestrzeni stanów zachowania i scharakterysowano znane sposoby weryfikacji dla rozszerzenia logiki czasu rozgałęziającego się. W rozdziale czwartym zawarto opis proponowanego rozwiązania zawierający potrzebne dodatkowe definicje i algorytmy. W rozdziale piątym zbadano złożoność czasową proponowanych algorytmów. W rozdziale szóstym zaprezentowano zastosowania proponowanych metod do badania modelu konkretnego systemu. Zastosowania dla innych modeli systemów dodano w załącznikach.

W zakończeniu autor podsumował przeprowadzone badania i wskazał kierunki dalszych prac.

1 O pewnych systemach obiektów i ich działaniu

1.1 Modele systemów, ich zachowania i związana z tym problematyka

Rozprawa dotyczy metod opisu i analizy działania systemów współbieżnych. Rozważanym modelem systemu jest struktura zwana modelem c-TdPN (akr. od contextual Timed Petri Net).

Definicja 1 Modelem c-TdPN *jest struktura* $N = (P, T, F, C, I, m)$*, gdzie*

- P *jest skończonym zbiorem elementów zwanych* miejscami*,*
- T *jest skończonym i rozłącznym z* P *zbiorem elementów zwanych* tranzycjami*,*
- $F \subseteq (P \times T) \cup (T \times P)$ *jest zbiorem par reprezentujących* łuki *łączące miejsca z tranzycjami i tranzycje z miejscami,*
- $C \subseteq P \times T$ *jest rozłącznym z* F *zbiorem par reprezentujących łuki zwane* łukami czytania *łączące miejsca z tranzycjami,*
- I *jest funkcją przyporządkowującą każdemu łukowi* $(p, t) \in F$ *przedział* $[0, \infty]$ *osi liczbowej z dodaną wartością* ∞ *zwany* przedziałem dostępności*, a każdemu łukowi czytania przedział dostępności* $[0, \infty]$*,*
- m *jest funkcją, zwaną* stanem początkowym*, przyporządkowującą każdemu miejscu pewien skończony zbiór nierozróżnialnych elementów zwanych* żetonami. □

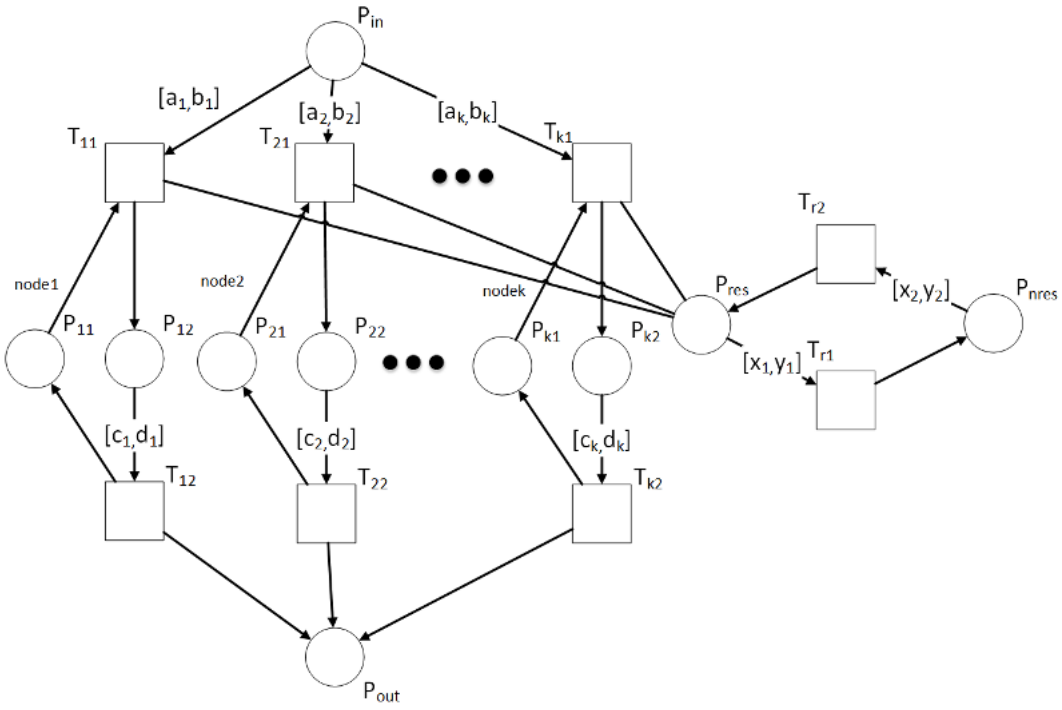
Funkcje takie jak m, albo inaczej wielozbiory miejsc, reprezentują możliwe *stany* systemu, przy czym samo m występujące w modelu reprezentuje *stan początkowy*. Tranzycje reprezentują *akcje* jakie system może *wykonać*, jeśli z każdego miejsca skąd prowadzą do nich łuki można wybrać przynajmniej po jednym żetonie i jeśli czas jaki upłynął od momentu pojawienia się takiego żetonu mieści się w przyporządkowanym łukowi przedziale dostępności. Wykonanie takiej akcji powoduje zużycie wybranych żetonów i utworzenie żetonu w każdym miejscu dokąd prowadzi łuk wychodzący z tranzycji, która tę akcję reprezentuje. Rozpoczynające się od stanu początkowego wykonywanie kolejno umożliwianych tranzycji tworzy graf wykonywanych tranzycji i osiąganych stanów.

Czas jaki upłynął od momentu umieszczenia żetonu w miejscu nazywany jest *wiekiem żetonu*.

Ograniczenie struktury N do (P, T, F) (odpowiednio do (P, T, F, C)) z opisaną interpretacją jest znaną w teorii systemów *siecią Petriego* (odpowiednio *kontekstową siecią Petriego*), a ograniczenie do (P, T, F, m) (odpowiednio do (P, T, F, C, m)) *systemem*

sieciowym (odpowiednio *kontekstowym systemem sieciowym*) ze stanem początkowym m. Formalnie jest to graf z węzłami dwóch rodzajów (miejsca i tranzycje) i z łukami łączącymi węzły różnych rodzajów.

Przykład 1 *Systemem współbieżnym jest opisana w [30] oraz zastosowana w [18, 19] Platforma Modelowania i Symulacji LabTSI(R). Platforma ta służy do zrównoleglania przy pomocy PVM (ang. Parallel Virtual Machine) [24] złożonych obliczeń na wielu heterogenicznych komputerach, z których każdy może wykonać każde z zadań jeśli nie jest zajęty i jeśli istnieją odpowiednie zasoby. Modelem tej platformy i zestawu zadań do wykonania jest struktura przedstawiona na rysunku 1, gdzie koła oznaczają miejsca a prostokąty tranzycje, tak jak to jest przyjęte dla sieci Petriego i pokrewnych modeli.*



Rysunek 1: Model c-TdPN dla systemu obliczeń rozproszonych.

Żetony w miejscu P_{in} oznaczają zadania do wykonania. Żetony w miejscu P_{out} oznaczją zadania wykonane. Żeton w miejscu P_{res} oznacza obecność potrzebnych zasobów. Żeton w miejscu P_{i1} oznacza gotowość i-tego komputera ($1 \leqslant i \leqslant k$) do wykonania zadania a żeton w miejscu P_{i2} brak takiej gotowości. Tranzycja T_{i1} oznacza rozpoczęcie wykonania zadania przez i-ty komputer a tranzycja T_{i2} zakończenie wykonania zadania.

Łuki (P_{in}, T_{i1}), (P_{i1}, T_{i1}), (P_{res}, T_{i1}) oraz odpowiadające im przedziały dostępności wskazują jakie żetony są potrzebne do wykonania tranzycji T_{i1} i w jakich momentach

czasu licząc od ich powstania mają być dostępne (brak takich przedziałów na rysunku oznacza przedział $[0, \infty)$). Analogicznie łuk (P_{i2}, T_{i2}) wskazuje jaki żeton jest potrzebny do wykonania T_{i2}.□

Działanie systemu jest reprezentowane przez tak zwany rozgałęziony proces czasowy systemu (w skr. RPC).

Definicja 2 Rozgałęzionym procesem czasowym *systemu, którego modelem jest struktura $N = (P, T, F, C, I, m)$, jest para składająca się ze struktury $U(N) = (B, E, G, H, l)$ zwanej* rozwinięciem *modelu N i z* ograniczeń *jakie musi spełniać funkcja θ przyporządkowująca każdemu wykonaniu $e \in E$ tranzycji $l(e) \in T$* moment wykonania *$\theta(e) \in [0, \infty]$, gdzie*

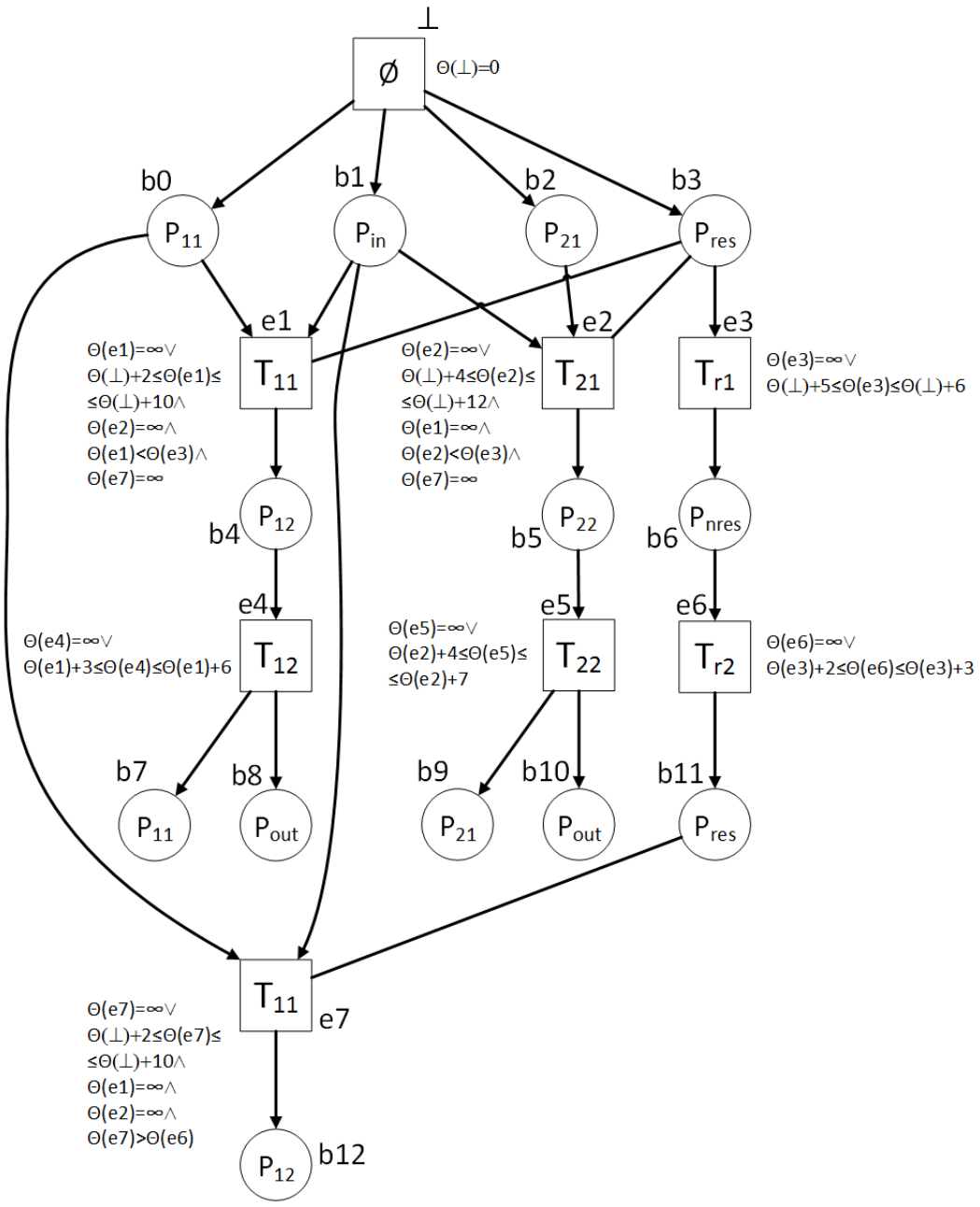
- *(B, E, G, H) jest acykliczną kontekstową siecią Petriego z łukami $g \in G = (B \times E) \cup (E \times B)$ i łukami czytania $h \in H = (B \times E)$, nie zawierającą miejsc kończących więcej niż jeden luk, zwaną* siecią rozwinięcia sieci systemu,

- *l jest funkcją przyporządkowującą miejscom sieci rozwinięcia miejsca sieci systemu i tranzycjom sieci rozwinięcia tranzycje sieci systemu, gdzie każde miejsce $b \in B$ sieci rozwinięcia reprezentuje potencjalnie możliwy konkretny żeton w miejscu $l(b) \in P$ sieci systemu, każda tranzycja $e \in E$ sieci rozwinięcia reprezentuje potencjalnie możliwe konkretne wykonanie tranzycji $l(e) \in T$ sieci systemu, każdemu łukowi $g \in G$ odpowiada łuk $l(g) \in F$ sieci systemu i każdemu łukowi $h \in H$ odpowiada łuk $l(h) \in C$ sieci systemu,*

- *funkcja θ przyporządkowuje każdemu zdarzeniu polegającemu na wykonaniu $e \in E$ tranzycji $l(e) \in T$ moment wykonania $\theta(e) \in [0, \infty]$ z dostępnymi potrzebnymi do tego żetonami, oraz w szczególności $\theta(e) = \infty$ gdy brak takiego momentu.* □

Zdarzenie e dla którego przypisano moment $\theta(e) = \infty$ jest nazywane *zdarzeniem niemożliwym.*

Przykład 2 *Początkowy fragment działania systemu zadań i dwóch komputerów heterogenicznych jak w przykładzie 1, gdzie $k = 2$, $[a_1, b_1] = [2, 10]$, $[c_1, d_1] = [3, 6]$, $[a_2, b_2] = [4, 12]$, $[c_2, d_2] = [4, 7]$, $[x_1, y_1] = [5, 6]$, $[x_2, y_2] = [2, 3]$, $m(P) = \{0\}$ dla $P \in \{P_{in}, P_{11}, P_{12}, P_{res}\}$ oraz $m(P) = \emptyset$ dla pozostałych, jest przedstawiony na rysunku 2.*



Rysunek 2: Rozgałęziony proces czasowy dla systemu zadań i dwóch komputerów.

Każde miejsce b sieci rozwinięcia reprezentuje konkretny żeton w miejscu $l(b)$ sieci systemu, a każda tranzycja e sieci rozwinięcia reprezentuje konkretne zdarzenie polegające na wykonaniu tranzycji $l(e)$ sieci systemu. Każdemu zdarzeniu e odpowiada formuła logiczna ograniczająca moment jego wykonania. Orzeka ona że, po pierwsze,

moment ten jest najwcześniejszym momentem w którym wszystkie potrzebne do wykonania żetony są w swoich przedziałach dostępności, i po drugie, że moment ten mieści się w przedziałach dostępności wszystkich potrzebnych do wykonania żetonów. Brak wykonania tranzycji jest reprezentowany przez przypisanie jej momentu wykonania ∞.

Przykładowo, przy założeniu, że system otrzymuje żetony stanu początkowego i zaczyna działać w chwili $\theta(\perp) = 0$ *oraz zadanie wykonano na pierwszym węźle, ograniczenia na moment wykonania zdarzenia e1 wyrażają się formułą* $\theta(\perp) + 2 \leqslant \theta(e1) \leqslant \theta(\perp) + 10$*, ograniczenia na moment wykonania zdarzenia e4 wyrażają się formułą* $\theta(e1) + 3 \leqslant \theta(e4) \leqslant \theta(e1) + 6$*, itp. - jak na rysunku 2. Ponadto w kontekście całego zachowania* $\theta(e2) = \infty$ *i* $\theta(e1) < \theta(e3)$. $\square$

Problem którego dotyczy rozprawa polega na:

- znalezieniu tych własności modeli c-TdPN, przy których istnieje skończony odcinek początkowy rozwinięcia modelu i dotyczące tego odcinka ograniczenia momentów wykonania tranzycji takie, które wyznaczają wszystkie rozgałęzione procesy czasowe modelowanego systemu,
- opracowaniu algorytmu rozstrzygania czy dany model systemu ma takie własności i wyznaczania odpowiedniego odcinka początkowego i dotyczących go ograniczeń, jeśli takie istnieją.

Rozwiązanie opisane w rozprawie polega na rozwinięciu i dostosowaniu sposobu konstruowania prefiksu zachowania systemu czasowej sieci Petriego ze stoperami i zmiennymi zwanymi parametrami. Ponadto polega na wyznaczaniu reprezentatywnego fragmentu przy założeniu wykluczenia parametrów i łuków stoperów, do reprezentacji własności zachowania modeli c-TdPN.

W literaturze nie znaleziono prac opisujących konstrukcję reprezentatywnego fragmentu zachowania modeli c-TdPN pozwalającego na orzekanie o stanach systemu pomiędzy założonymi momentami czasu. Może to wynikać z faktu, że poszukiwano zbyt uniwersalnych rozwiązań problemów takich jak osiągalność stanów (ang. reachability), pokrycie stanów osiągalnych (ang. coverability) oraz ograniczoność liczby żetonów w miejscach (ang. boundedness). Problemy te są nierozstrzygalne. Na przykład w pracach [42, 15] udowodniono, że problem osiągalności stanów w przypadku ogólnym w modelu c-TdPN bez miejsc odczytu jest nierozstrzygalny. Zaprezentowano dowód na znanym przykładzie maszyny Minsky'ego z dwoma licznikami [37]. Ponadto w pracy [13] opisano narzędzie TAPAAL weryfikowania własności opisanych w logice czasu rozgałęziającego się (CTL).

Problem osiągalności stanów staje się rozstrzygalny, nawet przy obecności miejsc odczytu [44, 11, 43, 10] jeżeli model c-TdPN bez łuków czytania zostanie zawężony do klasy sieci 1-bezpiecznych (1-safe, w rozumieniu tw. 4.10 [50]).

Rozważania należy zawęzić do klasy sieci 1-bezpiecznych w przypadku, gdy chcemy zaproponować racjonalny algorytm konstruowania reprezentatywnego zachowania dla modelu c-TdPN w terminologii RPC. Aczkolwiek stosując semantykę i dyskretyzację

przestrzeni stanów, jak w [9], możliwe jest orzekanie o stanach systemu i o sekwencjach tranzycji ("śladach") pomiędzy zadanymi momentami czasu w sposób zautomatyzowany.

Zasadnym jest więc podejmowanie problemu zdefiniowania rozwinięcia oraz fragmentu reprezentatywnego zachowania systemu dla tegoż modelu.

Przedstawione w pracy rozwiązanie postawionego problemu jest rozwinięciem i zostosowaniem pomysłu standardowej identyfikacji zdarzenia powtórnego wykonania tranzycji w specjalnie zdefiniowanych równoważnych stanach systemu [32] oraz zastosowania dyskretyzacji przez konstruowanie grafu stref ZBG [23].

1.2 Związek wybranego modelu z innymi znanymi modelami

W niniejszym podrozdziale zostanie przedstawiony związek modelu c-TdPN z dwoma znanymi w literaturze odmiennymi czasowymi sieciami Petriego. Są to sieci czasowe, w których tranzycja może być uruchomiona po pewnym czasie od pojawienia się żetonów w miejscach, z których wychodzą do niej łuki. Tranzycja ta produkuje natychmiast nowe żetony w miejscach na które wskazuje łukiem.

Model c-TdPN, gdy zostanie pozbawiony łuków czytania jest czasową siecią Petriego z ograniczeniami czasowymi dla łuków oraz wiekiem dla żetonów (akr. TdPN, od *Timed Petri Net*, lub TAPN od *Timed Arc Petri Net*), rozważaną w pracy Waltera [47] i innych [27, 1].

Gdy z modelu c-TdPN zostaną usunięte przedziały dostępności dla łuków, wiek dla żetonów, natomiast zostaną dodane przedziały czasowe opóźniające wykonanie tranzycji i miejsca umożliwiające wstrzymanie i wznowienie wykonywania tranzycji, to powstały model jest przedziałową siecią Petriego z obecnością stoperów, rozważaną w pracy Berthomieu [7], z dodatkowymi parametrami [32] oraz bez stoperów i parametrów w pracach Merlina [35, 36] jako czasowa sieć Petriego (akr. TPN, od *Time Petri Net*).

Rozgałęziony proces czasowy może reprezentować zachowanie dla obu wcześniej wspomnianych modeli. Innym modelem reprezentacji zachowania może być czasowy system tranzycyjny, który jest uboższym modelem, aczkolwiek jest przydatny w automatycznej weryfikacji własności modelowanego systemu.

1.2.1 Sieci z ograniczeniami czasowymi dla łuków i żetonów

Model $N = (P, T, F, C, I, m)$ typu c-TdPN, który nie zawiera łuków czytania $C = \emptyset$ jest znanym z literatury modelem typu TdPN i może być reprezentowany jako struktura $N = (P, T, F, I, m)$ z jej normalną interpretacją. Przy pomocy modelu TdPN nie można modelować konieczności uruchomienia tranzycji, która reprezentuje konieczność wykonania akcji systemu. Konieczność wykonania akcji w systemie blokuje postęp czasu.

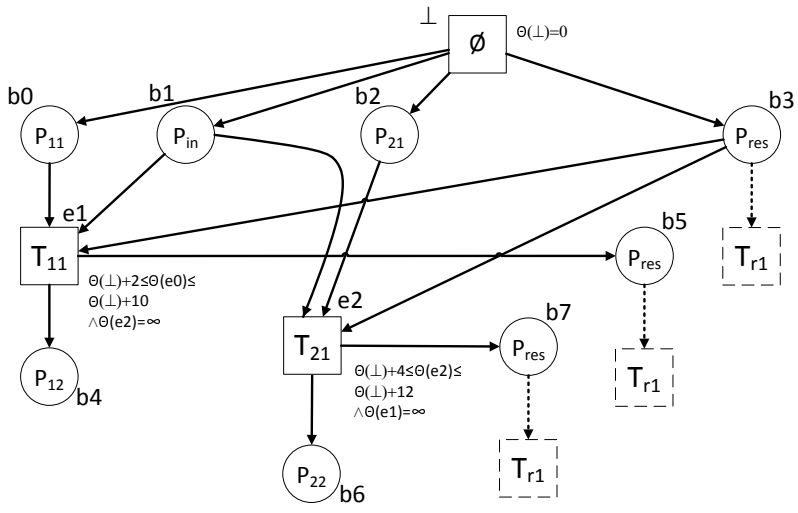
Znane inne rozszerzenia modelu pozwalające rozważać podobne interpretacje to m. in. stosowanie miejsc inhibicji uniemożliwiających pewne tranzycje, nakładanie ograniczeń zwanych inwariantami na wiek żetonów, ustalanie konieczności wykonywania tranzycji po określonym czasie jaki upływa od jej umożliwienia.

Model TdPN umożliwia obserwowanie upływu czasu bezpośrednio dla obiektów występujących w systemie, reprezentowanych przez obecność żetonów w wybranych miejsach.

Przykładowo, aby reprezentować system opisany w pierwszym rozdziale przy pomocy modelu TdPN należałoby zamienić każdy łuk czytania (T_{i1}, P_{res}) na parę łuków skierowanych (P_{res}, T_{i1}) i (T_{i1}, P_{res}), jak w znanym prostym modelowaniu (ang. plain encoding)[41, 3] sieci kontekstowych przy pomocy sieci zwykłych. Spowoduje to jednak brak możliwości wykonania par tranzycji T_{i1} i T_{j1}, dla $i \neq j$, w tej samej chwili. Istnieją inne metody modelowania łuków czytania (miejsc odczytu), nie mniej jednak powodują niepożądany rozrost sieci lub rozrost jej modelu zachowania. Fakt ten uzasadnia zastosowanie łuków czytania w modelu c-TdPN. Łuki czytania można modelować także opisanymi w literaturze łukami przenoszącymi wiek żetonu w nowe miejsce (tzw. łuki transportu).

Zachowanie TdPN może być także reprezentowanie przy pomocy rozgałęzionego procesu czasowego (def. 2), który nie zawiera łuków czytania $C = \emptyset$.

Przykładowo rysunek 3 przedstawia rozgałęziony proces czasowy zawierający uruchomienie dwóch tranzycji T_{11} i T_{21} systemu sieciowego z przykładu 2, w którym łuki czytania zamieniono na pętle.



Rysunek 3: Rozgałęziony proces czasowy modelu TdPN. Przerywana linia oznacza możliwe zdarzenie.

W porównaniu do rozgałęzionego procesu czasowego (rys. 2) uruchomienie tranzycji T_{r1} musi być reprezentowane przez trzykrotnie większą liczbę zdarzeń.

Dla modelu typu TdPN nie znaleziono w literaturze opisanego sposobu konstruowania reprezentatywnego fragmentu zachowania przy pomocy rozgałęzionego procesu czasowego.

1.2.2 Sieci z ograniczeniami czasowymi do przejść ze stoperami

Model c-TdPN, w którym rezygnuje się z własności:

- odmierzania czasu wystąpienia żetonu w miejscu,
- przedziałów dostępności na łukach,

ale dodaje się własności:

- przedziałów czasowych opóźnień ze zmiennymi zwanymi parametrami dla wykonań tranzycji,
- miejscami umożliwiającymi wstrzymanie i wznowienie tych wykonań,

nazywamy modelem przedziałowej sieci Petriego z obecnością stoperów i parametrów (akr. PSwPN od *Parametric Stopwatches Petri Net*).

Definicja 3 *Modelem PSwPN jest struktura* $M = (P, T, F, C, S, I, J, V, \widehat{V}, m)$ *gdzie:*

- *struktura* (P, T, F, C, m) *jest kontekstowym systemem sieciowym ze stanem początkowym* m*,*
- $S \subseteq P \times T$ *jest rozłącznym z* $F \cup C$ *zbiorem par reprezentujących łuki łączące miejsca z tranzycjami, zwane* łukami stoperów
- V *jest skończonym zbiórem parametrów,*
- $\widehat{V}$ *jest zbiorem wartościowań parametrów ze zbioru* V*, takim, że dla każdej tranzycji* $t \in T$ *i każdego wartościowania* $v \in \widehat{V}$ *zachodzi* $0 \leqslant I(t) \leqslant J(t)$*,*
- I *jest zależną od parametrów* V *funkcją przyporządkowującą każdej tranyzcji* $t \in T$ *statyczny minimalny moment opóźnienia wykonania,*
- J *jest zależną od parametrów* V *funkcją przyporządkowującą każdej tranyzcji* $t \in T$ *statyczny maksymalny moment opóźnienia wykonania, który jest zwany pilnym, gdy* $J(t) \neq \infty$*,*
- m *jest funkcją (stanem początkowym), przyporządkowującą każdemu miejscu nieujemną liczbę żetonów.* □

Podobnie jak dla modelu typu c-TdPN funkcje takie jak m reprezentują możliwe *stany* systemu, a samo m występujące w modelu reprezentuje *stan początkowy*.

Tranzycja $t \in T$ jest *czynna* jeżeli, w każdym miejscu z którego wychodzą do niej zwykłe łuki bądź łuki czytania znajduje się przynajmniej jeden żeton. W przeciwnym razie tranzycję nazwiemy *nieczynną*. Czynność tranzycji ma pewne stany. Jeżeli tranzycja stała się czynną to powiemy, że została *zainicjowana*. Tranzycja czynna, która ma dodatkowo żeton w miejscu stopera, z którego wychodzi do niej łuk, jest *postępująca*. Jeżeli żeton ten zostanie zabrany to tranzycja jest *zawieszona*. Tranzycja może być *wznowiona* po powrocie żetonu. Jeżeli czas postępu tranzycji jest co najmniej równy minimalnemu statycznemu opóźnieniu $I(t)$ to tranzycja jest *umożliwiona* i może być *wykonana*. Postęp tranzycji jest możliwy maksymalnie do momentu osiągnięcia maksymalnego statycznego opóźnienia $J(t)$. Gdy $J(t) = \infty$, to tranzycja nie ma tego ograniczenia. Wykonanie tranzycji finalizuje postęp reprezentowanego procesu.

Tranzycje reprezentują *akcje* jakie system może *wykonać*, jeśli z każdego miejsca skąd prowadzą do nich łuki można wybrać przynajmniej po jednym żetonie i jeśli czas postępu tranzycji mieści się pomiędzy jej minimalnym, a maksymalnym statycznym opóźnieniem wykonania.

Szczególnym przypadkiem akcji systemu, unikalnym dla modelu typu PSwPN, jest akcja zawierająca przerwanie. Taka akcja jest także finalizowana wykonaniem tranzycji. W takim przypadku wykonanie tranzycji ma wydłużone minimalne i maksymalne opóźnienie. Minimalny moment wykonania jest opóźniony o najkrótsze możliwe przerwanie, a maksymalny o najdłuższe możliwe przerwanie.

Wykonanie akcji powoduje zużycie wybranych żetonów i utworzenie żetonu w każdym miejscu dokąd prowadzi łuk wychodzący z tranzycji, która tę akcję reprezentuje. Rozpoczynające się od stanu początkowego wykonywanie kolejno umożliwianych tranzycji tworzy graf wykonywanych tranzycji i osiąganych stanów.

Funkcja *wartościująca* $v : V \to [0, \infty)$ nadaje wartości parametrom z V. $v(M) = (P, T, F, C, S, v(I), v(J), m_0)$ oznacza czasową sieć Petriego ze stoperami [7] rozszerzoną o miejsca odczytu.

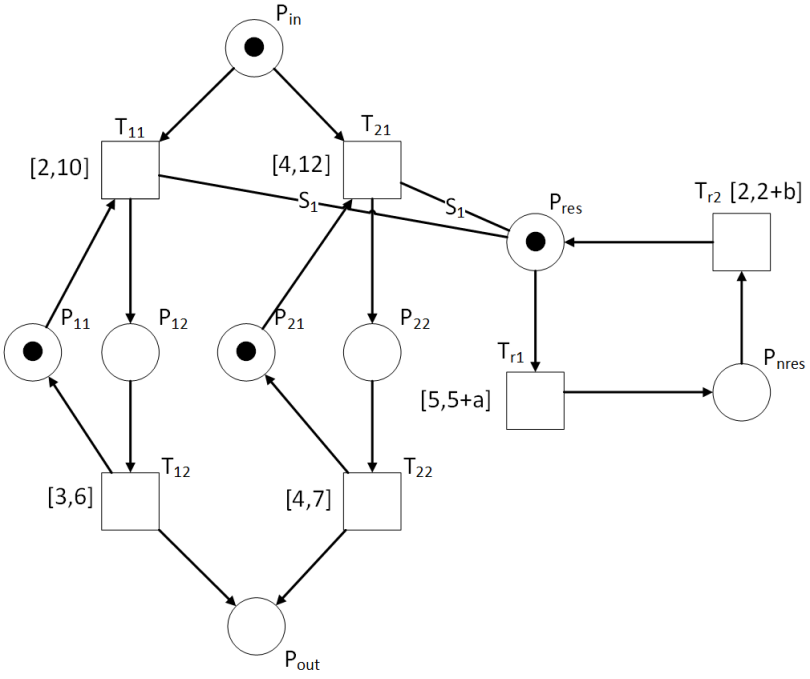
Ograniczenie struktury M typu PSwPN do struktury $(P, T, F, v(I), v(J), m)$ jest znanym systemem sieciowym czasowego modelu rozważanego przez Merlina [35]. W takim ograniczeniu postęp tranzycji nie może być zawieszony.

Przykład 3 *Przykładowo model dwuwęzłowego systemu obliczeń rozproszonych może być także reprezentowany przy pomocy PSwPN jako graf na rysunku 4. Przedziały są przyporządkowane do tranzycji i informują o jej minimalnym i maksymalnym opóźnieniu wykonania, a żetony nie mają określonego wieku. Konkretna struktura takiego systemu zainicjowana do obliczenia zadania na wybranym węźle to*

$$M_1 = (P_1, T_1, F_1, C_1, S_1, I_1, J_1, V_1, \widehat{V}_1, m_1)$$

o miejscach $P_1 = \{P_{in}, P_{out}, P_{j1}, P_{j2}, P_{res}, P_{nres}\}$ *oraz tranzycjach* $T_1 = \{T_{j1}, T_{j2}, T_{r1}, T_{r2}\}$, *dla* $j = 1, 2$. *Tym razem obliczanie zadania jest reprezentowane przez czynność tranzycji* T_{j2}, *a jego ukończenie przez jej wykonanie.* F_1 *jest zdefiniowana standardowo na grafie*

jako łuki z grotami, a S_1 jako łuki bez grotów z etykietą, $C_1 = \emptyset$ nie występuje w przykładzie. Akcja odcięcia zasobów do czytania powoduje zawieszenie tranzycji T_{j1} reprezentującej pobieranie zadania obliczeniowego. Ograniczenia $v(I)$ i $v(J)$ tranzycji $t \in T_1$ są także reprezentowane na grafie jako domknięte podzbiory $[I(t), J(t)]$ półosi $[0, \infty)$. W sieci są obecne dwa parametry $V_1 = \{a, b\}$ z ograniczeniem $\widehat{V}_1 = \{0 \leqslant a \leqslant 5, 0 \leqslant b \leqslant 5\}$. Początkowy stan systemu określa funkcja m_1: $m_1(P) = 1$ dla $P \in \{P_{in}, P_{11}, P_{21}, P_{res}\}$ oraz $m_1(P) = 0$ dla pozostałych. Żeton w miejscu P_{in} reprezentuje zadanie do obliczenia, w miejscach P_{j1} wolne wątki, a w miejscu P_{res} dostępne zasoby.□



Rysunek 4: Model systemu zadań i dwóch węzłów M_1 typu PSwPN.□

Działanie tego typu systemów jest reprezentowane przez tak zwany symboliczny rozgałęziony proces czasowy systemu (w skr. SRPC).

Definicja 4 Symboliczny rozgałęziony proces czasowy *systemu, którego modelem jest struktura $M = (P, T, F, C, S, I, J, V, \widehat{V}, m)$ jest trójką składającą się ze struktury $\beta = (B, E, G, H, l)$ zwanej rozwinięciem modelu M, z ograniczeń jakie musi spełniać funkcja θ przyporządkowującą każdemu wykonaniu $e \in E$ tranzycji $l(e) \in T$* moment wykonania *$\theta(e) \in [0, \infty]$, oraz z funkcji v wartościującej zbiór parametrów V, gdzie:*

- *(B, E, G, H) jest acykliczną kontekstową siecią Petriego z łukami $g \in G = (B \times E) \cup (E \times B)$ i łukami czytania $h \in H = (B \times E)$, nie zawierającą miejsc kończących więcej niż jeden luk, zwaną* siecią rozwinięcia sieci systemu,

- *l jest funkcją przyporządkowującą miejscom sieci rozwinięcia miejsca sieci systemu i tranzycjom sieci rozwinięcia tranzycje sieci systemu, gdzie każde miejsce $b \in B$ sieci rozwinięcia reprezentuje potencjalnie możliwy konkretny żeton w miejscu $l(b) \in P$ sieci systemu, każda tranzycja $e \in E$ sieci rozwinięcia reprezentuje potencjalnie możliwe konkretne wykonanie tranzycji $l(e) \in T$ sieci systemu, każdemu łukowi $g \in G$ odpowiada łuk $l(g) \in F$ sieci systemu i każdemu łukowi $h \in H$ odpowiada łuk $l(h) \in C \cup S$ sieci systemu,*

- *Struktura β jest acyklicznym grafem, którego łuki reprezentują porządek określony przy pomocy relacji $G \cup H$.*

- *$\theta : E \to [0, \infty]$ jest funkcją przypisującą każdemu zdarzeniu e ze zbioru E, jego moment wykonania $\theta(e) \in [0, \infty]$.*

- *v jest wartościowaniem zbioru parametrów V występujących w M.* $\square$

Każde zdarzenie $e \in E$ wykonania tranzycji $l(e) \in T$ jest reprezentowane przez trójkę $e = (t, A, A')$, gdzie $t = l(e)$, $A = \{b \in B : l(b) \in Ft\}$ jest zbiorem miejsc zawierających zużyty żeton przez t, a $A' = \{b \in B : l(b) \in Ct \cup St\}$ jest zbiorem miejsc odczytu.

Symbol $\Gamma(M)$ oznacza symboliczny rozgałęziony proces czasowy modelu systemu M.

Przykład 4 *Przykładowo rysunek 5 przedstawia początkowy fragment zachowania systemu M_1, reprezentowany przez $\Gamma(M_1)$. Funkcja v_1 określa wartościowanie parametrów: $v_1(a) = v_1(b) = 1$. Model systemu jest zainicjowany żetonami w miejscach $\{P_{11}, P_{in}, P_{21}, P_{res}\}$ w momencie $\theta(\perp) = 0$. Łuki czytania reprezentują łuki stoperów.*

Fragment początkowy jest na tyle obszerny, że zawiera dwa interesujące ciągi wykonań. Pierwszy jest charakterystyczny dla modeli bez stoperów (Merlin), oraz drugi charakterystyczny dla PSwPN.

Pierwszy ciąg kolejnych wykonań $\{\perp, e2, e7\}$ reprezentuje złożoną akcję przeprowadzenia obliczeń w jednym węźle. Ciąg wykonań opatrzony jest ograniczeniami:

- $\theta(\perp) + 2 \leqslant \theta(e2) \leqslant \theta(\perp) + 10$,

- $\theta(e2) \leqslant \theta(e0)$,

- $\theta(e3) = \theta(e4) = \theta(e6) = \infty$,

- $\theta(e2) + 3 \leqslant \theta(e7) \leqslant \theta(e2) + 6$.

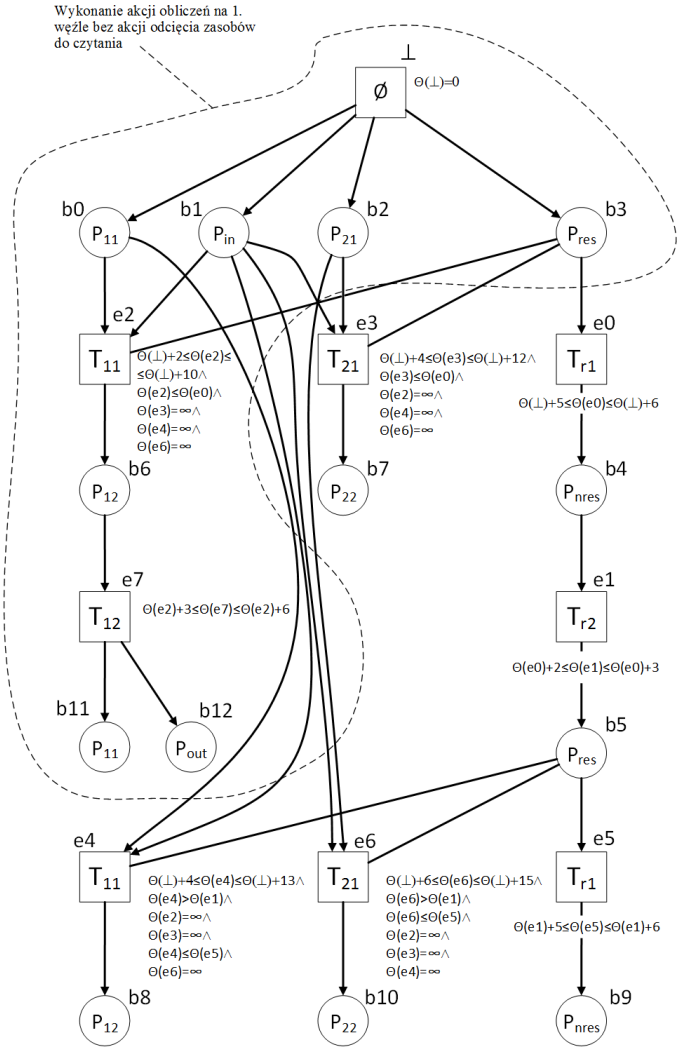
Ponieważ zdarzenia $e3$,$e4$ i $e6$ przeszkadzają w wykonaniu $\{\perp, e2, e7\}$ to zostały one opatrzone momentami nieskończonymi symbolizującymi brak wykonania.

Drugim interesującym ciągiem kolejnych wykonań jest $\{\perp, e0, e1, e6\}$, w którym tranzycja T_{21} była w pewnym momencie zawieszona. Ciąg ten reprezentuje pobranie zadania do węzła 2 po uprzednim odcięciu zasobów. Ciąg ten jest unikalny dla PSwPN.

Tranzycja T_{21} po wznowieniu została uruchomiona w zdarzeniu $e6$. Stąd minimalne ograniczenie opóźnienia uruchomienia T_{21} zostało dodatkowo opóźnione o 2 jednostki. W rezultacie stanowi moment $\theta(\perp) + 6$. Te 2 jednostki to minimalne opóźnienie uruchomienia T_{r2}.

Podobnie maksymalne ograniczenie opóźnienia uruchomienia T_{21} zostało dodatkowo opóźnione o 3 jednostki. W rezultacie stanowi moment $\theta(\perp) + 15$. Te 3 jednostki to maksymalne opóźnienie uruchomienia T_{r2}. □

Modele takie jak rozgałęziony proces czasowy, symboliczny rozgałęziony proces czasowy pozwalają zapanować nad rozrostem przestrzeni stanów poprzez łączenie nakładających się kolejnych wykonań tranzycji systemu.



Rysunek 5: Przykładowy $\Gamma(M_1)$ dla systemu zadań i dwóch węzłów M_1.

1.2.3 Czasowy system tranzycyjny

Innym modelem reprezentującym zachowanie wcześniej opisanych systemów czasowych, jest czasowy system tranzycyjny (w skr. TTS). W porównaniu do rozgałęzionego procesu czasowego, jest nieco uboższym w informacje modelem. Aczkolwiek jego struktura ułatwia implementację weryfikacji własności czasowych opisanych przy pomocy logik temporalnych.

Definicja 5 Czasowy system tranzycyjny *(akr. TTS, z ang. Timed Transition System) systemu, którego modelem jest struktura typu c-TdPN lub PSwPN, stanowi struktura* $S = (Q, Q_0, \Sigma, \longrightarrow)$ *gdzie:*

- Q *jest zbiorem stanów systemu,*
- $Q_0 \subseteq Q$ *to zbiór stanów początkowych systemu,*
- Σ *to skończony zbiór akcji systemu,*
- *a relacja* $\longrightarrow \subseteq Q \times (\Sigma \cup \mathbb{R}_+ \cup \{0\}) \times Q$ *to zbiór krawędzi reprezentujących zmianę stanu systemu.*$\square$

Interpretacja stanu czasowego systemu tranzycyjnego jest zależa od wybranego typu modelu systemu dla którego zachowanie (semantykę) reprezentuje.

Zmiana stanu systemu z q na q' może nastąpić na skutek finalizacji akcji $a \in \Sigma$ lub przez upływ $d \in R \cup \{0\}$ momentów. Zmiana reprezentowana jest przez $q \xrightarrow{x} q'$ co oznacza $(q, x, q') \in \longrightarrow$ gdzie $x \in \{a, d\}$. Dla $x = d$ możemy zapisać $q' = q + d$ co oznacza stan q po upływie d momentów [slr:czasu] bez wystąpienia jakiejkolwiek akcji systemu. Zarówno możliwość wystąpienia akcji jak i upływu czasu są zależne od typu modelu systemu.

Ścieżką (lub przebiegiem) w S nazwiemy maksymalną sekwencję następujących po sobie stanów spowodowanych naprzemiennie upływem jednostek czasu $d_i \in R \cup \{0\}$ i wystąpień akcji $a_i \in \Sigma$. Formalnie ścieżką ρ w S, zaczynającą się w stanie $q_i \in Q_0$ jest graf:

$$\rho = q_i \xrightarrow{d_i} (q_i + d_i) \xrightarrow{a_i} q_{i+1} \xrightarrow{d_{i+1}} (q_{i+1} + d_{i+1}) \xrightarrow{a_{i+1}} q_{i+2} \xrightarrow{d_{i+2}} \dots$$

W szczególnym przypadku ścieżkę można rozpocząć w dowolnym stanie $q \in Q$ nazywając ją sufiksem przebiegu.

Zbiór wszystkich ścieżek (przebiegów) rozpoczynających się w q jest reprezentowany przez $\pi(q)$. Ponadto sekwencję akcji $a_i, a_{i+1}, a_{i+2}, \dots$ w ścieżce ρ nazwiemy jej *śladem.*

2 Reprezentatywny fragment rozgałęzionego procesu

W tej części przedstawiono sposób modelowania zachowania w stylu rozwinięcia McMilana [17, 34] dla sieci PSwPN oraz (z założeniem ograniczenia cech modelu do

modelu TPN) sposób wyznaczania reprezentatywnego fragmentu (prefiksu zupełnego, opisany w [46],[32] oraz analogiczne rozwiązanie w [5]) zawierającego wszystkie uruchomienia umożliwionych tranzycji.

W prezentowanym rozwiązaniu założono po pierwsze, że liczba żetonów w miejscu jest mniejsza lub równa 1 oraz, że zachowania zeno nie są możliwe (zachowanie zeno to takie zachowanie, w którym tranzycja może być uruchamiana nieskończenie wiele razy w skończonym czasie).

$\Gamma(M) = (\beta, \theta, v)$ oznacza symboliczny rozgałęziony proces dla modelu systemu $M = (P, T, F, C, S, I, J, V, \widehat{V}, m)$ gdzie $\beta = (B, E, G, H, l)$ oraz $v \in \widehat{V}$.

Model zachowania ma aspekt strukturalny i aspekt czasowy. Aspekt strukturalny opisuje elementy składowe zachowania i połączenia przyczynowe pomiędzy nimi.

Natomiast aspekt czasowy charakteryzuje relacje pomiędzy akcjami systemu oraz momenty wystąpienia akcji w zachowaniu w odniesieniu do momentu zainicjowania systemu.

2.1 Strukturalny aspekt zachowania

Relacje H i G reprezentują przyczynowość w modelu zachowania. Relacje te definiują poprzednika (pre) i/lub następnika (post) dla prawie każdego elementu składowego zachowania.

Standardowy *prezbiór* i *postzbiór* (w tym kontekstowy) dla podzbioru $X \subseteq B \cup E$ i relacji $R \in \{G, H\}$ definiowane są jako: $RX = \{y : x \in X \wedge yRx\}$ i $XR = \{y : x \in X \wedge xRy\}$. W szczególności dla jednego elementu Rx oznacza $R\{x\}$ i xR oznacza $\{x\}R$. Pojęcia te definiowane są analogicznie dla $Y \subseteq l(B) \cup l(E)$ i $R \in \{F, C, S\}$.

Relacja H reprezentuje w $\Gamma(M)$ także łuki stoperów w M, co utrudnia rozróżnienie ich od łuków czytania. Zbiór warunków inicjujących tranzycję l(e) w zdarzeniu e, wyraża się wzorem $init(e) = (G \cup H \cap l^{-1}(C))e$.

Relacja *przyczynowości* pomiędzy dwoma węzłami $x, y \in B \cup E$, oznaczana przez $x < y$, zachodzi jeżeli relacja $G \cup H$ definiuje ścieżkę w grafie wychodzącą z x do y, w której przynajmniej dwa elementy są w relacji G. Relacja H sama w sobie nie definiuje przyczynowości.

Obecność w $\Gamma(M)$ elementów odpowiadających miejscom odczytu w strukturze M prowadzi do osłabienia występującej przyczynowości. Mówimy, że element $y \in B \cup E$ jest *słabo przyczynowo zależny* od elementu $x \in B$ i piszemy $x \nearrow y$ jeśli $x < y$ lub jeśli zbiory Hx i Gy mają wspólny element.

W konsekwencji relacja "słaba przyczynowość" prowadzi do zwiększenia zbiorów zdarzeń przyczynowo poprzedzających dane zdarzenie. Dla przykładu zdarzenie odcięcia zasobów $e0$, na rysunku 5, może nastąpić po zainicjowaniu systemu $\{\perp\}$, po pobraniu zadania do węzła pierwszego $\{\perp, e2\}$ lub drugiego $\{\perp, e3\}$. Stąd zdarzenie $e0$ może mieć trzy różne zbiory zdarzeń, które je poprzedzają.

Jeżeli pewne wykonania z podzbioru zdarzeń $X \subseteq E$ zużywają ten sam żeton lub relacja słabej przyczynowości powoduje cykl zawierający te wykonania, to podzbiór X jest konfliktowy, co oznaczymy przez $\#X$.

Mówimy, że zdarzenia $e, e' \in E$ są w *konflikcie bezpośrednim* e *conf* e' jeżeli ich konflikt jest spowodowany jedynie potrzebą zużycia wspólnych żetonów ($Ge \cap Ge' \neq \emptyset$).

Złożone akcje systemu są reprezentowane przez wykonania kolejno umożliwionych tranzycji, które sobie nawzajem nie przeszkadzają. Ciągi kolejnych wykonań tranzycji, które mają swój początek w stanie początkowym systemu reprezentują przebiegi zachowania tego systemu.

Formalnie *przebiegiem* w rozgałęzionym procesie $\Gamma(M)$ nazywamy podzbiór bezkonfliktowych zdarzeń $E' \subseteq E$, który jest co najmniej domknięty z dołu ze względu na relację $<$. W szczególności, zbiór zdarzeń koniecznych do wystąpienia zdarzenia e, zwany jego *historią*, symbolicznie oznaczany przez $\lceil e \rceil = \{e' \in E : e' < e\}$, jest przebiegiem.

Ze względu na relację $\nearrow$ zdarzenie e może mieć wiele historii. $\lceil e \rceil$ jest najmniejszą historią zdarzenia e. Historią zdarzenia e jest też zbiór powstały przez domknięcie zbioru $\lceil e \rceil$ relacją $\nearrow$ w $B \cup E$ tak, aby nie spowodować konfliktu pomiędzy zdarzenami.

Podzbiór warunków $B' \subseteq B$, które nie są w relacji przyczynowości ani w relacji konfliktu nazwiemy *warunkami równoległymi* (akr. co-zbiór).

Akcje zawarte w przebiegu zmieniają stan systemu, który jest reprezentowany przez maksymalny co-zbiór nazywany *odcięciem*. Odcięcie może być wyznaczone przez przebieg $E' \subseteq E$ jako $cut(E') = E'G - GE'$.

Przykładowo na rysunku 5 obliczone zadanie $b12$ w węźle pierwszym przed odcięciem zasobów jest następstwem zadania zadanego $b1$ stąd relacja $b1 < b12$. Odcięcie zasobów do czytania, gdy pobrano zadanie do węzeła 1, może mieć miejsce tylko wówczas gdy nastąpi po pobraniu zadania do obliczeń ze względu na powiązanie $e2 \nearrow e0$. Wykonywanie zadania przed odcięciem zasobów w węźle 1 jest w konflikcie bezpośrednim z wykonywaniem tego samego zadania w węźle 2 ($e2$ *conf* $e3$) ze względu na brak podzielności zadania $b1$. Podobnie $e2$ jest w konflikcie z $e4$ i $e6$.

Ponadto zbiór zdarzeń $\{\perp, e0, e1, e2, e5, e7\}$ jest bezkonfliktowym i dolnie domkniętym pod względem relacji $<$ zbiorem, a więc jest przebiegiem.

2.2 Aspekt czasowy zachowania

Przebieg E z funkcją θ przyporządkowującą jego zdarzeniom momenty wykonania jest nazywany *przebiegiem czasowym*, reprezentowanym przez (E, θ).

Funkcja θ ma tę własność, że moment $\theta(e)$ wykonania każdego zdarzenia następuje po umożliwieniu tranzycji $l(e)$ z opóźnieniem należącym do przypisanego temu zdarzeniu przedziałowi dostępności.

Niech $B' = cut(E')$ oznacza co-zbiór do którego doprowadził przebieg E'. *Moment zainicjowania tranzycji* t żetonami w miejscach $l(B')$ wyprodukowanymi przez ciąg kolenych wykonań $E' \subseteq E$ oznaczany jest przez

$$toe(B', t) = \max\{\theta(Gb) : b \in B' \wedge l(b) \in Ft \cup Ct\}.$$

Założenie $l(b) \in Ft \cup Ct$ pozwala pominąć warunki b odpowiadające żetonom w miejscach $l(b)$ z których wychodzi łuk stoperów do t, które to nie mają wpływu na inicjację.

Dla uproszczenia zakładamy, że zapis $\theta(Gb)$ jest momentem zdarzenia (jedynego) ze zbioru Gb.

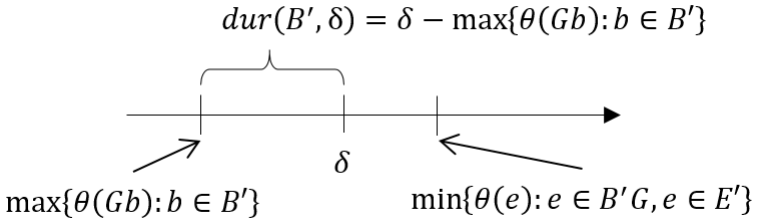
W przebiegu (E', θ) dla wykonania e możemy wyróżnić inne zdarzenia, które wykonały się przed e wykorzystując funkcję θ. Będą to po pierwsze zdarzenia z historii e, a po wtóre wszystkie pozostałe zdarzenia $e' \in E'$ takie, że $\theta(e') < \theta(e)$. Zależność ta reprezentowana jest zbiorem $earlier(e, E') = \{e' \in E' : \theta(e') < \theta(e)\}$.

Powiemy, że przekrój B' trwa lub jest dostępny w przebiegu (E', θ) określoną ilość czasu. Długość dostępności przekroju B' do pewnego globalnego momentu δ to odstęp w czasie pomiędzy najpóźniejszym momentem wykonania zdarzenia z $(G' \cup H)B'$, a najwcześniejszym momentem wykonania nowego zdarzenia korzystającego z B' lub momentem δ gdy będzie wcześniejszy. Zależność tą (rys. 6) wyrażamy wzorem:

$$dur(B', \delta) = \min\{\min\{\theta(e) : e \in B'G, e \in E'\}, \delta\} - \max\{\theta(Gb) : b \in B'\}$$

dla $\delta \geqslant \max\{\theta(Gb) : b \in B'\}$ oraz w przeciwnym razie:

$$dur(B', \delta) = 0.$$



Rysunek 6: Prezentacja $dur(B', \delta)$ na półosi czasu.

Ze względu na możliwość modelowania zawieszenia i wznowienia akcji systemu, tranzycja może mieć kilka różnych zbiorów powodujących jej postęp w przebiegu czasowym.

Zbiór wszystkich zbiorów powodujący postęp tranzycji t oznaczamy przez $acos(B'', t)$, gdzie B'' jest jej zbiorem inicjującym. Dlatego też zbiór B'' jest podzbiorem każdego zbioru z $acos(B'', t)$.

Długość postępu $adur(B'', t, \delta)$ tranzycji t, od momentu zainicjowania zbiorem B'' do pewnego momentu δ, jest sumą długości warunków dostępności $dur(A, \delta)$ powodujących jej A-ty postęp nie przekraczając momentu δ:

$$adur(B'', t, \delta) = \sum_{A \in acos(B'', t)} dur(A, \delta)$$

Jeżeli $adur(B'', t, \delta)$ jest nie mniejsze niż $I(t)$ oraz nie większe niż $J(t)$ to t jest umożliwiona.

Przykładowym przebiegiem czasowym kontynuowania wykonania zadania w węźle 2 (rys. 5) po wznowieniu zasobów do czytania, do momentu $d_2 = 14$ jest $E_2 = \{e0, e1, e5, e6\}$, gdzie $a_2 = b_2 = 1$ oraz funkcja θ_2 określa (przy założeniu $\theta_2(\perp) = 0$):

- odcięcie zasobów w momencie $\theta_2(e0) = 5$,
- przywrócenie zasobów w $\theta_2(e1) = 7$,
- w tym samym momencie pobranie zadania do wykonania w $\theta_2(e6) = 7$, oraz
- wymuszenie odcięcia zasobów w $\theta_2(e5) = 13$.

Funkcja θ_2 przebiegu czasowego musi spełniać ograniczenia przyporządkowane do rozgałęzionego procesu czasowego.

Zbiór $init(e6) = \{b1, b2\}$ inicjuje czynność tranzycji $T_{21} = l(e6)$, a moment jej zainicjowania to $toe(init(e6), l(e6)) = \theta(\perp)$. Dwa zbiory $acos(init(e6), l(e6)) = \{\{b1, b2, b3\}, \{b1, b2, b5\}\}$ powodują jej postęp. Jej sumaryczna długość postępu od momentu zainicjowania do momentu d_2 wyraża się wzorem:

$$\begin{aligned}
&adur(init(e6), l(e6), d_2) = \\
&dur(\{b1, b2, b3\}, 14) + dur(\{b1, b2, b5\}, 14) = \\
&(\theta_2(e0) - \theta_2(\perp)) + (\min\{\theta_2(e6), 14\} - \theta_2(e1)) = \\
&5 + 7 - 7 = 5.
\end{aligned}$$

Przed odcięciem zasobów postęp tranzycji T_{21} trwał 5 jednostek czasu, a po wznowieniu tranzycja została natychmiast uruchomiona.

Rozgałęziony proces czasowy agreguje przebiegi czasowe. Jeżeli dla przebiegu czasowego usuniemy konkretne wartości momentów wykonań tranzycji, a nałożymy ograniczenia dla momentów to otrzymamy wiele przebiegów czasowych reprezentowanych przez rozgałęziony proces czasowy.

Kontynuując przykład należy zmodyfikować konkretne wartości funkcji θ_2 na jej ograniczenia reprezentując moment:

- zainicjowania $\theta_3(\perp)$,
- odcięcia zasobów $\theta_3(\perp) + 5 \leqslant \theta_3(e0) \leqslant \theta_3(\perp) + 6$,
- przywrócenia zasobów $\theta_3(e0) + 2 \leqslant \theta_3(e1) \leqslant \theta_3(e0) + 3$,
- finalizacji tranzycji T_{21}: $\theta_3(\perp) + 6 \leqslant \theta_3(e6) \leqslant \theta_3(\perp) + 15$ i $\theta_3(e6) \leqslant \theta_3(e5)$,
- wymuszonego odcięcia zasobów $\theta_3(e1) + 5 \leqslant \theta_3(e5) \leqslant \theta_3(e1) + 6$ oraz
- ograniczenia do momentu $d_2 = 14$: $\forall_{e \in E_2} \theta_3(e) \leqslant 14$.

W związku z powyższym $adur(init(e6), l(e6), d_2) = \theta_3(e0) + \theta_3(e6) - \theta_3(e1)$ jest zależna od momentów wykonań $e0, e1$, $e6$ i ograniczenia $\theta_3(e6) \leqslant 14$. Zdarzenie $e6$ kontynuuje przerwanie wykonywania tranzycji T_{21} w zdarzeniu $e3$. $\theta_3(e0) - \theta_3(e1)$ to długość czasu przerwania.

2.2.1 Funkcja przyporządkowująca zdarzeniom momenty wykonania w symbolicznych rozgałęzionych procesach

Newralgicznym punktem aspektu czasu dla SRPC jest poprawne określenie momentów wykonania zawartych w nim zdarzeń, zachowujących relację przyczynowości, słabej przyczynowości, konfliktu oraz stałych czasowych modelu wewnątrz każdego przebiegu zapisanego w SRPC.

Funkcję momentu wykonania zdarzeń z E definiujemy jako θ o własnościach:

(1) moment inicjujący stan początkowy m jest równy zero ($\theta(\perp) = 0$),

(2) każdy cykl w E zawiera zdarzenie niemożliwe, oraz

(3) dla każdego zdarzenia $e \in E - \{\perp\}$ spełniony jest przynajmniej jeden z warunków:

1. $\theta(e) \neq \infty \wedge \max(\{\theta(Gb) : b \in Ge \cup He\}) \leqslant \theta(e) \wedge$
   $\wedge v(I)(l(e)) \leqslant adur(init(e), l(e), \theta(e)) \leqslant v(J)(l(e)) \wedge$
   $\wedge \forall_{\ e' \in E \ \wedge \ e' \text{ conf } e} \ \theta(e') = \infty \wedge$
   $\wedge \forall_{\ e' \in E \ \wedge \ e' \nearrow e} \ \theta(e') \leqslant \theta(e)$ lub
2. $\theta(e) = \infty \wedge \exists_{b \in Ge \cup He} \theta(Gb) = \infty$ lub
3. $\theta(e) = \infty \wedge$
   $\wedge \exists_{e' \in E}[(e \text{ conf } e' \vee e \nearrow e') \Rightarrow$
   $(\theta(e') \neq \infty \wedge adur(init(e), l(e), \theta(e')) \leqslant v(J)(l(e)))]$.

Pierwszy warunek oznacza, że moment wykonania e jest nie wcześniejszy niż wykonania poprzedzające, sumaryczny czas postępu tranzycji $l(e)$ mieści się w przedziale $[v(I)(l(e)), v(J)(l(e))]$, każde wykonanie, które jest w konflikcie bezpośrednim z e jest niemożliwe oraz każde wykonanie czytające warunki zużywane przez e ma moment nie późniejszy niż moment wykonania e.

Drugi przypadek zachodzi, gdy wykonanie e nie ma zapewnionych żetonów inicjujących, więc jest niemożliwe.

Ostatni przypadek występuje, gdy istnieje pewne pilniejsze możliwe wykonanie, które uniemożliwia wykonanie e ze względu na konflikt bezpośredni lub konieczny wcześniejszy moment wykonania niż e.

2.3 Symboliczne rozwinięcie dla PSwPN

Model SRPC może zawierać niepełną informację o ograniczeniach momentów wykonań tranzycji w nim zawartych. Dodanie do niego kolejnego wykonania tranzycji w skutkach może prowadzić do zawężenia lub nawet ze względu na cechę pilności uniemożliwić zawarte w SRPC wykonania tranzycji. Problem ten został rozwiązany przez odpowiedni dobór kolejności dodawania nowych wykonań do SRPC, zakładający pierwszeństwo najwcześniejszej pilności.

Uwzględniając dobór kolejności wg najwcześniejszej pilności powiemy, że $e = (l(e), Ge, He)$ jest *rozszerzeniem* dla $\Gamma(M)$ jeżeli:

- $e \notin E$, $l(e) \in T$,
- $\Gamma(M)$ zapewnia $Ge \cup He$ dla wykonania e tranzycji $l(e)$, tj. $Ge \cup He \subseteq B$ i $Ge \cup He$ to co-zbiór,
- wykonanie e jest możliwe i nie zablokuje możliwości wykonania innych zdarzeń $e' \in E$ już zapisanych w $\Gamma(M)$,
- wykonanie e może spowodować zaostrzenie ograniczeń momentu wykonania dla $e' \in E$.

Kluczowy jest zatem wybór takiego zdarzenia do rozszerzenia, aby miało najwcześniejszą pilność spośród wszystkich możliwych.

W wyniku rozszerzenia symbolicznego rozgałęzionego procesu $\Gamma(M)$ struktury M przez $e = (l(e), Ge, He)$ powstaje nowy SRPC $\Gamma'(M)$ taki, że:

- $E' = E \cup \{e\}$ i $E \cap \{e\} = \emptyset$,
- $B' = B \cup Q$ i $B \cap Q = \emptyset$, gdzie $Q = \{(b, e) : b = l^{-1}(p) \wedge p \in l(e)F\}$,
- funkcja θ' spełnia ograniczenia dla funkcji θ, dodatkowo określa ograniczenia momentu wykonania dla e oraz dla zdarzeń z E, które są w relacji ze zdarzeniem e.

Kolejność indeksowania zdarzeń i warunków na rysunku 5 nie jest przypadkowa. Prezentowany SRPC powstał przez rozszerzanie stanu początkowego m_1 o wykonania umożliwionych tranzycji w kolejności od najwcześniejszej pilności i jest otrzymanym w ten sposób najmniejszym SRPC zawierającym informacje o wykonaniu przynajmniej jednego zadania obliczeniowego przy zadanych parametrach. W konstruowanym w ten sposób SRPC po każdym rozszerzeniu $ei = (l(ei), G\{ei\}, H\{ei\})$ nie wystąpi sytuacja zablokowania wykonania zdarzenia ej w nim zawartego, gdzie $i = 0, 1, \ldots, 7$ i $j < i$.

2.4 Prefiks zupełny rozwinięcia TPN

Dodawanie nieskończenie wiele razy rozszerzeń do SRPC powoduje otrzymanie zupełnego rozwinięcia zawierającego informację o wykonaniu się wszystkich umożliwionych tranzycji sieci systemu M. W rezultacie taki model zachowania przez swoją nieskończoność zawiera nadmiar informacji o akcjach w systemie. Utrudnia to jego analizę i weryfikację własności modelu systemu zwłaszcza w praktyce, gdzie nie można skonstruować nieskończonego modelu zachowania.

W związku z tym przedstawiono znane rozwiązanie wyznaczania prefiksu zupełnego dla modelu PSwPN, z wyłączeniem parametrów, łuków czytania i stoperów, które gwarantuje przechowanie informacji o wszystkich umożliwionych uruchomieniach tranzycji od momentu zainicjowania systemu.

W rozwiązaniu wykorzystano znany z literatury sposób identyfikacji powtórnego wystąpienia tranzycji w zwykłych sieciach Petriego.

Istota rozwiązania polega na tym, że w obrębie każdego przebiegu czasowego odnajdywane są przekroje odpowiadające strukturalnie temu samemu stanowi systemu (jego znakowaniom). Jeżeli przekroje te (odcięcia) zwane równoważnymi powodują takie same warunki dla czynności zestawu tych samych tranzycji to możliwe jest zidentyfikowanie w takim każdym przebiegu czasowym powtórnego wykonania tranzycji.

Ponieważ modelem zachowania jest rozgałęziony proces czasowy, agregujący przebiegi czasowe, to równoważność jest określana ogólniej pomiędzy dwoma rozgałęzionymi procesami czasowymi zawierającymi w sobie przebiegi.

Dla każdego SRPC w kontekście rozwinięcia można wyznaczyć jego dopełnienie (do rozwinięcia) zwane *przyczynową przyszłością*. Powiemy, że dwa SPRC są równoważne jeżeli mają równoważną przyczynową przyszłość. Zatem mając dwa SRPC, które mają równoważną przyczynową przyszłość oraz wykonanie tranzycji, które rozszerza obszerniejszy SRPC, można stwierdzić, że zapis wykonania prowadzi już do powtórzenia informacji o zachowaniu się systemu.

Niech od tego momentu struktura $M = (P, T, F, I, J, m)$ oznacza szczególny przypadek PSwPN w którym nie występują łuki czytania ($C = \emptyset$), łuki stoperów ($S = \emptyset$), brak parametrów ($V = \emptyset$), a $\Gamma(M)$ reprezentuje jego symboliczny rozgałęziony proces czasowy. Tak określona struktura M jest znaną siecią TPN, rozważaną przez [35], a $\Gamma(M)$ reprezentuje jej zachowanie.

Z uwagi na brak parametrów w M, SRPC może być w uproszczeniu reprezentowany przez parę $\Gamma(M) = (\beta, \theta)$ będącą rozgałęzionym procesem czasowym (w skr. RPC). Dla każdej tranzycji t przedział $[I(t), J(t)]$ nie zawiera parametrów, co także upraszcza zapis.

2.4.1 Równoważność przyczynowej przyszłości rozgałęzionych procesów (równoważność stanów)

W RPC konieczne do zdefiniowania pojęcia równoważności przyczynowej przyszłości jest określenie, niezależnie od globalnego momentu $\theta(\perp)$, długości dostępności warunków będących zbiorami równoległymi.

Zredukowana długość dostępności warunku $b \in A$ w obrębie warunków $A \subseteq B$, które tworzą co-zbiór opisana jest wzorem:

$$age(b, \theta, A) = \min\{\max_{b' \in A}\{\theta(Gb')\} - \theta(Gb), \max\{K(t) : t \in T \wedge t \in l(b)F\}\}$$

gdzie $K(t) = J(t)$ gdy $J(t) \neq \infty$ oraz $K(t) = I(t)$ w przeciwnym razie (stąd nazwa zredukowana).

Wielkość $age(b, \theta, A)$ jest wyznaczona jako minimum z:

- odległości pomiędzy momentem wyprodukowania b, a momentem dostępności kompletnego zbioru A, oraz
- z najdłuższej możliwej czynności spośród tranzycji t zużywających b ($\max\{J(t) : t \in l(b)F\}$).

Jeżeli pewna tranzycja t nie ma ograniczenia pilnego uruchomienia ($J(t) = \infty$) to zamiennie za $J(t)$ sprawdzana jest najkrótsza możliwa aktywność $I(t)$. Wyrażenie $\max\{K(t) : t \in T \wedge t \in l(b)F\}$ pozwala na redukcję i rozwiązanie problemu zbyt późnych momentów wykonań produkujących warunki z A.

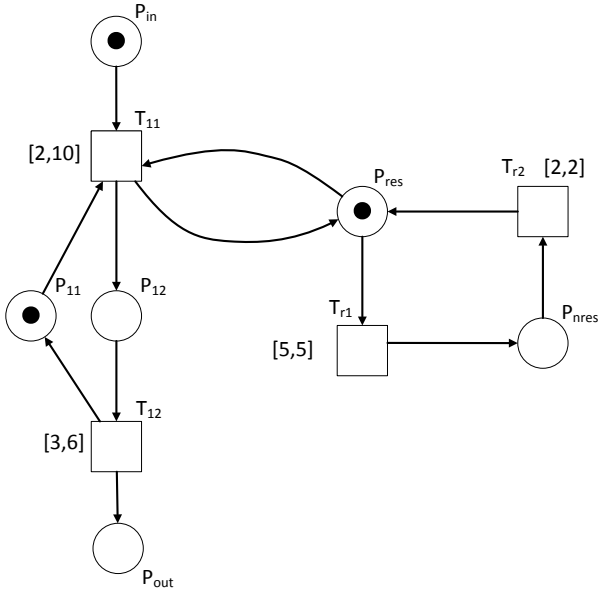
Wprowadzenie równoważności pomiędzy przyczynową przyszłością dwóch RPC pozwala na identyfikację zdarzenia wykonania tranzycji, które powiela informację o zachowaniu systemu.

Dla zbioru zdarzeń E oraz funkcji momentów θ określonej na E, symbol $E_{\theta<\infty} = \{e \in E : \theta(e) < \infty\}$ oznacza podzbiór bezkonfliktowych zdarzeń, które wystąpiły (θ określa skończony moment). Jest to wyodrębnienie ze zbioru E pewnego podzbioru zdarzeń będącego przebiegiem. Jeżeli E jest zbiorem zdarzeń w rozgałęzionym procesie (β, θ), to symbol $\beta_{\theta<\infty} = E_{\theta<\infty}$ jest wyodrębnionym przebiegiem z $\Gamma(M)$.

Dwa RPC (β, θ) i (β', θ') mają równoważną przyczynową przyszłość, jeżeli dla wszystkich odpowiadających sobie (jeden jest prefiksem drugiego) przebiegów $\beta_{\theta<\infty}$ i $\beta'_{\theta'<\infty}$ w $\Gamma(M)$:

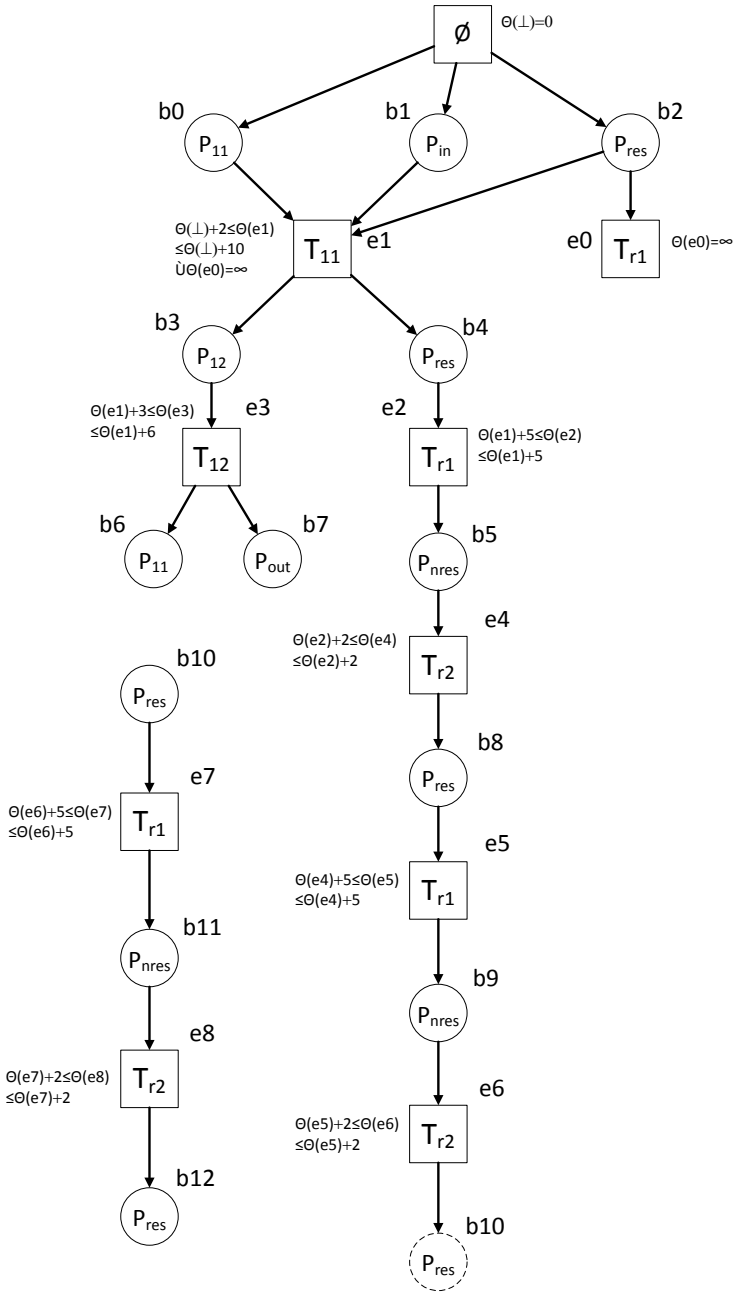
- $l(cut(\beta_{\theta<\infty})) = l(cut(\beta'_{\theta'<\infty}))$,
- $\forall_{b \in cut(\beta_{\theta<\infty})} \forall_{b' \in cut(\beta'_{\theta'<\infty})}$
  $l(b) = l(b') \Rightarrow age(b, \theta, cut(\beta_{\theta<\infty})) = age(b', \theta', cut(\beta'_{\theta'<\infty}))$.

Przykładowo niech model systemu z rysunku 4 ma niedostępny węzeł 2, wartości parametrów $a = b = 0$ oraz łuk czytania zamieniony na dwa łuki zwykłe.



Rysunek 7: M_1 model systemu TPN z Rysunku 4 z wyłączonym węzłem 2 i $a = b = 0$.

Niech modelem zachowania systemu (rys. 7) będzie rozgałęziony proces czasowy $\Gamma(M_1)$ (rys. 8), w którym założono dla uproszenia, że zasoby mogą być odcięte dopiero po pobraniu zadania obliczeniowego do węzeła. Stąd moment wykonania zdarzenia $e0$ jest równy ∞.



Rysunek 8: Model $\Gamma(M_1)$ reprezentujący fragment zachowania systemu M_1

Model zachowania w takim przypadku zawiera strukturalnie (pomijając informacje czasowe) jeden przebieg stanowiący wykonanie zadania w węźle pierwszym, a następnie naprzemienne odcinanie i przywracanie zasobów. Pozwala to w prosty sposób zauważyć powtarzający się fragment. Niech będą dane dwa prefiksy $\Gamma'(M_1) = (\beta', \theta')$ i $\Gamma''(M_1) = (\beta'', \theta'')$ dla $\Gamma(M_1) = (\beta_1, \theta_1)$, $\beta_1 = (B_1, E_1, G_1, \emptyset, l)$.

$\Gamma''(M_1)$ reprezentuje bezzwłocznie wykonanie zadania oraz trzykrotne odcięcie i przywrócenie zasobów, gdzie $B'' = B_1$, $E'' = E_1$, $G'' = G_1$ oraz funkcja momentu wykonania θ'' określa konkretne momenty:

- $\theta''(e1) = 2$ pobrania zadania do węzła 1,
- $\theta''(e2) = 7$ odcięcia zasobów,
- $\theta''(e3) = 5$ ukończenia zadania oraz
- naprzemiennie momenty przywrócenia zasobów ($\theta''(e4) = 9$, $\theta''(e6) = 16$, $\theta''(e8) = 23$) i odcięcia zasobów ($\theta''(e5) = 14$, $\theta''(e7) = 21$),

spełniając przy tym ograniczenia nałożone na θ_1.

W $\Gamma'(M_1)$ zasoby odcięto i przywrócono tylko dwa razy. $\Gamma'(M_1)$ jest prefiksem dla $\Gamma''(M_1)$, w którym $B' = B'' - \{b11, b12\}$, $E' = E'' - \{e7, e8\}$, relacja G' to ograniczenie relacji G'' do zbioru $B' \cup E'$, a funkcja θ' to zawężenie funkcji θ'' do E'.

Rozgałęzione procesy czasowe $\Gamma'(M_1)$ i $\Gamma''(M_1)$ mają równoważną przyczynową przyszłość ponieważ spełnione są następujące warunki (dla ich odpowiadających sobie przebiegów):

1. $l(cut(\beta'_{\theta'<\infty})) = l(cut(\beta''_{\theta''<\infty}))$,
2. $\forall_{b' \in cut(\beta'_{\theta'<\infty})} \forall_{b'' \in cut(\beta''_{\theta''<\infty})}$
   $l(b') = l(b'') \Rightarrow age(b', \theta', cut(\beta'_{\theta'<\infty})) = age(b'', \theta'', cut(\beta''_{\theta''<\infty}))$.

W rozważanych RPC są jedynie dwa przebiegi do sprawdzenia.

Po pierwsze należy zauważyć, że $\beta'_{\theta'<\infty} = E'_{\theta'<\infty} = E' - \{e0\}$ oraz $\beta''_{\theta''<\infty} = E''_{\theta''<\infty} = E'' - \{e0\}$.

Następnie, zgodnie z punktem 1, $L = l(cut(\beta'_{\theta'<\infty})) = l(cut(E' - \{e0\})) = l(\{b6, b7, b10\}) = \{P_{11}, P_{out}, P_{res}\}$ oraz $P = l(cut(\beta''_{\theta''<\infty})) = l(cut(E'' - \{e0\})) = l(\{b6, b7, b12\}) = \{P_{11}, P_{out}, P_{res}\}$. Należy zauważyć, że $b10$ i $b12$ reprezentują wystąpienie żetonu w miejscu P_{res}. Stąd $L = P$.

Po wtóre należy sprawdzić równość zredukowanego wieku dla par warunków $(b6, b6)$, $(b7, b7)$ i $(b10, b12)$. Przykładowo dla ostatniej pary $l(b10) = l(b12)$:
$L = age(b10, \theta', cut(\beta'_{\theta'<\infty})) = age(b10, \theta', \{b6, b7, b10\}) =$
$\min\{\max_{b' \in \{b6,b7,b10\}} \{G'b'\} - G'b10, \max\{K(t) : t \in T_1 \wedge t \in l(b10)F_1\}\} =$
$\min\{\max\{\theta'(e3), \theta'(e6)\} - \theta'(e6), \max\{J(T_{11}), J(T_{r1})\}\} =$
$\min\{\max\{5, 16\} - 16, \max\{10, 5\}\} = 0.$
$P = age(b12, \theta'', cut(\beta''_{\theta''<\infty})) =$
$\min\{\max\{\theta''(e3), \theta''(e8)\} - \theta''(e8), \max\{J(T_{11}), J(T_{r1})\}\} =$

$\min\{\max\{5,23\}-23, \max\{10,5\}\} = 0$. Stąd $L = P$ więc dla pary $(b10, b12)$ zredukowany wiek jest równy.

Analogicznie należy sprawdzić, że dla par $(b6, b6)$ i $(b7, b7)$ zredukowany wiek także jest równy.

2.4.2 Zdarzenie odcięcia

Do zdefiniowania zdarzenia odcięcia potrzebne jest pojęcie znanego z literatury adekwatnego porządku przebiegów.

Relacja $\prec$, określona na zbiorze przebiegów (bez czasu), jest *adekwatnym porządkiem* jeżeli:

- $\prec$ jest zwrotna i przechodnia,
- $\prec$ ustanawia porządek prefiksowy: jeżeli $C \subset C'$ to $C \prec C'$,
- $\prec$ zachowuje rozszerzenia.
- implikacja: $C \prec C' \Rightarrow C \cup \{e\} \prec C' \cup \{e'\}$, jest spełniona gdy $cut(C) = cut(C')$, dla wszystkich rozszerzeń $e = (t, Ge, \emptyset)$ dla C i odpowiadających im rozszerzeń $e' = (t, Ge', \emptyset)$ dla C'.

Następnie dla dowolnego zdarzenia $e \in E$ niech $Past(e)$ oznacza zbiór zawierający wszystkie najmniejsze (ze względu na liczbę zawartych zdarzeń) RPC w których zawarto historię zdarzenia e. W szczególnym przypadku $Past(e)$ może być jednoelementowym zbiorem, gdzie jedynym elementem jest $\lceil e \rceil$.

Rozszerzenie e' w $\Gamma(M)$ jest *zdarzeniem odcięcia* jeżeli każdy $(\beta', \theta') \in Past(e')$ zawiera inne (wcześniejsze) zdarzenie e, oraz istnieje taki $(\beta, \theta) \in Past(e)$, że:

- $\lceil e \rceil \prec \lceil e' \rceil$ są w relacji adekwatnego porządku,
- $l(cut(\lceil e \rceil)) = l(cut(\lceil e' \rceil))$, powodują wystąpienie żetonów w tych samych miejscach,
- (β, θ) i (β', θ') mają równoważną przyczynową przyszłość.

Jeżeli z modelu sieci zostanie usunięty aspekt czasu to pojęcie zdarzenia odcięcia redukuje się do pierwszych dwóch warunków, jak w przypadku zdarzenia odcięcia dla zwykłej sieci Petriego.

2.4.3 Reprezentatywny fragment

Identyfikacja wszystkich zdarzeń odcięcia dla RPC pozwala następnie na identyfikację reprezentatywnego fragmentu zachowania.

Przykładowo dla RPC z rysunku 8 zdarzeniem odcięcia będzie wykonanie tranzycji T_{r1} w rozszerzeniu $e = (T_{r1}, \{b12\}, \emptyset)$ ponieważ:

- $\lceil e \rceil = \lceil e7 \rceil \cup \{e7, e8\}$, zatem $\lceil e7 \rceil \prec \lceil e \rceil$ to prawda,

- $L = l(cut(\lceil e7 \rceil)) = l(cut(\{e1, e2, e3, e4, e5, e6\})) = l(\{b6, b7, b10\}) = \{P_{11}, P_{out}, P_{res}\}$,
  $P = l(cut(\lceil e \rceil)) = l(cut(\lceil e7 \rceil \cup \{e7, e8\})) = l(\{b6, b7, b12\}) = \{P_{11}, P_{out}, P_{res}\}$,
  $L = P$,

- Niech $(\beta, \theta) = \Gamma(M_1)$ oraz (β', θ') to zawężenie $\Gamma(M_1)$ do $E' = E - \{e7, e8\}$, $B' = B - \{b11, b12\}$, θ' to zawężenie θ do zbioru E'. Tutaj $\beta_{\theta<\infty}$ i $\beta'_{\theta'<\infty}$ są przebiegami, które mają ograniczenia dla momentów wykonań zdarzeń możliwych. Stąd:

  1. $l(cut(\beta'_{\theta'<\infty})) = l(cut(\beta_{\theta<\infty}))$, $L = l(E') = l(\{b6, b7, b10\}) = \{P_{11}, P_{out}, P_{res}\} = l(\{b6, b7, b12\}) = P$
  2. Następnie dla każdej pary $(b6, b6)$, $(b7, b7)$ i $(b10, b12)$ ma być równy zredukowany wiek. Dla ostatniej pary:
     $L = age(b10, \theta', cut(\beta'_{\theta'<\infty})) = age(b10, \theta', \{b6, b7, b10\}) =$
     $\min\{\max_{b' \in \{b6,b7,b10\}}\{G'b'\} - G'b10, \max\{K(t) : t \in T_1 \wedge t \in l(b10)F_1\}\} =$
     $\min\{\max\{\theta'(e3), \theta'(e6)\} - \theta'(e6), \max\{J(T_{11}), J(T_{r1})\}\} =$
     $\min\{\theta'(e6) - \theta'(e6), \max\{10, 5\}\} = 0.$
     Należy zauważyć, że $\max\{\theta'(e3), \theta'(e6)\} = \theta'(e6)$ ponieważ najpóźniejszy możliwy moment $\theta'(e3)$ to $\theta'(e1) + 6$, a najwcześniejszy możliwy moment $\theta'(e6)$ to $\theta'(e1) + 15$.
     Podobnie: $L = age(b12, \theta, cut(\beta_{\theta<\infty})) =$
     $\min\{\max\{\theta(e3), \theta(e8)\} - \theta(e8), \max\{J(T_{11}), J(T_{r1})\}\} =$
     $\min\{0, \max\{10, 5\}\} = 0.$
     Stąd $L = P$.

A więc prefiks $\Gamma(M_1)$ jest prefiksem zupełnym.

3 Reprezentatywny fragment grafu stref

3.1 Dyskretyzacja zachowania

Ze względu na problem eksplozji stanów automatyczna weryfikacja własności czasowych rozgałęzionego procesu czasowego sprawia problem w praktyce.

W celu rozwiązania tego problem stosuje się dyskretyzację przestrzeni stanów bowiem model zachowania po dyskretyzacji jest łatwiejszy do weryfikacji własności czasowych procesów. Dla demonstracji dyskretyzacji wybrano model TTS, a do opisu własności logikę temporalną TCTL, opisaną w kolejnym rozdziale.

3.1.1 TTS jako model zachowania dla TPN

Niech $S = (Q, Q_0, \Sigma, \longrightarrow)$ będzie czasowym systemem tranzycyjnym, gdzie dla przypomnienia Q jest zbiorem stanów, $Q_0 \subseteq Q$ zbiorem stanów początkowych, Σ skończonym zbiorem akcji, a relacja $\longrightarrow \subseteq Q \times (\Sigma \cup \mathbb{R}_+ \cup \{0\}) \times Q$ reprezentuje zbiór krawędzi.

Upływ czasu dla tranzycji reprezentowany jest przez nadawanie wartości zegarom przypisanym do tranzycji. Każda tranzycja $t_i \in T$ ma przyporządkowany dokładnie jeden zegar $x_{t_i} \in X$ (lub skrótowo $x_i \in X$), a X oznacza zbiór wszystkich zegarów. Upływ czasu w zegarach jest równomierny (różne prędkości upływu czasu dla zegarów nie są możliwe) i zgodny z zegarem globalnym.

Funkcja $v : T \to \mathbb{R}_+ \cup \{0\}$ przypisuje każdej tranzycji t_i wartość $v(t_i)$ reprezentującą wskazanie zegara x_i. Wzór $(v + d)(t) = v(t) + d$ reprezentuje upływ d jednostek czasu w zegarze x_t i jest możliwy dla tranzycji czynnej t', jeżeli $v(t) + d \leqslant J(t)$. Nie należy utożsamiać funkcji v z funkcją nadającą wartości parametrów w modelu PSwPN.

Semantyka systemu $M = (P, T, F, I, J, m_0)$ jest reprezentowana przez czasowy system tranzycyjny $S_M = (Q, \{q_o\}, \Sigma, \longrightarrow)$, gdzie:

- $Q = (P \to \mathbb{N} \cup \{0\}) \times (T \to \mathbb{R}_+ \cup \{0\})$,
- $q_0 = (m_0, v_0)$, gdzie v_0 jest funkcją przyporządkowującą każdej tranzycji zerową wartość zegara.
- $\Sigma = T$,
- $\longrightarrow \in Q \times (T \cup \mathbb{R}_+ \cup \{0\}) \times Q$,

Symbol $Y \to X$ oznacza wszystkie możliwe funkcje przekształcające zbiór Y w X, i tak $P \to \mathbb{N} \cup \{0\}$ oznacza wszystkie możliwe znakowania, a $T \to \mathbb{R}_+ \cup \{0\}$ wszystkie możliwe funkcje wartościujące zegary (wskazania zegarów) tranzycji z sieci M. Zatem stan systemu zawiera informację o znakowaniu i wskazaniu zegarów.

Zmiana stanu na skutek upływu czasu d, zwana *ciągłą*, jest definiowana jako: $(m, v) \xrightarrow{d} (m, v')$ wtw gdy $v' = v + d$ oraz każda tranzycja t czynna w znakowaniu m nie przekracza momentu swojej pilności: $v(t) + d \leqslant J(t)$.

Zmiana stanu na skutek wykonania tranzycji t, zwana *dyskretną*, jest definiowana jako: $(m, v) \xrightarrow{a} (m', v')$ wtw gdy akcja a reprezentuje wykonanie czynnej w m tranzycji t, $m' = (m - Ft) \cup tF$, $I(t) \leqslant v(t) \leqslant J(t)$ (t jest umożliwiona), oraz każda nowo zainicjowana tranzycja t' w m' ma zresetowany zegar $x_{t'}$ co powoduje przypisanie $x_{t'} = 0$. Zakłada się, że zegary pozostałych tranzycji nie uległy zmianie w m'.

3.1.2 Dyskretyzacja TTS bazująca na strefach

Modelowanie własności w systemach TPN jest możliwe. Jednakże, automatyczna weryfikacja może sprawiać problemy, ze względu na eksplozję stanów. Dlatego też konieczne jest zastosowanie metody pozwalającej uniknąć eksplozji stanów. Stąd konieczność dyskretyzacji przestrzeni stanów w modelach typu TPN.

Przestrzeń stanów reprezentowana przez TTS dla czasowego systemu posiada nieprzeliczalnie wiele stanów. Zabieg dyskretyzacji ma na celu pofragmentowanie przestrzeni stanów tak, aby ułatwić i umożliwić automatyczną weryfikacją własności opisanych w logikach temporalnych.

Przy dyskretyzacji przestrzeni stanów S_M, podstawowym pojęciem jest *strefa* (def 8,str 7 [9]) lub region. Do dalszych rozważań wybrano strefę ze względu na praktyczną implementację [16].

Algorytm generujący prefiks grafu stanów wyznacza graf kolejnych klas abstrakcji stanów stanowiących węzły i uruchomień tranzycji stanowiących łuki. Graf ma skończoną liczbę węzłów co pozwala na determinację warunku stopu przy jego konstrukcji.

Dla wygody zapisu wskazanie zegara x_i reprezentowane przez $v(t_i)$ jest utożsamiane z symbolem x_i.

Strefa Z określona na zbiorze zegarów X jest wypukłym zbiorem wartosciowań zegarów reprezentowanym przez koniunkcję wyrażeń postaci: $x_i - x_j \prec c_{ij}$, $x_i \prec c_{i0}$ lub $-x_j \prec c_{0j}$, gdzie $x_i, x_j \in X$, $c_{ij}, c_{i0}, c_{0j} \in \mathbb{Q} \cup \{-\infty, \infty\}$ oraz $\prec \in \{<, \leqslant, =, >, \geqslant\}$. Strefy z nieuproszczoną postacią oganiczeń zegarów są kłopotliwe w porównywaniu pomiędzy sobą. Postać kanoniczna strefy upraszcza zapis zawartych w niej ograniczeń zegarów.

Postać kanoniczna strefy to nowa strefa, określona nad zbiorem zegarów X, której ograniczenia z_{ij} są zapisane w postaci $x_i - x_j \prec z_{ij}$ dla $x_i, x_j \in X \cup \{x_0\}$, $z_{ij} \in \mathbb{Q} \cup \{\infty\}$, $\prec \in \{<, \leqslant\}$, x_0 jest zegarem stale pokazującym 0 oraz dla każdej pary wskazań zegarów x_i, x_j, łącznie z zegarem zerowym, zachodzi $z_{ij} = \sup(x_i - x_j)$. Operator $\sup(x_i - x_j)$ oznacza kres górny różnicy $x_i - x_j$ w strefie Z.

Dla strefy Z określonej nad zbiorem zegarów $X \cup \{x_0\}$ definiujemy pojęcie:

- zawężenia, reprezentującego osłabienia ograniczeń przez eliminację zegarów ze strefy,
- przyszłosci, usunięcie górnych ograniczeń dla reprezentowanych zegarów,
- stanu symbolicznego, ograniczającego stany systemu S, na który składa się znakowanie i strefa.

Ograniczenie $Z_{|X'}$ powstaje przez przekształcenie strefy Z do postaci kanonicznej, a następnie usunięciu z niej zegarów ze zbioru $X - X'$.

Przyszłością strefy Z jest strefa $\overrightarrow{Z}$ powstała przez przekształcenie Z do postaci kanonicznej, a następnie zamiany ograniczeń $x_i - x_0 \prec z_{i0}$, na ograniczenia $x_i - x_0 \prec \infty$, przy czym $x_i \neq x_0$.

Symboliczny stan systemu TPN jest parą (m, Z), gdzie m jest znakowaniem, a Z jest strefą nad zbiorem tych zegarów dla których przyporządkowana tranzycja jest czynna w znakowaniu m, plus zegar zerowy.

Upływ czasu lub uruchomienie umożliwionej tranzycji może zmienić symboliczny stan, stąd dla stanu $s = (m, Z)$ i tranzycji t zainicjowanej przez m definiujemy dwa rodzaje następstw.

Jeżeli tranzycja t jest umożliwiona w stanie s, to *zbiór dyskretnych następników* stanu s otrzymany przez uruchomienie t ma postać:

$$Post_t(s) = (m', Z')$$

gdzie:

- $m' = (m - Ft) \cup tF$ oraz
- $Z' = ((Z \cap \{x_t \geqslant I(t)\})_{|O}) \cap \bigcap_{x \in N}\{x = 0\}$, zbiór O zawiera zegary stowarzyszone z tranzycjami, które były czynne w m i kontynuują swoją czynność w m' (łącznie z zegarem zerowym) oraz zbiór N zawiera zegary tranzycji nowo lub ponownie zainicjowanych w m'.

Zbiór czasowych następników stanu s jest symbolicznym stanem:

$$\overrightarrow{Post}(s) = (m, Z')$$

gdzie $Z' = \overrightarrow{Z} \cap \bigcap_{x_j \in X - \{x_0\}}\{x_j \leqslant J(t_j)\}$.

Nieformalnie symboliczny stan $Post_t(s)$ reprezentuje wszystkie stany które są osiągalne ze stanu s przez dyskretne uruchomienie tranzycji t (w dowolnym momencie w czasie jej umożliwienia). Natomiast symboliczny stan $\overrightarrow{Post}(s)$ jest zbiorem wszystkich stanów, które są osiągalne z s przez upływ czasu bez zmiany znakowania m. Czas może upływać maksymalnie do osiągnięcia pilnosci uruchomienia jednej z czynnych tranzycji. W szczególnosci $\overrightarrow{Post}_t(s) = \overrightarrow{Post}(\overrightarrow{Post}_t(s))$ oznacza symboliczny stan osiągalny ze stanu s po wykonaniu superpozycji obu następstw.

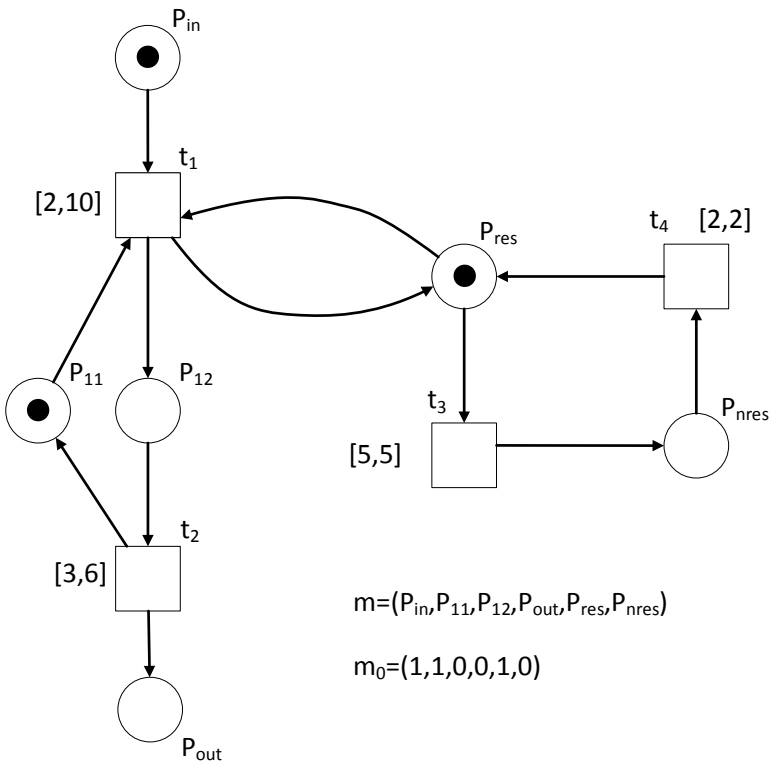
Algorytm generujący zdyskretyzowaną przestrzeń stanów reprezentowaną przez zbiór $Pass$, poczynając od stanu początkowego, interacyjnie dodaje kolejne nowe symboliczne stany do zbioru $Pass$, aż do momentu braku takich stanów. Na podstawie zbioru $Pass$ oraz pamiętanych uruchomień tranzycji możliwe jest wyznaczenie grafu symbolizującego przestrzeń stanów po dyskretyzacji.

Algorytm 1: Algorytm generujący przestrzeń stanów symbolicznych

Dane:
$s_0 = (m_0, Z_0)$
$Wait := \{\overrightarrow{Post}(s_0)\}$
$Pass = \emptyset$
Rezultat: Wygenerowana przestrzeń stanów, zapisana w $Pass$
while $Wait \neq \emptyset$ **do**
  $s = pop(Wait)$
  if $s \notin Pass$ **then**
    for $t \in enabled(s)$ **do**
      $s' = \overrightarrow{Post}(Post_t(s))$
      $Wait = Wait \cup \{s'\}$
    $Pass = Pass \cup \{s\}$

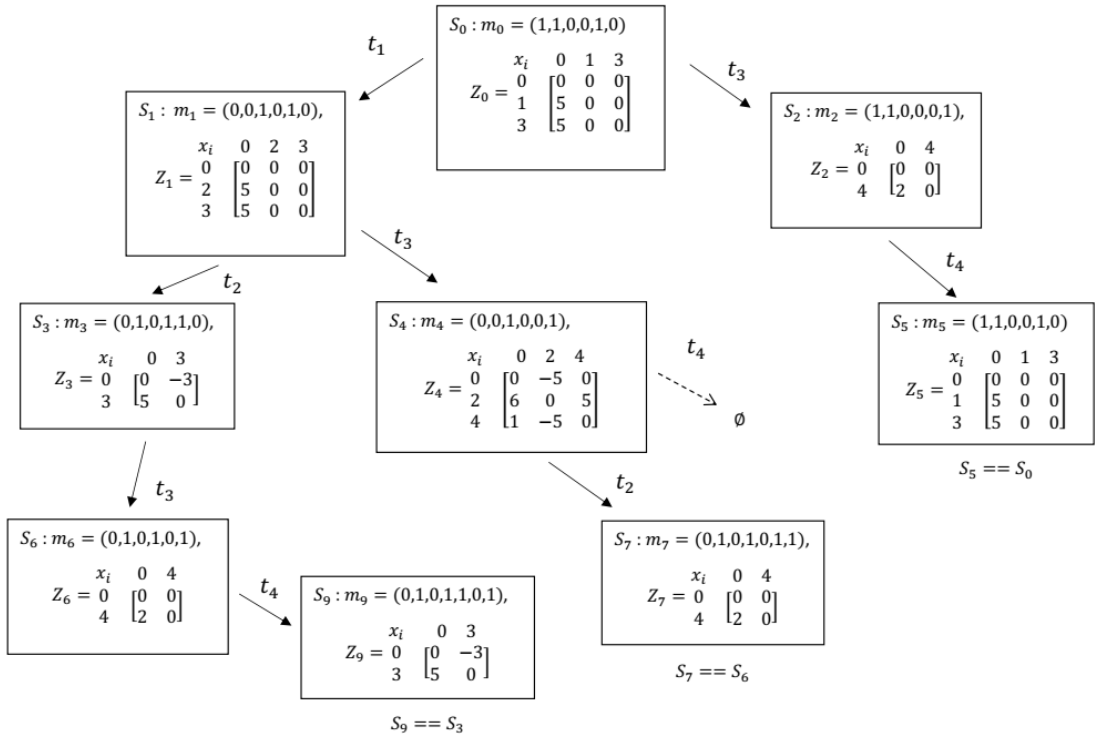
Tak opisany algorytm, przy założeniu, że w pewnym momencie nie będzie nowych stanów, można zastosować w praktyce do konstrukcji prefiksu zachowania (np. przy wykorzystaniu wzorca projektowego *kompozyt* w programowaniu).

Przykład 5 *Niech będzie dany model jednowęzłowego systemu obliczeń M_2 typu TPN.*



Rysunek 9: Model M_2 typu TPN realizujący obliczenia na jednym węźle.

M_2 to model M_1 z rysunku 7, w którym zmieniono dla jasności opisu nazwy tranzycji. Dla tegoż modelu wygenerowano algorytmem nr 1 zbiór stanów symbolicznych Pass. Dodatkowo odnotowano w celu zapamiętania tranzycje powodujące zmianę stanu znakowania. Następnie wyznaczono graf, którego wierzchołkami są symboliczne stany systemu, a krawędzie opatrzone tranzycjami je powodującymi.□



Rysunek 10: Graf stanów symbolicznych reprezentujący przestrzeń stanów modelu M_2.

Należy zauważyć, że na rysunku węzeł s_0 to stan początkowy po zastosowaniu $\overrightarrow{Post}$. Przy tym na jeden węzeł s_k składa się:

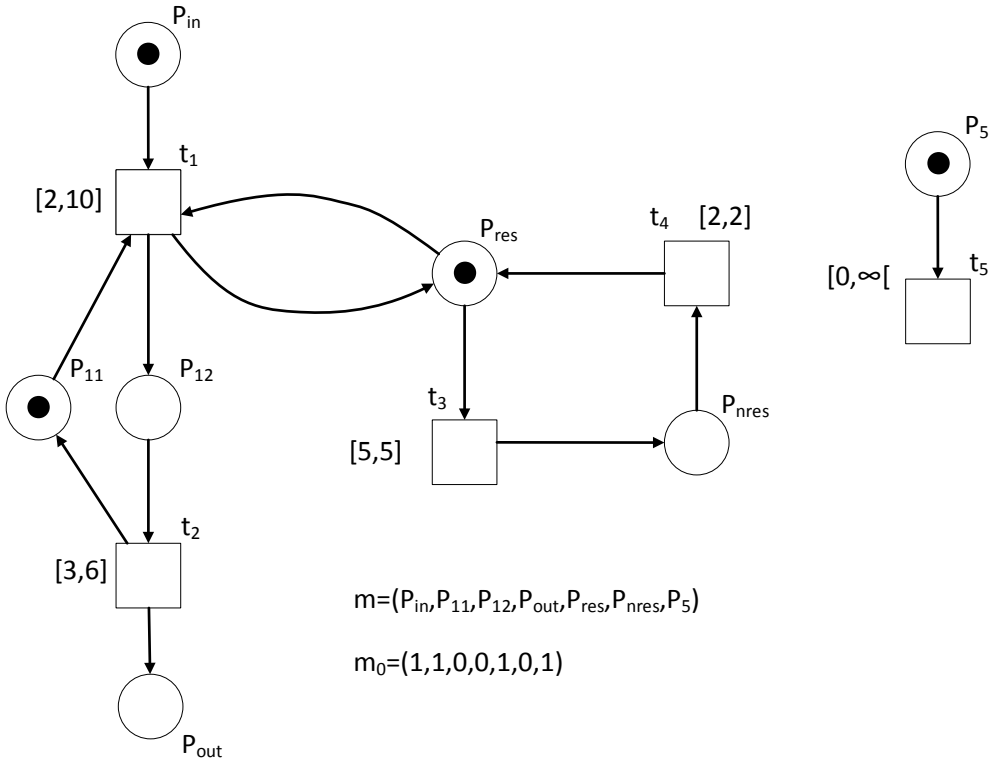
$$m_k = (p_0, p_1, \ldots, p_n),$$

$$Z_k = \begin{array}{c} \\ x_0 \\ x_{i_1} \\ \ldots \\ x_{i_l} \end{array} \begin{array}{c} \begin{array}{cccc} x_0 & x_{i_1} & \ldots & x_{i_l} \end{array} \\ \begin{pmatrix} z_{00} & z_{01} & \ldots & z_{0l} \\ z_{10} & z_{11} & \ldots & z_{1l} \\ \ldots & \ldots & \ldots & \ldots \\ z_{l0} & z_{l1} & \ldots & z_{ll} \end{pmatrix} \end{array}$$

znakowanie m_k, lista zegarów $x_0, x_{i_1}, x_{i_2}, ...x_{i_l} \in X \cup \{x_0\}$, gdzie $i_1, i_2, \ldots, i_l$ to indeksy tranzycji, a z_{rq} to ograniczenie dla $x_{i_r} - x_{i_q}$. W szczególności $x_{i_0} := x_0$.

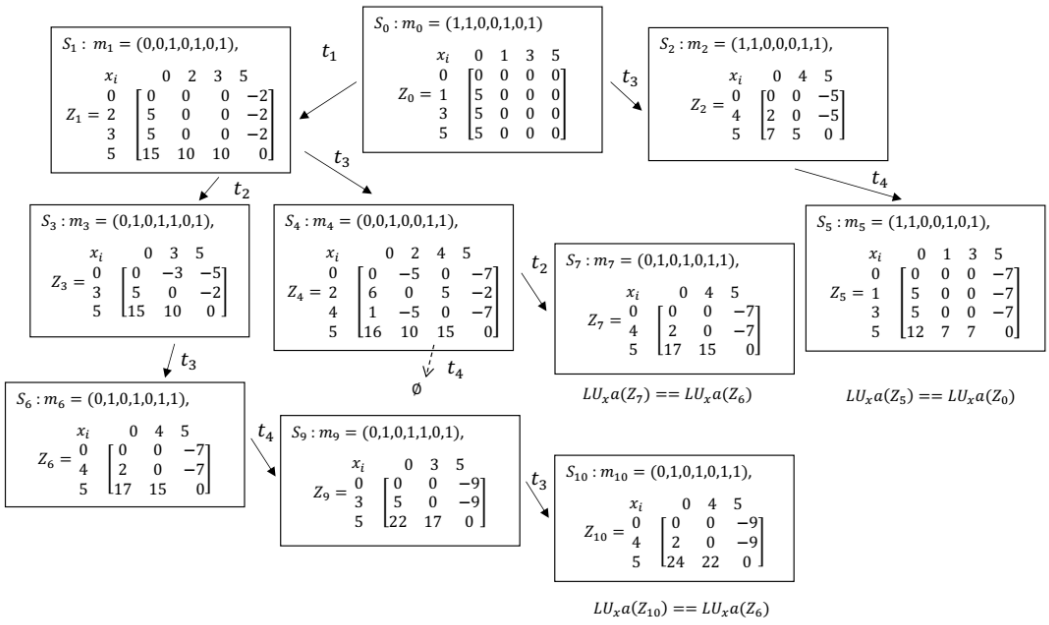
W tak otrzymanym prefiksie umożliwienie przechowywania informacji odnoszących się do czasu globalnego (od momentu inicjacji) odbywa się przez sztuczne dodanie umożliwionej tranzycji do modelu systemu, która nie może zostać wykonana.

Przykład 6 *Niech model M_2 z poprzedniego przykładu, zostanie rozszerzony o dodatkową umożliwioną tranzycję t_5. Założono, że tranzycja t_5 nie może być wykonana.*



Rysunek 11: Model M_2 rozszerzony o tranzycję t_5.

Dla tegoż modelu także wygenerowano algorytmem nr 1 zbiór stanów symbolicznych Pass. Dodatkowo odnotowano w celu zapamiętania tranzycje powodujące zmianę stanu znakowania. Następnie wyznaczono graf, którego wierzchołkami są symboliczne stany systemu, a krawędzie opatrzone tranzycjami je powodującymi. W tym przypadku obecność dodatkowej tranzycji dostarcza informacji o zależności pomiędzy momentami wykonań tranzycji modelu a momentem rozpoczęcia działania systemu. Powoduje to jednak potrzebę modyfikacji warunku stopu algorytmu nr 1.□



Rysunek 12: Graf stanów symbolicznych reprezentujący przestrzeń stanów modelu M_2 rozszerzonego o tranzycję t_5.

Brak końca identyfikacji nowych stanów ma miejsce, gdy w modelu systemu jest obecna tranzycja bez górnego ograniczenia (np. tranzycja t_5 na rysunku 11) dla czasu jej czynności. Powoduje to brak dostatecznego warunku stopu algorytmu. Rozwiązaniem problemu stopu jest w tym przypadku zastosowanie tzw. aproksymacji stref [14], pozwalającej porównywać stany symboliczne. Dzięki temu, w pewnej iteracji algorytmu, brak jest nowych stanów, których aproksymacja wcześniej nie wystąpiła.

3.1.3 Aproksymacja strefy

Aproksymacja strefy, w kontekście systemu sieciowego z określonym znakowaniem, to funkcja przekształcająca strefę w strefę.

Funkcja ta osłabia ograniczenia zegarów strefy tak, że nie powoduje osłabienia ograniczeń wykonań tranzycji umożliwionych w powiązanym znakowaniu.

Aproksymacja strefy pozwala wychwycić ponowne wystąpienie stanu symbolicznego, który nie wnosi nowych informacji o zachowaniu (ani o znakowaniach ani o sekwencjach tranzycji).

Możliwe jest uzyskanie równości aproksymacji stref, które mają takie same zestawy zegarów, ale ograniczenia nie są identyczne, za to są takie same po aproksymacji. Następnie równość może być wykryta na etapie eliminacji redundancji danych, gwarantując przy tym spełnienie warunku stopu dla algorytmu nr 1.

W literaturze jest opisanych kilka aproksymacji. Do przykładu z siecią TPN wybrano znaną aproksymację LU_x, stosowaną dla semantyk czasowych automatów [4], dostosowaną do tej sieci.

Aproksymacja LU_x, oznaczana przez $LU_x a(Z) = Z'$, gdzie dla każdej pary zegarów $x_i, x_j \in X \cup \{x_0\}$ nowe ograniczenie z'_{ij} wyraża się wzorem:

$$z'_{ij} = \begin{cases} \infty & dla \quad z_{ij} > I(t_i) \\ \infty & dla \quad J(t_j) = \infty \\ z_{ij} & dla \quad pozostałych \end{cases}$$

Wzór zapewnia osłabienie ograniczenia z_{ij} gdy:

- $z_{ij} > I(t_i)$, ponieważ wskazanie zegara x_i, ewentualnie pomniejszone o x_j jest już większe niż najwcześniejszy moment umożliwienia t_i
- $J(t_j) = \infty$, ponieważ t_j nie jest nigdy pilna.

Przykład 7 *Dla stanu symbolicznego s_{10}, którego elementami jest strefa Z_{10}, gdzie $X_{10} = \{x_0, x_4, x_5\}$, oraz znakowanie $m_{10} = (0, 1, 0, 1, 0, 1, 1)$, spowodowanego pobraniem zadania do wykonania do węzła (t_1), wykonania zadania (t_2), odcięcia (t_3), przywrócenia (t_4) i znów odcięciem zasobów (t_3), obliczamy aproksymację:*

$$Z_{10} = \begin{bmatrix} 0 & 0 & -9 \\ 2 & 0 & -9 \\ 24 & 22 & 0 \end{bmatrix}; Z'_{10} = LU_x\text{-}Approx(Z_{10}) \begin{bmatrix} 0 & 0 & \infty \\ 2 & 0 & \infty \\ \infty & \infty & \infty \end{bmatrix}.$$

Podobnie dla stanu symbolicznego s_6, którego elementami jest strefa Z_6, $X_6 = \{x_0, x_4, x_5\}$ oraz takie samo znakowanie, spowodowanego pobraniem zadania do wykonania do węzła (t_1), wykonania zadania (t_2), odcięcia (t_3):

$$Z_6 = \begin{bmatrix} 0 & 0 & -7 \\ 2 & 0 & -7 \\ 17 & 15 & 0 \end{bmatrix}; Z'_6 = LU_x\text{-}Approx(Z_6) \begin{bmatrix} 0 & 0 & \infty \\ 2 & 0 & \infty \\ \infty & \infty & \infty \end{bmatrix}$$

Otrzymana równość $Z'_{10} = Z'_6$ identyfikuje stan symboliczny s_{10} ponawiający informację o zachowaniu zapisaną w s_6.□

W takim przypadku konieczna jest modyfikacja warunku stopu algorytmu nr 1. Nowy warunek stopu polega na modyfikacji testu $s \notin Pass$. Wynik testu jest pozytywny, nie tylko gdy w Pass znajduje się węzeł z taką samą strefą co $s.Z$, ale również gdy aproksymacja jego strefy jest taka sama.

Tak więc w przykładzie 7 warunek $s_{10} \notin Pass$ zachodzi, ponieważ strefa $s_{10}.Z_{10}$ ma akroksymację ze strefą $s_6.Z_6$ w węźle s_6. Zatem węzeł s_{10} nie zostanie dodany do zbioru $Pass$, jedynie może widnieć jako liść na grafie stref - drzewie stref.

3.2 Weryfikacja modelowa własności temporalnych zachowania

W tym rozdziale zostanie przedstawiona znana metoda weryfikacji własności systemu opisanych formułami logiki TCTL. Zostanie przybliżone pojęcie logiki TCTL zdefiniowanej na systemie TTS. Syntaktyka logiki TCTL zostanie zdefiniowana dla systemu TTS.

3.2.1 Syntaktyka logiki TCTL dla TPN

Logika TCTL do reprezentacji własności modelu TPN zostanie zdefiniowana z wykorzystaniem formuł GMEC (ang. General Mutual Exclusion Constraint [25]) orzekających czy w danym stanie, w danym znakowaniu, system sieciowy ma odpowiednią liczbę żetonów w odpowiednich miejscach. Formuła GMEC jest definiowana indukcyjnie:

$$\gamma := \sum_{i=1}^{n} a_i \cdot m(p_i) \sim c | \gamma \vee \gamma | \gamma \wedge \gamma | \gamma \Rightarrow \gamma$$

gdzie a_i to liczba całkowita, $m(p_i)$ reprezentuje liczbę żetonów w miejscu $p_i \in P$, n to liczba wszystkich miejsc sieci, $\sim \in \{<, \leqslant, =, >, \geqslant\}$ oraz c jest liczbą naturalną.

Spełnienie formuły γ w znakowaniu oznaczymy symbolem $m \models \gamma$.

Syntaktyka logiki TCTL dla modelu TPN (akr. TPN-TCTL), jest definiowana indukcyjnie:

$$\varphi := \textbf{false} | \gamma | \neg\varphi | \varphi \Rightarrow \varphi | E\varphi U_I \varphi | A\varphi U_I \varphi$$

gdzie γ to formuła GMEC, I to podzbiór półosi dodatniej, **false** to stała, A, E są operatorami ścieżkowymi odpowiednio „dla wszystkich ścieżek" i „istnieje ścieżka", U_I to operator temporalny „dopóki" zależny od przedziału I.

Ponadto z powyższej gramatyki można wyprowadzić: **true** := $\neg$**false**, $EF_I\varphi = E\textbf{true}U_I\varphi$, $AF_I\varphi = A\textbf{true}U_I\varphi$, $EG_I\varphi = \neg AF_I\neg\varphi$,$AG_I\varphi = \neg EF_I\neg\varphi$. Symbole G_I, F_I to odpowiednio operator konieczności i możliwości, oba zależne od przedziału I.

3.2.2 Semantyka logiki TPN-TCTL

Semantyka dla logiki TPN-TCTL została określona na czasowym systemie tranzycyjnym S_M. Relacja spełnialności $\models$ w stanie $q = (m, v) \in Q$ jest definiowana jako:

$$
\begin{aligned}
q \models \gamma \quad & wtw \quad m \models \gamma, \gamma \in GMEC \\
\neg(q \models false) & \\
q \models \neg\varphi \quad & wtw \quad \neg(q \models \varphi) \\
q \models (\varphi \Rightarrow \psi) \quad & wtw \quad \neg(q \models \varphi) \vee q \models \psi \\
q \models E\varphi U_I \psi \quad & wtw \quad \exists \rho = q_1 \xrightarrow{d_1} q_1 + d_1 \xrightarrow{a_1} \cdots \in \pi(q), \exists_{i>0}, \exists \delta \in [0, d_i]: \\
& \quad (\sum_{j=1}^{i-1} d_j) + \delta \in I \wedge q_i + \delta \models \psi \wedge \\
& \quad (\forall_{\delta' \in [0,\delta)}, q_i + \delta' \models \varphi) \wedge \\
& \quad (\forall_{j:0<j<i} \forall_{\delta' \in [0,d_j]}, q_j + \delta' \models \varphi)
\end{aligned}
$$

$$
\begin{aligned}
q \models A\varphi U_I \psi \quad & wtw \quad \forall \rho = q_1 \xrightarrow{d_1} q_1 + d_1 \xrightarrow{a_1} \cdots \in \pi(q), \exists_{i>0}, \exists \delta \in [0, d_i]: \\
& \quad (\sum_{j=1}^{i-1} d_j) + \delta \in I \wedge q_i + \delta \models \psi \wedge \\
& \quad (\forall_{\delta' \in [0,\delta)}, q_i + \delta' \models \varphi) \wedge \\
& \quad (\forall_{j:0<j<i} \forall_{\delta' \in [0,d_j]}, q_j + \delta' \models \varphi)
\end{aligned}
$$

Model systemu M ma własność φ opisaną formułą TPN-TCTL, co zapiszemy symbolicznie $M \models \varphi$, wtw gdy w stanie początkowym (m_0, v_0) jego zachowania S_M jest spełniona formuła φ ($(m_0, v_0) \models \varphi$).

Formuły EF_I, AF_I, EG_I, AG_I można analogicznie zdefiniować przy wykorzystaniu operatora U_I.

3.2.3 Weryfikacja własności systemów TPN w praktyce

Opisane własności w [9] jakie mogą być weryfikowane automatycznie dla modeli TPN dzięki dyskretyzacji zostały zdefiniowane jako podzbiór *TPN-TCTL*$_S$ logiki *TPN-TCTL*:

$$TPN\text{-}TCTL_S := E\varphi U_I \psi | A\varphi U_I \psi | EF_I \varphi | AF_I \varphi | EG_I \varphi | AG_I \varphi | \varphi \rightsquigarrow_{I_l} \psi$$

gdzie $\varphi, \psi \in GMEC$, $(\varphi \rightsquigarrow_{I_l} \psi) = AG(\varphi \Rightarrow AF_{I_l}\psi)$, I i I_l są spójnymi podzbiorami dodatniej półosi przy czym kres dolny I_l jest zawsze równy 0. *TPN-TCTL*$_S$ nie zawiera formuł zagnieżdżonych.

Gdy dana własność systemu jest spełniona oznacza:

$E\varphi U_I \psi$ – jest taki przebieg systemu w którym od samego początku φ musi być spełnione do momentu, aż ψ stanie się prawdziwe dokładnie w globalnym przedziale I,

$A\varphi U_I\psi$ – w każdym przebiegu systemu od samego początku φ musi być spełnione do momentu, aż ψ stanie się prawdziwe, dokładnie w globalnym przedziale I,

$EF_I\varphi$ – jest taki przebieg systemu, że w pewnym momencie przedziału globalnego I, φ stanie się prawdziwe,

$AF_I\varphi$ – w każdym przebiegu systemu, w pewnym momencie przedziału globalnego I φ stanie się prawdziwe,

$EG_I\varphi$ – jest taki przebieg systemu, że w każdym momencie przedziału globalnego I, φ jest prawdziwe,

$AG_I\varphi$ – w każdym przebiegu systemu, w każdym momencie przedziału globalnego I, φ jest prawdziwe,

$\varphi \rightsquigarrow_{I_l} \psi$ – w każdym przebiegu systemu, od samego początku zawsze, jeżeli φ stanie się prawdziwe to we wszystkich przypadkach ψ musi stać się prawdziwe nie później niż $\sup I_l$ od tego momentu.

Ze wszystkich powyższych wyróżniającą się własnością do weryfikowania jest ograniczony czas reakcji (z ang. *bounded response*) $\varphi \rightsquigarrow_{I_l} \psi$. Przy założeniu, że $c_l = \sup I_l$, własność ta oznacza: jeżeli φ stanie się prawdziwe to zawsze w przyszłosci ψ będzie prawdziwe nie później niż do c_l jednostek czasu.

Niech b oznacza pomocniczą zmienną logiczną, której wartość *true* odpowiada potrzebie spełnienia warunku: formuła ψ stanie się prawdziwa najpóźniej do c_l jednostek czasu.

Własność $\varphi \rightsquigarrow_{I_l} \psi$ może być przekształcona wtedy do postaci:

$$AG_{[0,\infty)}(b \Rightarrow z \leqslant c_l)$$

gdzie z to wskazanie dodatkowego zegara odmierzającego czas od momentu spełnienia b.

Każde ustawienie $b = true$ jest odnotowane i sprawdzane czy nie ma takiej sytuacji, aby dodatkowy zegar z przekroczył ograniczenie c_l.

W pracy [9] przedstawiono implementację weryfikacji własności $\varphi \rightsquigarrow_{I_l} \psi$, wykorzystującą powyższy fakt, która generuje na bieżąco potrzebny do weryfikacji fragment przestrzeni stanów.

Ponadto dla uproszczenia w implementacji można badać zaprzeczenie formuły $EF_{[0,\infty)}(\neg b \wedge z > c_l)$.

Weryfikacja własności $AG_{[0,\infty)}(\varphi \Rightarrow AF_{[0,c_l]}\psi)$ polega na wywołaniu funkcji $CheckBoundedResponse(s_0, \varphi, \psi, c)$ dla $c = c_l$ natomiast funkcja pomocnicza $CheckBoundedResponse\_aux(s_0, \varphi, \psi, c)$ bada prawdziwość zaprzeczenia własności.

Funkcja 2: Pseudokod weryfikacji własnosci $\varphi \rightsquigarrow_{I_l} \psi$

CheckBoundedResponse(s_0, φ, ψ, c):
$Visit := \emptyset$
$s_0 = (m_0, v_0)$
return $\neg$*CheckBoundedResponse_aux*(s_0, φ, ψ, c)

Funkcja 3: Pseudokod pomocniczy weryfikacji własnosci $\varphi \rightsquigarrow_{I_l} \psi$

CheckBoundedResponse_aux(s_0, φ, ψ, c):
$(M, Z, b) := s$
if $Z \cap (c, \infty) \neq \emptyset$ **then**
  return True
if $s \models \psi$ **then**
  $b := False$
else
  if $\neg b \wedge s \models \phi$ **then**
    $b := True$
    $Z := Z[z \leftarrow 0]$
$Z := \overrightarrow{Post}(Z)$
$Visit := Visit \cup \{(m, Z, b)\}$ **for** $t \in enabled(s)$ **do**
  $s' := Post_t(m, Z)$
  if $s \notin Visit$ **then**
    if *CheckBoundedResponse_aux*(s', φ, ψ, c) **then**
      return True
return False

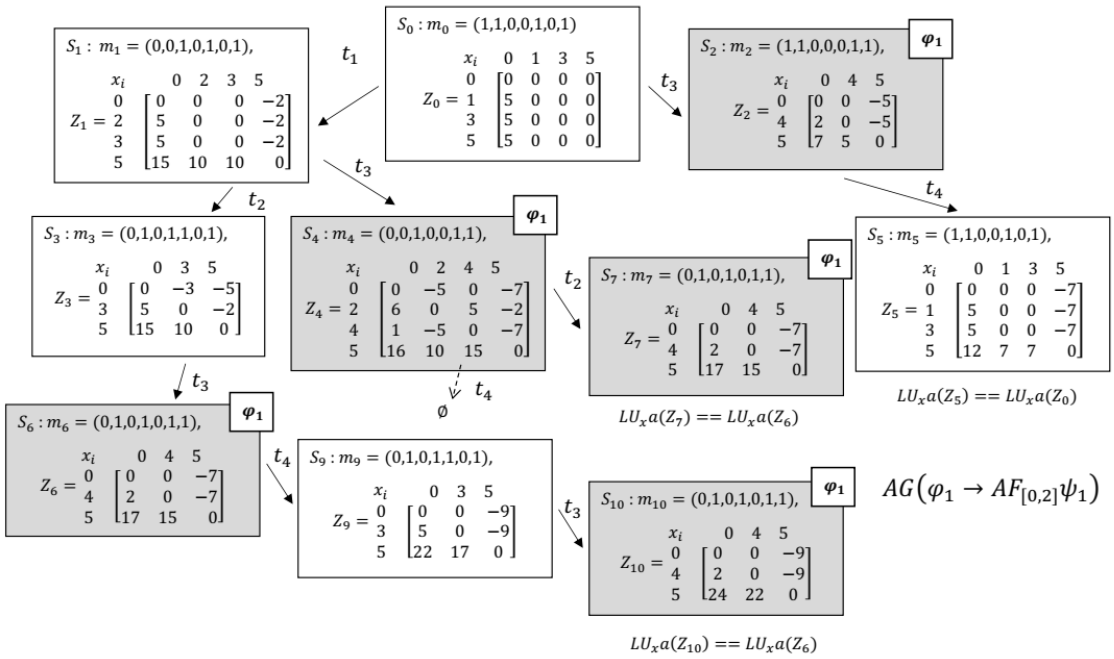
Przykład 8 *Rozważono następującą własność typu ograniczonej reakcji:*

> *W każdym przypadku jeżeli zasoby zostaną odcięte to po 2 jednostkach czasu (np. sekundach) zostaną zawsze przywrócone.*

Własność ta jest reprezentowana przez formułę:

$$AG(\varphi_1 \rightarrow AF_{[0,2]}\psi_1),$$

gdzie $\varphi_1 : m(P_{nres}) = 1 \wedge m(P_{res}) = 0$ *oraz* $\psi_1 : m(P_{nres}) = 0 \wedge m(P_{res}) = 1$.



Rysunek 13: Wyróżnienie stanów symbolicznych spełniających formułę φ_1 (pozostałe stany spełniają ψ_1).

Jak widać w każdym stanie symbolicznym, w którym odcięto zasoby (szary kolor) zawsze po upłynięciu co najwyżej 2 jednostek czasu system przejdzie do stanu, w którym zasoby będą przywrócone do czytania.

Jest to jedna z wielu własności TPN-TCTL, które można zbadać dla systemu TPN przy wykorzystaniu fragmentu reprezentatywnego.

4 Proponowane rozwiązanie - reprezentatywny fragment

W niniejszym rozdziale przedstawiono algorytmy konstruowania reprezentatywnego fragmentu zachowania modeli c-TdPN oraz sposoby weryfikacji własności modelowanych systemów.

Zarówno rozgałęziony proces czasowy jak i czasowy system tranzycyjny reprezentujący semantykę c-TdPN może posłużyć jako reprezentatywny fragment zachowania będący następnie podstawą weryfikowania własności modelowanego systemu.

Rozdzał ma złożoną strukturę. Rozpoczyna się od krótkiego sformułowania powiązania pomiędzy modelami RPC i TTS.

Następnie opisane są dwa algorytmy konstruowania reprezentatywnego fragmentu zachowania:

- algorytm konstruowania fragmentu rozgałęzionego procesu czasowego (RPC),
- algorytm konstruowania fragmentu grafu stref (przez dyskretyzację) (TTS).

W dalszej części opisano weryfikowane własności.

4.1 Powiązania pomiędzy modelami zachowania RPC i TTS

Stany systemu tranzycyjnego TTS używanego do definiowania semantyki logik temporalnych odpowiadają konfiguracjom rozgałęziającego się procesu czasowego reprezentującego zachowanie się systemu, to znaczy takim częściom tego procesu, które:

- nie zawierają konfliktów,
- wraz z każdym swoim elementem zawierają elementy od których on (słabo) przyczynowo zależy.

Niech będzie dany model c-TdPN $N = (P, T, F, C, I, m)$. Niech $U(N) = (B, E, G, H, l)$ wraz z funkcją ograniczeń θ oznaczają rozgałęziony proces czasowy modelu N. Niech $S_N = (Q, Q_0, \Sigma, \longrightarrow)$ będzie czasowym systemem tranzycyjnym (TTS) reprezentującym zachowanie modelu N, a $SA_N = (QA, QA_0, \Pi, \Longrightarrow)$ jego dyskretyzacją (grafem stref). Zakładamy, że N jest 1-bezpieczna.

Niech będzie dany przebieg czasowy (E, θ) w $U(N)$, oraz niech będzie dana odpowiadająca mu ścieżka $\rho = q_0 \xrightarrow{d_0} q_0 + d_0 \xrightarrow{a_1} q_1 \xrightarrow{d_1} q_1 + d_1 \xrightarrow{a_2} \ldots$ w S_N.

Niech dalej $M = cut((E, \theta))$ oznacza przekrój przebiegu (E, θ). Wtedy dla pewnego momentu globalnego d oraz odpowiadającego mu przekroju M możemy wyznaczyć czas przebywania żetonów od ich momentu pojawienia się w miejscach M na podstawie momentów wykonań zdarzeń.

W ścieżce ρ możemy zatem wyznaczyć taki stan $q = (v, m)$, w którym $m = l(M)$, a funkcja v okeśla długość przebywania żetonów występujących w miejscach m do momentu d.

Związek ten pozwala na dostrzeżenie podobieństw pomiędzy dwoma, kolejno opisanymi w dalszej części reprezentatywnymi fragmentami zachowania się systemu.

4.2 Algorytm konstruowania reprezentatywnego rozgałęzionego procesu czasowego (modelu RPC)

Niech termin prefiks zupełny oznacza reprezentatywny fragment rozgałęzionego procesu czasowego.

Niech $UNF(N) = (B, E, G, H, l)$ wraz z funkcją ograniczeń θ oznaczają prefiks rozgałęzionego procesu czasowego modelu c-TdPN reprezentowanego przez $N = (P, T, F, C, I, m)$.

Prezbiory, postzbiory, relacja słabej przyczynowości $\nearrow$ oraz konfliktu bezpośredniego $\#$ dla N i $UNF(N)$ są definiowane tak jak dla modelu PSwPN w podrozdziale 2.1.

Funkcja definiująca momenty wykonania zdarzeń w $UNF(N)$ wymaga dostosowania do charakteru modelu c-TdPN.

Funkcja θ momentu wykonania zdarzeń E z $UNF(N)$ ma własności:

(1) moment inicjujący stan początkowy m jest równy zero ($\theta(\perp) = 0$),

(2) każdy cykl w E zawiera zdarzenie niemożliwe (o momencie ∞), oraz

(3) dla każdego zdarzenia $e \in E - \{\perp\}$ spełnione są warunki:

1. $\theta(e) \neq \infty \wedge \theta(e) \geqslant \max(\{\theta(Gb) : b \in Ge \cup He\}) \wedge$
   $\wedge \forall_{b \in Ge \cup He} \min I(l(b), l(e)) \leqslant \theta(e) - \theta(Gb) \leqslant \max I(l(b), l(e)) \wedge$
   $\wedge \forall_{\ e' \in E \ \wedge \ e' \text{ conf } e} \theta(e') = \infty \wedge$
   $\wedge \forall_{\ e' \in E \ \wedge \ e' \nearrow e} \theta(e') \leqslant \theta(e)$ lub
2. $\theta(e) = \infty \wedge \exists_{b \in Ge \cup He} \theta(Gb) = \infty$ lub
3. $\theta(e) = \infty \wedge$
   $\wedge \exists_{e' \in E}[(e \text{ conf } e' \vee e \nearrow e') \Rightarrow (\theta(e') < \theta(e))]$ lub
4. $\theta(e) = \infty$.

Zmianą w stosunku do funkcji momentów zdarzeń z PSwPN jest punkt (3), w którym nie rozważa się długości czynności tranzycji modelu, a długość przebywania żetonów w miejscach ((3).1). Rrezygnuje się także z opisu pilnego wykonania tranzycji, którego brak w modelu c-TdPN, co powoduje dodanie punktu (3).4.

Przykład 9 *Niech $E_1 = \{\perp, e3, e6, e7\}$ i θ_1 oznaczają początkowy fragment przebiegu czasowego zawartego we fragmencie rozgałęzionego procesu czasowego z rysunku 2. Framgent ten zawiera kolejno odcięcie zasobów ($e3$), przywrócenie zasobów ($e6$) oraz rozpoczęcie wykonania zadania w węźle drugim ($e7$). Zdarzenie $\perp$ inicjuje przebieg czasowy. Od momentu odcięcia zasobów $\theta_1(e3) = 5$ zablokowane są wystąpienia $e1$ i $e2$ w E_1. Zdarzenia te mają przypisane momenty $\theta_1(e1) = \theta_1(e2) = \infty$ ponieważ w kontekście rozgałęzionego procesu czasowego powinny wystąpić wcześniej niż $e3$ (zgodnie z (3).3). Fakt ten można zapisać relacją słabej przyczynowości $e3 \nearrow e1$ i $e3 \nearrow e2$. Zdarzenia z momentem ∞ nie są dodawane co zbioru E_1.*

Zdarzenia $e4$ i $e5$ mają przypisany moment ∞ ponieważ zdarzenia je poprzedzające $e1$ i $e2$ nie wystąpiły w E_1 (zgodnie z (3).2).

W momencie $\theta(e6) = 7$ przywrócono zasoby oraz natychmiast rozpoczęto w momencie $\theta(e7) = 7$ wykonywać zadanie. Każde ze zdarzeń $e3$,$e6$ i $e7$ spełniają punkt (3).1.

Przedstawiony w przykładzie porzątkowy fragment trwał do momentu 7.□

Prefiks rozgałęzionego procesu czasowego może być rozbudowany o zdarzenie rozszerzające e jeżeli:

1. $e \notin E$,
2. $l(e) \in T$,

3. $Ge \cup He \subseteq B$,

4. moment $\theta(e)$ jest nie wcześniejszy niż moment dostępności kompletu elementów z $Ge \cup He$,

5. dla każdego $b \in Ge \cup He$ różnica momentów $\theta(e) - \theta(Gb)$ jest nie krótsza niż $\min\{I(l(b), l(e))\}$ oraz nie dłuższa niż $\max\{I(l(b), l(e))\}$.

Teoretycznie rozbudowywanie $UNF(N)$ poprzez dodanie zdarzenia rozszerzającego prowadzi do największego (prefiksu) rozgałęznionego procesu czasowego $U(N)$ nazywanego zupełnym rozwinięciem (prefiksem zupełnym).

Definicja 6 *$(UNF_{\theta<\infty}(N), \theta)$ oznacza prefiks czasowy wybranego przebiegu (E, θ) zawartego w $UNF(N)$.*

Symbol $\theta < \infty$ nawiązuje do wykluczenia z E zdarzeń o momencie ∞.

Definicja 7 *Prefiksy rozgałęzionych procesów czasowych $UNF(N)$ i $UNF'(N)$ mają równoważną przyczynową przyszłość jeżeli dla wszystkich odpowiadających sobie (jeden jest prefiksem drugiego) przebiegów $UNF_{\theta<\infty}(N)$ i $UNF'_{\theta'<\infty}(N)$ zachodzi:*

1. $l(cut(UNF_{\theta<\infty}(N))) = l(cut(UNF'_{\theta'<\infty}(N)))$,

2. $\forall_{b \in cut(UNF_{\theta<\infty}(N))} \forall_{b' \in cut(UNF'_{\theta'<\infty}(N))}$
   $l(b) = l(b') \Rightarrow age(b, \theta, cut(UNF_{\theta<\infty}(N))) = age(b', \theta', cut(UNF'_{\theta'<\infty}(N)))$.

przy czym wiek warunków jest obliczany wzorem:

$$age(b, \theta, A) = \min\{\max_{b' \in A}\{\theta(Gb')\} - \theta(Gb), \max\{K(t) : t \in T \wedge t \in l(b)(F \cup C)\}\}$$

gdzie

- $K(t) = \max_{p \in Ft}\{L(I(p,t))\}$,
- $L(I(p,t)) = \begin{cases} \max\{I(p,t))\} & dla \quad \max\{I(p,t))\} \neq \infty \\ \min\{I(p,t))\} & dla \quad \max\{I(p,t))\} = \infty \end{cases}$

□

Wiek warunku $age(b, \theta, A)$ jest definiowany analogicznie jak dla RPC reprezentującego zachowanie modelu TPN (podrozdział 2.4.1). Wyjątkiem jest zmiana we wzorze $K(t)$ polegająca na rozważaniu przedziałów dostępności dla łuków wchodzących do tranzycji t. Ponadto $K(t)$ jest wyznaczane także dla tranzycji powiązanych relacją C z miejscem $l(b)$.

Zbiór $Past(e)$ dla zdarzenia e jest definiowany analogicznie jak dla modeli PSwPN. Niech dla $e \in E$ elementami zbioru $Past(e)$ będą RPC $(UNF_i(N), \theta_i)$, dla $i = 1, 2, \ldots, n$ o tej własności, że:

1. Każdy $(UNF_i(N), \theta_i)$ zawiera wybraną historię zdarzenia e.

2. Każdy $(UNF_i(N), \theta_i)$ jest możliwie najmniejszy pod względem liczby zdarzeń, ale nie mniejszy niż wybrana historia zdarzenia e.

3. zdarzenie e ma n historii.

Definicja 8 *Rozszerzenie e_2 jest zdarzeniem odcięcia w prefiksie $(UNF(N), \theta)$ jeżeli każdy $(UNF_2(N)) \in Past(e_2)$ zawiera inne (wcześniejsze) zdarzenie e_1, oraz istnieje taki $(UNF_1(N)) \in Past(e_1)$, że:*

- $\lceil e_1 \rceil \prec \lceil e_2 \rceil$*, są w relacji adekwatnego porządku,*
- $l(cut(\lceil e_1 \rceil)) = l(cut(\lceil e_2 \rceil))$*, powodują wystąpienie żetonów w tych samych miejscach,*
- $(UNF_1(N), \theta_1)$ *i* $(UNF_2(N), \theta_2)$ *mają równoważną przyczynową przyszłość.*$\square$

Konstruowanie reprezentatywnego rozgałęzionego procesu dla modeli typu c-TdPN odbywa się podobnie jak dla modeli PSwPN. Poczynając od stanu początkowego modelu systemu dodawane są kolejno informacje o zdarzeniach wykonania umożliwianych tranzycji oraz identyfikowane są zdarzenia odcięcia powtarzające takie informacje.

Konstrukcja prefiksu rozwinięcia modeli c-TdPN przebiega według opracowanego algorytmu nr 4. Algorytm jako dane wejściowe otrzymuje strukturę reprezentującą zainicjowany model systemu c-TdPN $N = (P, T, F, C, I, M_0)$ oraz regułę wyboru zdarzenia rozszerzającego.

Wynikiem działania algorytmu jest początkowy fragment rozgałęzionego procesu czasowego reprezentowany przez:

- zbiór UNF przechowujący informację o reprezentatywnym fragmencie zachowania oraz
- zbiór R ograniczeń momentów wykonań zdarzeń zapisanych w UNF.

Zbiór UNF przechowuje dwa typy elementów:

- (b, e) oznacza wyprodukowany warunek b przez zdarzenie e, oraz
- $e = (t, A, B)$ oznacza zdarzenie e wystąpienia tranzycji t, które zużywa warunki A oraz czyta warunki B.

Elementami zbioru R są zbiory $R(e)$. Każdy zbiór $R(e)$ przechowuje formuły logiczne ograniczające moment wystąpienia zdarzenia e.

Konstruowany prefiks rozgałęzionego procesu, reprezentowany przez zbiór UNF, w stanie początkowym zawiera warunki odpowiadające znakowaniu początkowemu oraz zdarzenie $\perp$ je powodujące. Konstruowany zbiór ograniczeń reprezentowany przez R zawiera początkowo informację o momencie wykonania zdarzenia $\perp$. Zbiór PE zawiera zdarzenia zwane *potencjalnymi rozszerzeniami* dla UNF spełniającymi 1-3 warunki

dla zdarzenia rozszerzającego. Funkcja pe wyznacza dla UNF potencjalne zdarzenia rozszerzające.

Algorytm cyklicznie wybiera zgodnie z założeniem wyboru zdarzenie rozszerzające e i dodaje je do UNF, nakłada ograniczenia na moment jego wykonania, a także w razie konieczności dodatkowe ograniczenia dla zdarzeń znajdujących się wcześniej w prefiksie rozgałęzionego procesu (w zbiorze UNF). Ponadto po każdym zabiegu rozszerzenia prefiksu zbiór UNF jest uzupełniany o nowo wyprodukowane warunki oraz odbywa się aktualizacja zbioru potencjalnych zdarzeń rozszerzających.

Algorytm 4: Pseudokod konstrukcji prefiksu zupełnego rozwinięcia modelu c-TdPN.

Dane:

1. $N = (P, T, F, C, I, M_0)$ model c-TdPN,
2. reguła wyboru zdarzenia rozszerzającego ze zbioru PE,
3. UNF - zawiera zdarzenie inicjujące $\perp$ oraz elementy postaci $(b, \perp)$.
4. $R = \{R(\perp)\}$, gdzie $R(\perp) = \{\theta(\perp) = 0\}$,
5. $PE = pe(UNF)$.

Rezultat: Prefiks zupełny reprezentowany przez parę UNF i R
while $PE \neq \emptyset$ **do**

1. Wybierz zdarzenie e uruchomienia tranzycji $t = l(e)$ ze zbioru PE zgodnie z przyjętą regułą. Następnie przypisz: $PE = PE - \{e\}$.
2. Określ dla zdarzenia e ograniczenia momentu wykonania związane z łukami wejściowymi do t:
   - 2.1. Zainicjowanie zbioru $R(e) = \emptyset$.
   - 2.2. Dla każdego $p \in F\{l(e)\}$, gdzie $l(b) = p$ i $\delta = \theta([Gb]_{UNF})$: $R(e) = R(e) \cup \{\min I(l(b), l(e)) + \delta \leqslant \theta(e) \leqslant \max I(l(b), l(e)) + \delta\}$.
   - 2.3. Gdy koniunkcja formuł z $R(e)$ powoduje brak rozwiązania względem $\theta(e)$, usuń $R(e)$ z R i przejdź do kolejnego wykonania pętli.

3 Rozszerz zbiór UNF:

- 3.1. $UNF = UNF \cup \{e\}$.
- 3.2. Gdy e jest zdarzeniem odcięcia, przejdź do kolejnego wykonania pętli.
- 3.3. Dla każdego $p \in l(e)F$, takiego, że $p = l(b)$: $UNF = UNF \cup \{(b, e)\}$.

---

4. Sprawdź konieczność uzupełnienia ograniczeń w R:

   4.1. związanych z relacją słabej przyczynowości:

      4.1.1 Dla każdego $e' \in [GGe]_{UNF}$: $R(e) = R(e) \cup \{\theta(e') \leqslant \theta(e)\}$

      4.1.2 Dla każdego $e' \in [(Ge)H]_{UNF}$:
      $R(e') = R(e') \cup \{\theta(e') \leqslant \theta(e)\}$

      4.1.3 Dla każdego $e' \in [(He)G]_{UNF}$:
      $R(e) = R(e) \cup \{\theta(e) \leqslant \theta(e')\}$

   4.2 związanych z konfliktem:

      4.2.1 Dla każdego $e' \in [(Ge)]_{UNF} G \wedge e' \neq e$:
      $R(e) = R(e) \cup \{\theta(e') = \infty\}$
      i $R(e') = R(e') \cup \{\theta(e) = \infty\}$

5. Aktualizacja $PE = pe(UNF)$.

Znaczenie symboli $[Gb]_{UNF}$, $[GGe]_{UNF}$, $[(Ge)H]_{UNF}$, $[(He)G]_{UNF}$ i $[(Ge)G]_{UNF}$ w kontekście zbioru UNF jest analogiczne co znaczenie Gb, GGe, $(Ge)H$, $(He)G$ i $(Ge)G$ w kontekście rozgałęzionego procesu czasowego:

$$[Gb]_{UNF} := e \text{ gdy } (b, e) \in UNF,$$

$$[GGe]_{UNF} := \{e' : \exists_{b \in A} e = (t, A, B) \wedge (b, e') \in UNF \wedge e \in UNF\},$$

$$[(Ge)H]_{UNF} := \{e' : \exists_{b \in A \cap B'} e' = (t', A', B') \wedge e = (t, A, B) \wedge e, e' \in UNF\},$$

$$[(He)G]_{UNF} := \{e' : \exists_{b \in A' \cap B} e' = (t', A', B') \wedge e = (t, A, B) \wedge e, e' \in UNF\},$$

$$[(Ge)G]_{UNF} := \{e' : \exists_{b \in A' \cap A} e' = (t', A', B') \wedge e = (t, A, B) \wedge e, e' \in UNF\}.$$

Przykładowo działanie algorytmu można prześledzić analizując rysunek 2. Niech UNF i R będzie w stanie początkowym. Zbiór UNF przechowuje zdarzenie inicjujące system w momencie $\theta(\perp) = 0$ reprezentowane przez $\perp$, zadanie do realizacja $(b1, \perp)$, dwa wolne węzły obliczeniowe $(b0, \perp)$ i $(b2, \perp)$ oraz zasoby współdzielone $(b3, \perp)$. $R = \{R(\perp)\}$ i $R(\perp) = \{\theta(\perp) = 0\}$. Zbiór PE zawiera trzy potencjalne zdarzenia rozszerzenia: pobranie zadania do węzeła 1, do węzeła 2 lub odcięcie zasobów współdzielonych. Zdarzenia są reprezentowane kolejno przez $e1$, $e2$ i $e3$.

Następnie niech UNF i R będzie w stanie tuż po dodaniu do konstruowanego prefiksu zdarzeń $e1$ i $e2$. Na tym etapie zbiór prefiksu rozwinięcia stanowi:
$UNF = \{\perp, (b0, \perp), (b1, \perp), (b2, \perp), (b3, \perp),$
$e1, (b4, e1), e2, (b5, e2)\}$,
przy czym $e1 = (T_{11}, \{b0, b1\}, \{b3\})$, $e2 = (T_{21}, \{b1, b2\}, \{b3\})$,
zbiór ograniczeń $R = \{R(\perp), R(e1), R(e2)\}$, gdzie:
$R(\perp) = \{\theta(\perp) = 0\}$,

$R(e1) = \{\theta(\perp) + 2 \leqslant \theta(e1) \leqslant \theta(\perp) + 10, \theta(e2) = \infty\}$,
$R(e2) = \{\theta(\perp) + 4 \leqslant \theta(e2) \leqslant \theta(\perp) + 12, \theta(e1) = \infty\}$
oraz zbiór $PE = \{e3, e4, e5\}$ potencjalnych zdarzeń rozszerzających UNF.

Niech następnie wybranym zdarzeniem rozszerzającym dla powyższego UNF będzie odcięcie zasobów reprezentowane przez $e3$. Stąd $R = R \cup \{R(e3)\}$, gdzie $R(e3) = \{\theta(\perp) + 5 \leqslant \theta(e3) \leqslant \theta(\perp) + 6\}$, $UNF = UNF \cup \{e3, (b6, e3)\}$, $e3 = (T_{r1}, \{b3\}, \emptyset)$ oraz uzupełnienie w $R(e1)$ i $R(e2)$ związane z relacją słabej przyczynowości: $R(e1) = R(e1) \cup \{\theta(e1) \leqslant \theta(e3)\}$ i $R(e2) = R(e2) \cup \{\theta(e2) \leqslant \theta(e3)\}$.

Kolejny obszerniejszy przykład wyznaczania reprezentatywnego fragmentu zachowania przy pomocy algorytmu nr 4 znajduje się w załączniku B.

Twierdzenie 1 *Niech będzie dany model systemu N typu c-TdPN oraz rozgałęziony proces czasowy $(U(N), \theta)$ reprezentujący jego zachowanie. Algorytm nr 4 generuje parę (UNF_N, R) reprezentującą prefiks zachowania $(U(N), \theta)$ modelu systemu N, który jest zarazem prefiksem zupełnym.*

Dowód 1 *W celu pokazania, że (UNF_N, R) jest prefiksem zupełnym należy podkreślić, że zawiera zdarzenia reprezentujące wykonanie wszystkich umożliwionych tranzycji zapisanych w $U(N)$.*

Ponieważ N jest modelem systemu typu c-TdPN, a para $(U(N), \theta)$ modelem jego zachowania (rozgałęzionym procesem czasowym) to $N = (P, T, F, C, I, m_0)$, $U(N) = (B, E, G, H, l)$ oraz θ jest funkcją określającą momenty zdarzeń w E. Ponadto w miejscach modelu N może pojawić się co najwyżej 1 żeton.

Para (UNF_N, R) reprezentuje prefiks zachowania modelu systemu N wygenerowany przez algorytm nr 4. Zbiór R zawiera ograniczenia dla funkcji momentu wykonania θ_R zdarzeń zapisanych w UNF_N. Zbiór UNF_N zawiera elementy dwóch rodzajów:

- *$e = (t, A, B)$ reprezentujący zdarzenie e wykonania tranzycji t, które zużywa żetony z miejsc $l(A)$ i czyta żetony z miejsc $l(B)$,*
- *(b, e), gdzie b jest warunkiem spowodowanym przez zdarzenie e i oznacza wyprodukowanie żetonu w miejscu $l(b)$.*

Następnie zostanie pokazane, że (UNF_N, R) zawiera zdarzenia reprezentujące wykonanie wszystkich umożliwionych tranzycji modelu systemu N od momentu jego zainicjowania.

Zatem niech $t_a \in T$ oznacza dowolną tranzycję, która stała się umożliwiona i została wykonana po pewnym czasie od momentu zainicjowania ($\theta(\perp) = 0$) systemu reprezentowanego przez model N.

Ponieważ miało miejsce wykonanie tranzycji t_a to istnieje zdarzenie $e_a \in E$, takie że $\theta(e_a) \neq \infty$, $e_a = (t_a, Ge_a, He_a)$, $t_a = l(e_a)$, $Ge_a \subseteq B$, $He_a \subseteq B$, $l(Ge_a) \subseteq Ft_a$, $l(He_a) \subseteq Ct_a$. Ponadto istnieje także przebieg czasowy reprezentowany przez (E_a, θ_a), że $Ge_a \cup He_a \subseteq cut(E_a - \{e_a\})$, $Ge_a \cup He_a$ jest co-zbiorem w E_a, $\perp \in E_a$, $\theta_a(\perp) = 0$ oraz $E_a - \{e_a\}$ jest historią zdarzenia e_a.

Przebieg czasowy (E_a, θ_a) jest zawarty w $U(N)$. Założono, że (E_a, θ_a) nie zawiera powtórnego wykonania tranzycji t_a. Niech zbiór zdarzeń E_a ma postać $E_a = \{e_0, e_1, \ldots, e_n\}$ przy czym $e_0 :=\perp$ i $e_n := e_a$.

Stosując technikę dowodu nie wprost, założono że odpowiednik (EN, R_{EN}) przebiegu czasowego (E_a, θ_a) nie jest przechowywany w prefiksie (UNF_N, R).

Dane są dwa ciągi wstępujące o wyrazach $(EU_{(i)}, \tau_{(i)})$ i $(EN_{(i)}, R_{(i)})$ dla $i = 0, 1, 2, \ldots, m$. Ciąg $\{(EU_{(i)}, \tau_{(i)})\}$ jest ciągiem prefiksów przebiegu czasowego (E_a, θ_a) i ma własności:

- $EU_{(0)} = \{\perp\}$,
- $EU_{(i)} = \{e_0, e_1, \ldots, e_i\}$, *dla* $i = 0, 1, \ldots, n$,
- $EU_{(n)} = E_a$,
- $EU_{(i)} \subseteq EU_{(i+1)} \subseteq E_a$, *dla* $i < n$,
- *funkcja $\tau_{(i)}$ powstała przez ograniczenie dziedziny funkcji θ_a do zbioru $EU_{(i)}$,*
- $\tau_{(n)} := \theta_a$.

Ciąg $\{(EN_{(i)}, R_{(i)})\}$ jest ciągiem prefiksów przebiegu reprezentowanego przez (EN, R_{EN}) i ma własności:

- $EN_{(0)} := \{(\perp, \emptyset, \emptyset), (b_1^{(0)}, \perp), (b_2^{(0)}, \perp), \ldots, (b_{n_0}^{(0)}, \perp)\}$,
- $EN = \{en_0, en_1, en_2, \ldots, en_n\} \cup \bigcup_{i=0,1,\ldots,n} b^{(i)}$, *przy czym* $b^{(i)} := \{(b_1^{(i)}, en_i), (b_2^{(i)}, en_i), \ldots, (b_{n_i}^{(i)}, en_i)\}$,
- $R_{(n)} := R_{EN}$,
- $EN_{(i)} \subseteq EN_{(i+1)} \subseteq EN$, *dla* $i < n$,
- $R_{(i)}$ *to zbiór powstały przez usunięcie ograniczeń dla zdarzeń należących do* $EN' - EN_{(i)}$,
- $EN_{(n)} := EN$.

Zbiór EN reprezentuje zbiór zdarzeń i warunków przebiegu czasowego (EN, R_{EN}) z funkcją θ_{EN}, wygenerowanego przez algorytm nr 4. EN jest podzbiorem UNF_N postaci
.

Następnie zostanie pokazane krok po kroku, że jeżeli kolejne prefiksy $(EU_{(i)}, \tau_{(i)})$ przebiegu (E_a, θ_a) zawierają się w $(U(N), \theta)$ to jego odpowiedniki $(EN_{(i)}, R_{(i)})$ muszą także zawierać się w (UNF_N, R).

Najwcześniejszym zdarzeniem w E_a jest zdarzenie $\perp$ inicjujące system, reprezentowane przez e_0 o momencie $\tau_0(e_0)$. Zdarzenie to produkuje warunki e_0G, a fragment $(EU^{(0)}, \tau_{(0)})$ procesu czasowego warunki $cut(EU^{(0)})$ ($e_0G \subseteq cut(EU^{(0)})$).

Natomiast algorytm nr 4 dodaje do prefiksu zdarzenie początkowe $\perp$ reprezentowane przez en_0 i wyprodukowane warunki $b^{(0)}$. Stąd $EN^{(0)} = \{en_0\} \cup b^{(0)}$ i $EN^{(0)} \subseteq UNF_N$. Ponadto $R_{(0)} = \{R(en_0)\}$, gdzie $R(en_0) = \{\theta_R(en_0) = 0\}$.

Kolejnym zdarzeniem w (E_a, θ_a) *jest* e_1*, które ma zapewnione warunki* $Ge_1 \cup He_1 \subseteq cup(EU^{(0)})$*, oraz moment wykonania* $\tau_{(1)}(e_1)$ *taki, że:*

i $\tau_{(1)}(e_1) \geqslant \max(\{\tau_{(1)}(Gb) : b \in Ge_1 \cup He_1\})$,

ii $\forall_{b \in Ge_1 \cup He_1} \min I(l(b), l(e_1)) \leqslant \tau_{(1)}(e_1) - \tau_{(1)}(Gb) \leqslant \max I(l(b), l(e_1))$,

iii $\forall_{e' \in E \ \wedge \ e' \text{ conf } e_1} \tau_{(1)}(e') = \infty$,

iv $\forall_{e' \in E \ \wedge \ e' \nearrow e_1} \tau_{(1)}(e') \leqslant \tau_{(1)}(e_1)$

Więc fragment procesu czasowego po dodaniu e_1 *stanowi* $(E^{(1)}, \tau_{(1)})$.

Natomiast algorytm nr 4 identyfikuje zdarzenia mogące wystąpić dla zbioru $EN^{(0)}$ *i* $R_{(0)}$.

Zdarzenia te stanowią zbiór $PE = pe(EN^{(0)})$*. Wśród tych zdarzeń znajduje się także* en_1 *odpowiadające zdarzeniu* e_1.

Ponieważ PE *jest skończonym zbiorem to zdarzenie* en_1 *zostanie w końcu wybrane przez algorytm nr 4 zgodnie z punktem 1.*

Zgodnie z punktami 2.1-2.2 algorytmu nr 4 zostaną określone ograniczenia momenu $\theta_R(en_1)$*: dla każdego* $p \in F\{l(en_1)\}$*, gdzie* $l(b) = p$ *i* $\delta = \theta_R(Gb)$ *zostaną dodane ograniczenia:* $\min I(l(b), l(en_1)) + \delta \leqslant \theta_R(en_1) \leqslant \max(l(b), l(en_1)) + \delta$ *do* $R(en_1)$.

Podsumowując, punkty 1, 2.1 i 2.2 algorytmu nr 4 odpowiadają punktom i-ii.

Następnie zgodnie z punktami 3.1-3.3 algorytmu nr 4, zbiór $EN^{(1)} = EN^{(0)} \cup \{en_1\} \cup b^{(1)}$.

Zgodnie z punktem 4.1.1 algorytmu nr 4, każde zdarzenie przyczynowo poprzedzające en_1 *w* E_1 *wykonało się wcześniej niż* en_1*. Powoduje to modyfikację* $R(en_1)$*:*

$$\forall_{e' \in [GGen_1]_{EN^{(1)}}} R(en_1) = R(en_1) \cup \{\theta_{EN}(e') \leqslant \theta_{EN}(en_1)\}.$$

Zgodnie z punktem 4.1.2 algorytmu nr 4, każde zdarzenie e' *słabo przyczynowo poprzedzające musiało wykonać się przed* en_1*. Powoduje to modyfikację* $R(e')$*:*

$$\forall_{e' \in [(Gen_1)H]_{EN^{(1)}}} R(e') = R(e') \cup \{\theta_{EN}(e') \leqslant \theta_{EN}(en_1)\}.$$

Zgodnie z punktem 4.1.3 algorytmu nr 4 zdarzenie en_1 *powinno wykonać się przed zdarzeniami, które zużywają warunki czytane przez nie:*

$$\forall_{e' \in [(Hen_1)G]_{EN^{(1)}}} R(en_1) = R(en_1) \cup \{\theta_{EN}(en_1) \leqslant \theta_{EN}(e')\}.$$

Zgodnie z punktem 4.2.1 algorytmu nr 4, zdarzenia konfliktowe z en_1 *w* $EN^{(1)}$ *nie wystąpiły, co powoduje modyfikację:*

$$\forall_{e' \in [(Gen_1)G]_{EN^{(1)}} \wedge e' \neq e} R(en_1) = R(en_1) \cup \{\theta_{EN}(e') = \infty\} \wedge R(e') = R(e') \cup \{\theta_{EN}(en_1) = \infty\}.$$

Punkty 4.1.1-4.1.3 algorytmu nr 4 odpowiadają punktowi iii. Natomiast punkt 4.2.1 odpowiada puntkowi iv.

Powyższy tok rozumowania należy powtórzyć $n-1$ razy, aby zauważyć, że informacje z (E_a, θ_a) są zawarte w (EN, R_{EN}) z funkcją θ_{EN}, który zawiera się w $(UNF(N), R)$ z funkcją θ_R. Co jest sprzeczne z założeniem braku takiego odpowiednika.

Tak więc zbiór generowany przez algorytm nr 4 zawiera zdarzenie każdej umożliwionej tranzycji modelu systemu c-TdPN.

□.

Analiza złożoności dla algorytmu nr 4 została przeprowadzona w zależności od parametrów modelu systemu N:

m - liczby miejsc,

n - liczby tranzycji,

l - maksymalnej liczby łuków wyjściowych lub wejściowych do tranzycji,

oraz w zależności od:

k - liczby iteracji pętli głównej,

a - liczby operacji potrzebnych do obliczenia $cut(\lceil e \rceil)$.

Jako operacje jednostkowe wybrano:

- operacje arytmetyczne,
- tworzenie składowych elementu,
- dodawanie i usuwanie elementu do/ze zbioru.

Oszacowana liczba operacji jednostkowych potrzebna do zainicjowania algorytmu to:

$$init(l, m, n) = 8 + 7m + m + n + l \cdot n.$$

Każde wykonanie pętli głównej ma oszacowaną liczbę operacji jednostkowych:

$$loop(i, l, m, n) = 24 + 11l + n \cdot (12l + l^3 \cdot (i + 1) + 2l^2 \cdot (i + 1)) + cutOff(i, l, m, n)$$

przy czym funkcja $cutOff$ reprezentuje szacowaną liczbę operacji potrzebnych do sprawdzenia czy badane zdarzenie jest zdrzeniem odcięcia. Funkcja ta wyraża się wzorem:

$$cutOffEventE(o, l, m, n) = i^2 l^2 + (i - 1) \cdot (i^2 l^2 + m^2 + m + 2 \cdot m \cdot ageCost(l, m, n))$$

gdzie $ageCost(l, m, n) = 5 + 2 \cdot m + 2 \cdot n + 5l \cdot n$, jest szacownym kosztem obliczenia $age(b, \theta, A)$.

Podsumowując szacowana liczba operacji jednostkowych potrzebnych do działana algorytmu nr 4 wyraża się wzorem:

$$complexRPC(a, k, l, m, n) = init(l, m, n) + \sum_{i=1}^{k} loop(i, l, m, n).$$

Zatem pesymistyczna złożoność algorytmu nr 4 jest rzędu:

$$O(k^4 \cdot l^2) + O(k^2 \cdot l^3 \cdot n) + O(k^2 \cdot l \cdot m \cdot n) + O(k^2 \cdot m^2).$$

4.3 Algorytm konstruowania reprezentatywnego czasowego systemu tranzycyjnego (modelu TTS)

Konstrukowanie prefiksu rozwinięcia dla systemów c-TdPN, których zachowanie się jest reprezentowane przez czasowy system tranzycyjny, może przebiegać podobnie jak dla systemów TPN w algorytmie nr 1. Różnica w nowym proponowanym rozwiązaniu to zastosowanie szkieletu tegoż algorytmu do innego modelu sieci Petriego, inny sposób wyznaczania stanów symbolicznych $\overrightarrow{Post}(s)$ i $Post_t(s)$ oraz inny sposób weryfikowania warunku stopu $s \notin Pass$.

W modelu c-TdPN tranzycja może być przedawniona pod wpływem upływu czasu w odróżnieniu od tranzycji w TPN.

Pilność wykonania tranzycji można wymusić poprzez założenie maksymalnego czasu wykonania tranzycji od momentu jej umożliwienia [2] lub poprzez dodanie ograniczeń co do przebywania żetonów w miejsach.

Dla celów praktycznych do c-TdPN zostaną dodane dla miejsc ograniczenia co do długości przebywania w nich żetonu, zwane *inwariantami* (inv).

Definicja 9 *Model systemu c-TdPN z inwariantami to model c-TdPN rozszerzony o dodatkową relację "inv" pozwalajacą na określanie maksymalnej długości przebywania żetonu w wybranym miejscu modelu systemu.*

Ponadto niektóre pojęcia wymagają przedefiniowania.

Niech $N = (P, T, F, C, I, inv, m_0)$ oznacza model c-TdPN z inwariantami, a $S_N = (Q, \{q_0\}, \Sigma, \longrightarrow)$ czasowy systemem tranzycyjny reprezentujący jego zachowanie, gdzie:

- $Q = (P \to \mathbb{N} \cup \{0\}) \times (T \to \mathbb{R}_+ \cup \{0\})$,
- $q_0 = (m_0, v_0)$,
- $\Sigma = T$,
- $\longrightarrow \subseteq Q \times (T \cup \mathbb{R}_+ \cup \{0\}) \times Q$.

v_0 to funkcja przyporządkowująca tym razem każdemu miejscu wartość 0. Zegary są przyporządkowane do miejsc modelu sieci N. Dla każdego $p_i \in P$ istnieje dokładnie jeden zegar $x_i \in X$. Zakładamy także, że w miejscu może być co najwyżej 1 żeton. Stąd jeżeli x_p to zegar przypisany do miejsca p to $inv(x_p)$ oznacza inwariant przypisany do tego miejsca.

Zmiana stanu $q = (m, v)$ na $q' = (m', v')$ powstała na skutek uruchomienia tranzycji t, oznaczana jako $q \xrightarrow{t} q'$, jest możliwa, jeżeli:

- $m' = (m - Ft) \cup tF$,
- dla każdego $p \in Ft \cup Ct$:
  - $m(p) \geqslant 1$,

– $\min I(p,t) \leqslant v(p) \leqslant \max I(p,t)$,

– $v'(p) = 0$ gdy $p \in tF$,

gdzie $v(p) = x_p$ to wskazanie zegara przypisanego do miejsca p.

Zmiana stanu q na q' powstała na skutek upływu d jednostek czasu, oznaczana $q \xrightarrow{d} q'$ jest możliwa, jeżeli:

- $m' = m$,
- $d \in R_+ \cup \{0\}$,
- dla każdego $p \in Ft \cup Ct : v'(p) = v(p) + d$, oraz
- dla każdego miejsca p w którym jest żeton:
  - $v'(p) = v(p) + d$
  - $v'(p) \leqslant inv(p)$.

System tranzycyjny S_N po dyskretyzacji stanowi system dyskretny

$$SA_N = (QA, QA_0, \Pi, \Longrightarrow)$$

gdzie zbiory QA, QA_0, Π, $\Longrightarrow$ oznaczają kolejno zbiór stanów symbolicznych, stanów symbolicznych początkowych, akcji, oraz przejść systemu. System SA_N może być reprezentowany przez graf o wierzchołkach w QA, krawędziach opisanych relacją $\Longrightarrow \subseteq QA \times \Pi \times QA$ i oznakowanych akcjami z Π. Akcje systemu reprezentują uruchomione trancyjne ze zbioru T.

Stan symboliczny agreguje niektóre stany systemu S_N o tym samym znakowaniu. Stan symboliczny s jest reprezentowany przez $s = (m, Z)$, przy czym m to znakowanie sieci N, a Z to strefa ograniczająca wskazania zegarów przypisanych do miejsc, w których jest przynajmniej jeden żeton w m oraz zegara zerowego.

Wskazania zegarów są ograniczone przedziałem $[0, min)$, gdzie min to najmniejsza wartość $inv(p)$ dla miejsca w którym jest żeton lub ∞ gdy brak ograniczeń.

Zegar zerowy stale wskazuje wartość 0. Symboliczny stan początkowy $s'_0 = (m_0, Z'_0)$ stanowi znakowanie startowe m_0 oraz strefę Z_0. Strefa ta ogranicza wskazania zegarów przypisanych do miejsc gdzie są żetony w stanie początkowym w taki sposób, że każdy zegar ma wskazywać 0 (Z'_0 to macierz zerowa).

Zamiana stanu symbolicznego s na s'' reprezentowana przez $\Longrightarrow$ polega na:

1. wyznaczeniu stanu symbolicznego agregującego wszystkie możliwe stany systemu S_N powstałe przez uruchomienie tranzycji t w s, reprezentowanym przez $s' = PostTd_t(s)$,

2. wyznaczeniu dla s' wszystkich stanów systemu S_N powstałych przez możliwie najdłuższy upływ czasu reprezentowany przez $\overrightarrow{PostTd}(s')$.

W skrócie $s'' = (\overrightarrow{PostTd} \circ PostTd_t)(s)$, przy czym strefa w s' nie może być pusta. W razie gdy strefa w s'' lub s' jest pusta oznacza to, że t nie była umożliwiona w s.

Definicja 10 *Stan symboliczny $s' = PostTd_t(s) = (m', Z')$ powstaje przez uruchomienie umożliwonej tranzycji t w stanie symbolicznym s, gdzie:*

1. $m' = (m - Ft) \cup tF$
2. $Z' = ((Z \cap \forall_{p \in Ft}\{\min I(p,t) \leqslant x_p \leqslant \max I(p,t)\})_{|Ol}) \cap \bigcap_{x_j \in Ne}\{x_j = 0\}$

przy czym zegar x_p jest powiązany z miejscem p.

Stan $s' = PostTd_t(s)$ w praktyce obliczany jest etapowo:

- wyznaczenie nowego znakowania $m' = (m - Ft) \cup tF$,
- do Z' przypisanie strefy powstałej przez nałożenie dla Z ograniczeń wynikających z umożliwienia tranzycji t:

$$Z' = Z \cap \forall_{p \in Ft}\{\min I(p,t) \leqslant x_p \leqslant \max I(p,t)\},$$

- do Z' przypisanie jej postaci kanonicznej,
- pozostawienie w Z' informacji o ograniczeniach dla zegara x_0 i tych zagarów przypisanych do miejsc $m - Ft$ w których jest przynajmniej jeden żeton (zbiór Ol),
- dodanie kolejno do Z' ograniczeń nowych zresetowanych zegarów powiązanych z miejscami tF (zbiór Ne),
- różnice zegarów z Ne i tych co już były w strefie Z' z braku ograniczenia ustawiane są na ∞,
- końcowo do Z' przypisanie jej postaci kanonicznej.

Definicja 11 *Stan symboliczny $s'' = \overrightarrow{PostTd_d}(s') = (m'', Z'')$ powstaje w wyniku najdłuższego możliwego upływu czasu d od momentu ustanowienia stanu s', gdzie:*

- $m'' = m'$,
- $Z'' = \overrightarrow{Z'} \cap \forall_{x \in X_{Z'}} x \leqslant d \wedge x \leqslant inv(x)$,

a $\overrightarrow{Z'}$ jest przyszłością strefy Z'.

Brak skończonej wartości dla $d = \infty$ oznacza brak ograniczenia co do maksymalnego momentu opóźnienia: $\overrightarrow{PostTd_d}(s') = \overrightarrow{PostTd}(s')$, co redukuje ostatni punkt do: $Z'' = \overrightarrow{Z'} \cap \forall_{x \in X_{Z'}} x \leqslant inv(x)$.

Na zbiór QA_N składają się tylko stan symboliczny $s_0 = \overrightarrow{PostTd_d}(s'_0)$ oraz wszystkie stany symboliczne, które można otrzymać przez zastosowanie złożenia $\overrightarrow{PostTd} \circ PostTd_t$ dla wyprodukowanych wcześniej stanów poczynając od s_0, gdzie $t \in T$ jest umożliwioną tranzycją dla tych stanów.

Tak określone pojęcia zostają następnie użyte zamiennie w algorytmie nr 1 tworząc nowy algorytm produkujący graf stref dla modeli systemów c-TdPN z inwariantami i dodatkowym warunkiem stopu.

Algorytm 5: Algorytm generujący graf stref modeli systemów c-TdPN

Dane:
system - to zainicjowany system c-TdPN z inwariantami z dodatkowym miejscem technicznym g zawierającym żeton i przypisany zegar wskazujący na 0 ($x_g = 0$)
$s'_0 = (m_0, Z'_0)$ - moment początkowy modelu systemu, dla każdych i, j: $Z'_0.z_{ij} = 0$
d - ograniczenie maksymalnego momentu generowania przestrzeni stanów lub $d = \infty$ gdy brak ograniczenia
$s_0 = \overrightarrow{PostTd_d}(s'_0)$
$wait := \{s_0\}$
$pass = \emptyset$
approx - wybrana aproksymacja stanów: brak, zwykła k, k_x, LU_x lub Exp
Rezultat: Wygenerowana zdyskretyzowana przestrzeń stanów ci-TdPN, zapisana w $Pass$
Algorytm:

```
1: pass=GenerateZoneGraph(system,wait,pass,approx):
2: while wait ≠ ∅ do
3:    s = pop(wait);
4:    if ¬approx.Include(s, pass) then
5:       for t ∈ system.enabledTd(t, s) do
6:          s' = PostTd_d(PostTd_t(s));
7:          if Z_s' ≠ ∅ then
8:             wait = wait ∪ {s'};
9:          end if
10:      end for
11:      pass = pass ∪ {s};
12:   end if
13: end while
```

Po wywołaniu funkcji *GenerateZoneGraph*() zbiór *pass*, w przypadku gdy wyko-

nanie algorytmu zakończy się, przechowuje wszystkie węzły przestrzeni stanów modelu systemu do momentu d, lub jej reprezentatywny fragment w przypadku $d = \infty$.

Rolą głównej pętli funkcji jest sprawdzanie czy są jeszcze nieprzebadane stany (linie 2-13). W przypadku gdy nie ma już nieprzebadanych stanów wykonanie algorytmu kończy się. W przeciwnym razie nieprzebadany stan zostaje przypisany do symbolu s i usunięty ze zbioru *wait* (linia 3). Następnie jeżeli w *pass* znajduje się stan, który ma taką samą aproksymację *approx* co s, to algorytm przechodzi do kolejnego obiegu pętli głównej, a s jest pomijany (linia 4). W przeciwnym razie dla każdej czynnej tranzycji t w stanie s obliczany jest nowy stan symboliczny s' i dodany do zbioru *wait* w przypadku gdy ma niepustą strefę (linie 5-10). Następnie zapisywana jest informacja o przebadanym stanie s (linia 11), a wykonanie algorytmu przechodzi ponownie do pętli głównej.

Następnie zostanie przedstawiony dowód na to, że przestrzeń stanów SA, bez stosowania aproksymacji, zawiera wszystkie przebiegi modelu systemu c-TdPN.

Twierdzenie 2 *Niech N oznacza model systemu typu c-TdPN, a model S_N czasowy system tranzycyjny reprezentujący jego semantykę. Algorytm nr 5 generujący graf stref SA_N (bez aproksymacji) modeluje wszystkie możliwe do osiągnięcia stany modelu systemu N.*

W dowodzie Tw. 2 korzysta się z pojęcia ścieżki symbolicznej w SA_N.

Definicja 12 Ścieżką symboliczną π *w SA_N nazywamy sekwencję stanów symbolicznych postaci:*

$$\pi : \pi[0] \Rightarrow \pi[1] \Rightarrow \pi[2] \Rightarrow \ldots$$

przy czym stan symboliczny $\pi[i] = (M_{\pi[i]}, Z_{\pi[i]})$ jest nazywany węzłem ścieżki*, gdzie $M_{\pi[i]}$ to znakowanie systemu, $Z_{\pi[i]}$ to strefa oraz $v \in Z_{\pi[i]}$ jest wybranym wartościowaniem zegarów przyporządkowanych do miejsc, w których wystąpił żeton w $M_{\pi[i]}$.*

Zbiór wszystkich ścieżek w SA_N, które startują w s, oznaczany jest przez $\Pi(s)$.

Dowód 2 *Niech ρ oznacza dowolną ścieżkę wychodzącą ze stanu poczatkowego q_0 w S_N (jego przebieg) oraz niech SA_N nie ma ograniczenia czasowego, $d = \infty$. Stosując technikę dowodu nie wprost załóżmy, że nie znajdziemy w SA_N takiej ścieżki symbolicznej π, która zawierałaby ścieżkę ρ.*

Następnie, aby doprowadzić tok rozumowania do sprzeczności zostanie pokazane, że rozważając krok po kroku stany ścieżki ρ w S_N zawsze znajdzie się taka ścieżka symboliczna w SA_N, że zawiera te stany.

Należy rozważyć dwa przypadki: ścieżki nieskończonej oraz skończonej gdy inwariant powoduje zakleszczenie.

W przypadku gdy ścieżka jest nieskończona to ρ ma postać:

$$\rho = q_0 \xrightarrow{d_0} q_0 + d_0 \xrightarrow{a_0} q_1 \xrightarrow{d_1} q_1 + d_1 \xrightarrow{a_1} \ldots.$$

Ponieważ $q_0 = (m_0, v_0) \in \rho$ jest stanem początkowym to w takim razie znajdziemy taki stan symboliczny $s_0 = (M_0, Z_0)$ w SA_N w którym $m_0 = M_0$ oraz $v_0 \in Z_0$, ponieważ stan symboliczny s_0 powstał przez upływ czasu poczynając od stanu poczatkowego q_0 to również go zawiera.

Jeżeli ścieżka ρ od stanu q_0 reprezentuje jedynie upływ czasu w znakowaniu m_0 to cała ścieżka zawarta jest w stanie symbolicznym s_0. Stąd w SA_N znajduje się ścieżka symboliczna zawierająca fragment: $q_0 \xrightarrow{d_0} q_0 + d_0$ dla $d_0 \in R_+ \cup \{0\}$.

W przeciwnym razie po upływie d_0 jednostek czasu w stanie $\delta_0 = q_0 + d_0$ zostaje wykonana akcja a_0 reprezentująca uruchomienie tranzycji t powodując stan q_1. Tak więc w SA_N dla każdego $p \in Ft \cup Ct$ prawdą jest, że $\min I(p,t) \leqslant v_0(p) + d_0 \leqslant \max I(p,t)$ oraz dla każdego p w którym jest żeton $v_0(p) + d_0 \leqslant inv(p)$. Stąd ponieważ t jest umożliwiona w stanie δ_0 zawartym w s_0 to tym bardziej $t \in enabledTd(s_0)$ więc istnieje taki stan symboliczny s_1, że $(s_0, s_1) \in \Longrightarrow$. Oznacza to, że $s_1 = \overrightarrow{PostTd}(PostTd_t(s_0)) = (M_1, Z_1)$ gdzie:

- $M_1 = M_0 - Ft + tF$,
- $Z' = ((Z \cap \forall_{p \in Ft}\{\min I(p,t) \leqslant x_p \leqslant \max I(p,t)\})_{|Ol}) \cap \bigcap_{x_j \in Ne}\{x_j = 0\}$ *oraz*
- $Z_1 = \overrightarrow{Z'} \cap \forall_{x \in X_{Z'}} x \leqslant inv(x)$.

Tak więc strefa Z_1 powstała ze strefy Z_0 przez usunięcie ograniczeń zegarów powiązanych z miejscami Ft oraz dodaniu zresetowanych zegarów powiazanych z miejscami tF, a następnie obliczeniu przyszłości dla tak otrzymanej strefy Z' i ograniczenia jej do inwariantów jej zegarów.

Ponadto wystąpienie akcji a_0, powoduje że $v_1(x) = 0$ to moment dla zresetowanych zegarów x przez tą akcję, oraz $v_1(x) = v_0(x) + d_0$ dla pozostałych, gdzie v_1 to funkcja wskazań zegarów w stanie q_1.

Stąd w SA_N znajduje się ścieżka zawierająca fragment $q_0 \xrightarrow{d_0} q_0 + d_0 \xrightarrow{a_0} q_1$.

Analogiczny tok rozumowania dla obu przypadków można zastosować dla kolejnych stanów q_1, $q_1 + d_1$, q_2, $q_2 + d_2$, ... itd.

Stąd sprzeczność, ponieważ dla dowolnej ścieżki $\rho \in \pi(q_0)$ z S_N zawsze znajdziemy ścieżkę symboliczną $\pi \in \Pi(s_0)$ w SA_N, która ją zawiera.$\square$.

W przypadku gdy ograniczenia czasowe dla zegarów rosną nieskończenie to algorytm nr 5 zakończy swoje wykonanie tylko wówczas, gdy zostanie uwzględnione skończone ograniczenie d maksymalnego momentu działania systemu lub zastosowana aproksymacja pozwalająca na wychwycenie stanów, z których można uruchomić ten sam zestaw tranzycji w takich samych momentach, zwanych *dziedziną uruchomień*. Poniżej została przywołana definicja dziedziny uruchomień [39] (str.97) dostosowana do modelu c-TdPN.

Definicja 13 *Niech (m, Z) będzie stanem symbolicznym,gdzie Z jest w postaci kanonicznej, oraz $q = (m, v)$ jest jednym z jego stanów.* Dziedziną uruchomienia *tranzycji w stanie q jest przekształcenie I_v stanu q jest definiowana jako $I_q : enabledTd(M) \rightarrow$*

$(R_+ \cup \{0, \infty\}) \times (R_+ \cup \{0, \infty\})$ *przyporządkowujące każdej tranzycji* $t_i \in enabledTd(M)$ *przedział półosi dodatniej* $I_q(t_i)$*, którego minimum określa czas oczekiwania na najwcześniejszy moment umożliwienia, a maksimum na najpóźniejszy moment umożliwienia.* $I_q(t_i) = \bigcap_{p_j \in Ft_i \cap Ct_i} [max(0, \min I(p_j, t_i) - v_j), \max I(p_j, t_i) - v_j]$*. Dziedzina uruchomień stanu symbolicznego* s *stanowi unię przedziałów* I_q *po stanach* $q \in s$*.*

W przypadku stref definiowana jest aproksymacja pozwalająca porównać dwie strefy.

Do badań nad systemami c-TdPN z inwariantami wybrano cztery znane z literatury aproksymacje, w tym rozwiązanie zastosowane w narzędziu TAPAAL. Są to kolejno aproksymacje k, k_x, LU_x [4], ext_{pl} [12]. Każda z aproksymacji została dostosowana dla modelu c-TdPN z inwariantami.

Niech $s' = (m', Z')$ oznacza stan powstały przez aproksymację stanu $s = (m, Z)$ oraz $x_i, x_j \in X$ zbiór zegarów stref Z i Z' (łącznie z zegarem zerowym). Aproksymacja nie zmienia znakowania stanu symbolicznego, a jedynie wartości ograniczeń zapisanych w strefie tak, aby nie powodować powiększenia dziedziny uruchomień czynnych tranzycji w s.

Zwykła aproksymacja k polega na wybraniu jednej liczby całkowitej k, równej największemu skończonemu ograniczeniu w modelu sieci. Jeżeli wskazanie zegara przekroczy liczbę k, to jego dalsza wartość precyzyjna jest zbędna. Nowa stref $Z' = k_a(Z)$ jest obliczana wg wzoru:

$$z'_{ij} = \begin{cases} \infty & dla \quad z_{ij} > k \\ -k & dla \quad z_{ij} < -k \\ z_{ij} & dla \quad pozostałych \end{cases}$$

Aproksymacja k_x różni się tym od aproksymacji k, że dla każdego zegara x jest dobierana osobna wartość k_x odpowiadająca najwięszemu skończonemu ograniczeniu w modelu dla tego zegara. Strefa $Z' = k_a^x(Z)$ jest obliczana wg wzoru:

$$z'_{ij} = \begin{cases} \infty & dla \quad z_{ij} > k_{x_i} \\ -k_{x_j} & dla \quad z_{ij} < -k_{x_j} \\ z_{ij} & dla \quad pozostałych \end{cases}$$

W *aproksymacji* LU_x każdy zegar x_i ma przyporządkowaną parę $L(x_i)$ i $U(x_i)$, gdzie $L(x_i)$ to maksymalne dolne ograniczenie dla x_i, a $U(x_i)$ to jego maksymalne górne ograniczenie w modelu. Stefa $Z' = LU_a^x(Z)$ jest obliczana wg wzoru:

$$z'_{ij} = \begin{cases} \infty & dla \quad z_{ij} > L(x_i) \\ -U(x_i) & dla \quad -z_{ij} > U(x_j) \\ z_{ij} & dla \quad pozostałych \end{cases}$$

Kolejna aproksymacja, zwana ekstrapolacją stosowaną w narzędziu TAPAAL, wymaga dostosowania do modelu c-TdPN z inwariantami oraz wprowadzenia dodatkowych pojęć. W aproksymacji opuszczono fragmenty dotyczące łuków transportu, ze wzgledu na ich brak w rozważanym modelu.

Definiujemy maksymalną stałą, różną od ∞, występującą w inwariancie przypisanym do miejsca p lub występującą w przedziałach dostępności przypisanych do łuków wychodzących z p. Stała reprezentowana jest przez

$$mc(p) = \max\{mci(i(p)), \max_{t \in pF}\{mci(c(p,t))\}\}$$

gdzie $i(p)$ oznacza przedział czasu jaki może osiągnać zegar przy obecności żetonu w miejscu p, $c(p,t)$ oznacza przedział dostępności przypisany do łuku (p,t), a $mci(I)$ maksymalną liczbę całkowitą nieujemną zawartą w I.

Ponadto $lb^{\eta}_{Z}(p) = -Z_{0,p}$ oraz $ub^{\eta}_{Z}(p) = Z_{p,0}$ są odpowiednio dolnym i górnym ograniczeniem dla wskazania zegara przypisanego żetonu w miejscu p.

Strefa $Z' = ext_{pl}(Z)$ jest definiowana (dla każdych miejsc $p_i, p_j \in P$, $p_i \neq p_j$):

1. $Z' = Z$
2. Jeżeli $mc(p_i) < lb^{\eta}_{Z}(p_i)$ to $lb^{\eta}_{Z'}(p_i) := mc(p_i)$ i $ub^{\eta}_{Z'}(p_i) := \infty$
3. Jeżeli $ub^{\eta}_{Z}(p_i) > mc(p_i)$ to $ub^{\eta}_{Z'}(p_i) := \infty$
4. Jeżeli $mc(p_i) < lb^{\eta}_{Z}(p_i)$ lub $mc(p_j) < lb^{\eta}_{Z}(p_j)$ to $Z'_{p_i,p_j} := \infty$
5. Jeżeli $Z_{p_i,p_j} > mc(p_i)$ to $Z'_{p_i,p_j} := \infty$

Metoda ext_{pl} w stosunku do opisanej w [12] nie uwzględnia łuków transportu oraz zakłada, że każda ze stałych jest porównywana relacją $\leqslant$, a porównanie z nieskończonością jest redukowane do relacji $<$. Modyfikacja ta nie narusza poprawności metody aproksymacji.

Twierdzenie 3 *Algorytm nr 5 konstruujący graf stref SA_N osiągnie warunek stopu $wait = \emptyset$.*

W dowodzie twierdzenia 3 należy pokazać, że każda ścieżka symboliczna, mająca nieskończoną ilość węzłów, zawiera dwa węzły, które mają tę samą dziedzinę wykonań umożliwionych tranzycji.

4.4 Dodatkowy warunek momentu stopu w algorytmie konstruowania modelu TTS

Algorytm nr 5 ma zdefiniowany dodatkowy warunek stopu, dzięki któremu możliwe jest wygenerowanie początkowego fragmentu przestrzeni stanów badanego modelu systemu do pewnego momentu d bez użycia aproksymacji. Spowoduje to wygenerowanie uboższego w informacje fragmentu zachowania (zdyskretyzowanej przestrzenii stanów modelu systemu), aczkolwiek umożliwi weryfikację jego własności na podstawie zawartych ścieżek i stanów możliwych do osiągnięcia bez przekraczenia momentu d.

Fragment taki może następnie być wykorzystany do:

- szybszego uzyskania reprezentacyjnego modelu zachowania ograniczonego do zadanego momentu,
- szybszej weryfikacji własności z wczesnym momentem przerwania,
- sprawdzenie czy do momentu d zostanie przeprowadzona badana akcja systemu,
- odpowiedź na pytanie o to co może wydarzyć się w systemie do najwyżej zadanego momentu d.

Dodatkowy warunek stopu umożliwia także badanie działania algorytmu:

- do momentu d z wybraną aproksymacją,
- do momentu d bez aproksymacji.

4.5 Logika TdPN-TCTL(TTS) do opisu własnosci zachowań modelu systemu

Weryfikowane własności zostaną opisane w logice nazywanej c-TdPN-TCTL, będącej rozszerzeniem logiki TCTL.

Niech N, S_N i SA_N będą zdefiniowane jak w poprzednim rozdziale.

Syntaktyka logiki c-TdPN-TCTL ma następującą postać:

$$\varphi := true|\gamma|\varphi \wedge \varphi|\varphi \vee \varphi|\neg\varphi|E\varphi U_I\varphi|A\varphi U_I\varphi$$

przy czym *true* to słowo kluczowe, a $\gamma \in GMEC$.

Semantyka logiki ci-TdPN-TCTL została zdefiniowana przez operator $\models$:

$$\begin{aligned}
s &\models true \\
s &\models \gamma \quad wtw \quad m \text{ zapewnia spełnienie formuły } \gamma \\
s &\models \neg\varphi \quad wtw \quad \neg(s \models \varphi) \\
s &\models \varphi \wedge \psi \quad wtw \quad s \models \varphi \wedge s \models \psi \\
s &\models \varphi \vee \psi \quad wtw \quad s \models \varphi \vee s \models \psi \\
s &\models x_g \in I \quad wtw \quad x_g \in I \cap [-s.z_{0g}, s.z_{g0}] \\
s &\models E\varphi U_I\psi \quad wtw \quad \exists \rho \in \Pi(s), \exists_{i \geqslant 0}: \\
&\qquad\qquad \rho[i] \models x_g \in I \wedge \psi \text{ oraz} \\
&\qquad\qquad \forall_{j:0 \leqslant j < i} \rho[j] \models \varphi \\
s &\models A\varphi U_I\psi \quad wtw \quad \forall \rho \in \Pi(s), \exists_{i \geqslant 0}: \\
&\qquad\qquad \rho[i] \models x_g \in I \wedge \psi \text{ oraz} \\
&\qquad\qquad \forall_{j:0 \leqslant j < i} \rho[j] \models \varphi
\end{aligned}$$

przy czym $s = (m, Z)$ to stan symboliczny, x_g to zegar przypisany do żetonu znajdującego się w technicznym (sztucznym) miejscu systemu, służącym do odmierzania czasu od momentu jego inicjacji. Natomiast $s.z_{ij}$ oznacza ograniczenie zegarów i i j ze strefy węzła s.

5 Ocena jakości proponowanego rozwiązania

5.1 Algorytmy weryfikacji własności modelu systemu

W wyniku analizy prefiksu rozwinięcia reprezentowanego przez TTS oraz prac [9, 26] autor opracował algorytmy weryfikacji formuł logiki c-TdPN-TCTL. Należą do nich formuły typu: $EF_I\varphi$, $AF_I\varphi$, $EG_I\varphi$, $AG_I\varphi$ oraz złożone $AG_I\varphi \rightarrow EF_J\psi$ i $AG_I\varphi \rightarrow AF_J\psi$.

Niech ponownie N, S_N i SA_N oznaczają kolejno model systemu, model jego zachowania, model zachowania po dyskretyzacji.

W dalszej części stan symboliczny będzie nazywany węzłem o pewnych wyróżnionych cechach. Niech dalej φ, ψ będą formułami GMEC. Węzeł w ma następujące cechy:

- $w.m$ - znakowanie,
- $w.Z$ - strefa,
- $w.z_{ij}$ konkretne ograniczenie górne różnicy zegarów $x_i - x_j$ ze strefy Z,
- $-w.z_0g$ i $w.z_{g0}$ to odpowiednio najwcześniejsze i najpóźniejsze możliwe wskazanie x_g w węźle w,
- $w.potomni$ - zbiór węzłów do których wychodzi strzałka $\Rightarrow$ z węzła w ($w.potomni = \emptyset$ gdy brak strzałek),
- $w.ref$ to węzeł aproksymowany z w (zawsze gdy $w.ref \neq \emptyset$ to $w.potomni = \emptyset$),
- $w.spel(\varphi)$ - prawdziwe gdy znakowanie $w.m$ zapewnia prawdziwość φ,
- $w.espel(\varphi, I)$ - prawdziwe, gdy formuła φ jest spełniona w momencie z przedziału $I \cap w.Z$ przynajmniej w jednej ścieżce przechodzącej przez węzeł w, gdzie I to przedział w którym ma znaleźć się wskazanie zegara x_g.

Niech I, J będą przedziałami półosi dodatniej, przy czym $I = [I.a, I.b]$ oraz $J = [J.a, J.b]$. Dla przedziałów definiuje się operacje:

- odejmowania (redukcji) wartości skalarnej c: $J - c = [J.a - c, J.b - c]$,
- bezpiecznej redukcji: $I = I.redukuj(c) = [\max\{0, I.a - c\}, \max\{0, I.b - c\}]$.

Ponadto niech graf stref SA_N oznacza przestrzeń stanów systemu N po dyskretyzacji, a $s_0 = \overrightarrow{PostTd_d}(s'_0)$ węzeł początkowy grafu.

5.1.1 Algorytm weryfikacji własność $EF_I\varphi$

Własność $EF_I\varphi$ będzie weryfikowana przy pomocy funkcji $EF(\varphi, I)$ oraz rekurencyjnej funkcji pomocniczej $EFRef(wezel, \varphi, I)$. Zadaniem funkcji pomocniczej jest sprawdzenie czy przez $wezel$ przechodzi ścieżka, mająca początek w węźle s_0, w której spełniona jest własność $F_I\varphi$ (spełniona φ w momencie z $I \cap wezel.Z$). Funkcja $EF(\varphi, I)$ zwraca prawdę, gdy własność $EF_I\varphi$ jest spełniona w badanym grafie SA_N (w stanie symbolicznym s_0).

Funkcja 6: Funkcja weryfikacji formuły $EF_I\varphi$

EF(φ, I):

```
1: return EFRef(s_0, φ, I);
```

Węzeł $s_0 = \overrightarrow{PostTd_d}(s'_0)$ gdzie s'_0 to początkowy stan symboliczny grafu SA_N.

Funkcja 7: Rekurencyjna funkcja pomocnicza $EFRef$

EFRef$(wezel, \varphi, I)$:

```
1: if I.b < −wezel.z_0s then
2:    return FALSE;
3: else if wezel.espel(φ, I) then
4:    return TRUE;
5: else if wezel.potomni ≠ ∅ then
6:    for w ∈ wezel.potomni do
7:       if EFRef(w, φ, I) then
8:          return TRUE;
9:       end if
10:   end for
11:   return FALSE;
12: else if wezel.ref ≠ ∅ then
13:   for w ∈ wezel.ref.potomni do
14:      if EFRef(w, φ, I.redukuj(wezel.ref.z_0g − wezel.z_0g)) then
15:         return TRUE;
16:      end if
17:   end for
18:   return FALSE;
19: else
20:   return FALSE;
21: end if
```

Procedura realizująca funkcję $EFRef$ w $wezel$ sprawdza pięć przypadków.

Po pierwsze jeżeli $wezel$ występuje później niż przedział I to procedura zwraca fałsz, ponieważ wszystkie fragmenty ścieżek wychodzące z tego węzła są nieistotne (linia 1-2).

Po drugie jeżeli formuła φ jest spełniona przy znakowaniu $wezel.m$ oraz ograniczenia zegara x_g zawierają moment z przedziału I to procedura zwraca prawdę (linie 3-4).

Po trzecie, gdy $wezel$ ma węzły potomne to należy sprawdzić czy przynajmniej przez jeden z tych węzłów przechodzi dalej ścieżka spełniająca własność $F_I\varphi$. Jeżeli tak, to procedura zwraca prawdę, a w przeciwnym razie fałsz (linie 5-11).

Po czwarte jeżeli wcześniejsze przypadki nie miały miejsca, to procedura sprawdza, czy węzeł $wezel$ ma aproksymację w SA_N (czy $wezel.ref \neq \emptyset$). Jeżeli tak to, procedura zwraca prawdę, gdy przez jeden z węzłów potomnych węzła $wezel.ref$ przechodzi ścieżka spełniającą własność $F_J\varphi$, przy czym $J = I.redukuj(wezel.ref.z_{0g} - wezel.z_{0g})$ (linie 12-17). W przeciwnym razie procedura zwraca fałsz (linia 18).

W pozostałych przypadkach procedura zwraca fałsz (linie 20).

5.1.2 Algorytm weryfikacji własność $EG_I\varphi$

Własność $EG_I\varphi$ jest weryfikowana przy pomocy funkcji $EG(\varphi, I)$ oraz rekurencyjnej funkcji pomocniczej $EGRef(wezel, \varphi, I)$. Procedura realizująca funkcję pomocniczą sprawdza czy przez $wezel$ przechodzi ścieżka, mająca początek w węźle s_0, dla której spełniona jest własność $G_I\varphi$. Procedura realizująca funkcję $EG(\varphi, I)$ zwraca prawdę, gdy własność $EG_I\varphi$ jest spełniona w badanym grafie SA_N.

Zakładamy, że inwarianty nie występują w N.

Funkcja 8: Funkcja weryfikacji formuły $EG_I\varphi$

EG(φ, I):
1: return $EFRef(s_0, \varphi, I)$;

Węzeł $s_0\overrightarrow{PostTd_d}(s'_0)$ gdzie s'_0 to początkowy stan symboliczny grafu SA_N.

Funkcja 9: Rekurencyjna funkcja pomocnicza $EGRef$

EGRef$(wezel, \varphi, I)$:

```
1: if −wezel.z_0g > I.a then
2:    return FALSE;
3: else if wezel.spel(φ) then
4:    return TRUE;
5: else if wezel.potomni ≠ ∅ then
6:    for w ∈ wezel.potomni do
7:       if EGRef(w, φ, I) then
8:          return TRUE;
9:       end if
10:   end for
11:   return FALSE;
12: else if wezel.ref ≠ ∅ then
13:   for w ∈ wezel.ref.potomni do
14:      if EGRef(w, φ, I.redukuj(wezel.ref.z_0g − wezel.z_0g)) then
15:         return TRUE;
16:      end if
17:   end for
18:   return FALSE;
19: else
20:   return FALSE;
21: end if
```

Procedura realizująca funkcę $EGRef$ testuje w węźle $wezel$ pięć przypadków.

Po pierwsze jeżeli najmniejsze możliwe wskazanie zegara x_g w $wezel$ jest większe niż $I.a$ to ani w tym węźle ani w węzłach potomnych badana własność nie będzie spełniona ponieważ węzeł występuje za późno ($-wezel.z_{0g} > I.a$). Należy zauważyć, że jeżeli własność byłaby spełniona dla rodzica tego węzła to już nie byłaby badana dla $wezel$ (dzięki założeniu wykluczenia inwariantów w modelu systemu). Wynika to z tzw. leniwych tranzycji w systemach c-TdPN. Procedura zwraca fałsz (linie 1-2).

W przeciwnym razie ($-wezel.z_{0s} \leqslant I.a$) jeżeli znakowanie $wezel.m$ spełnia formułę φ to znaczy, że przez $wezel$ przechodzi szukana ścieżka, stąd procedura zwraca prawdę (linie 3-4).

Gdy wcześniejsze przypadki nie są spełnione oraz $wezel$ ma węzły potomne, to do zwrócenia prawdy przez procedurę wystarczy aby przez jeden z węzłów potomnych w przechodziła ścieżka spełniająca własność $G_I\varphi$. W przeciwnym razie procedura zwraca fałsz (linie 5-11).

Po czwarte procedura sprawdza czy $wezel$ ma aproksymację. Jeżeli tak, to procedura zwraca prawdę, gdy przynajmniej przez jeden z węzłów potomnych węzła $wezel.ref$ przechodzi ścieżka w której spełniona jest własność $G_J\varphi$, gdzie
$J = I.redukuj(wezel.ref.z_{0g} - wezel.z_{0g})$ (linie 12-17). W przeciwnym razie fałsz (linia 18).

W końcu, gdy w żadnym z powyższych przypadków wywołanie nie zakończyło się powodzeniem procedura zwraca fałsz ponieważ od węzła *wezel* nie ma możliwości, aby jakaś ścieżka spełniła badaną własność (linia 20).

5.1.3 Algorytm weryfikacji własności $AF_I\varphi$ i $AG_I\varphi$

Własności $AF_I\varphi$ i $AG_I\varphi$ są weryfikowane przy pomocy funkcji $AF(\varphi, I)$ i $AG(\varphi, I)$. Obie funkcje używają odpowiednio funkcji $EGRef$ i $EFRef$.

Funkcja 10: Funcja weryfikacji formuły $AF_I\varphi$

AF(φ, I):
1: return $\neg EGRef(s_0, \neg\varphi, I)$

Własność $AF_I\varphi$ jest spełniona, jeżeli procedura realizująca funkcję $EGRef(s_0, \neg\varphi, I)$ w wyniku działania zwróci fałsz.

Funkcja 11: Funcja weryfikacji formuły $AG_I\varphi$

AG(φ, I):
1: return $\neg EFRef(s_0, \neg\varphi, I)$

Podobnie własność $AG_I\varphi$ jest spełniona, jeżeli procedura realizująca funkcję $EFRef(s_0, \neg\varphi, I)$ w wyniku działania zwróci fałsz.

5.1.4 Algorytm weryfikacji własności ograniczonej reakcji

Złożona własność $AG_I\varphi \rightarrow EF_J\psi$ pozwala na weryfikowanie wystąpienia stanu w którym spełniona jest formuła ψ, spowodowanego wystąpieniem stanu spełniającego formułę φ. Własność $AG_I\varphi \rightarrow EF_J\psi$ jest spełniona w SA_N jeżeli procedura realizujące funkcję $AGsEF(\varphi, \psi, I, J)$ zwraca prawdę. Procedura realizująca pomocniczą funcję rekurencyjną $AGsEFRef(we, \varphi, \psi, I, J)$ zwraca prawdę, jeżeli przez węzeł we przechodzą tylko ścieżki, mające początek w s_0 dla których spełniona jest własność $G_I\varphi \rightarrow EF_J\psi$.

Założono, że $J.b \neq \infty$.

Funkcja 12: Funkcja weryfikująca formułę $AG_I\varphi \rightarrow EF_J\psi$

AGsEF(φ, ψ, I, J):
1: return AGsEFRef$(s_0, \varphi, \psi, I, J)$;

Funkcja 13: Rekurencyjna funkcja pomocnicza $AGsEFRef$

AGsEFRef$(we, \varphi, \psi, I, J)$:

```
1: if −we.z_0g > I.b then
2:   return TRUE;
3: end if
4: if we.spel(φ) then
5:   if −we.z_0g ⩾ I.a ∧ ¬EFRef(we, ψ, K_1) then
6:     return FALSE;
7:   end if
8:   if −we.z_0g < I.b ∧ ¬EFRef(we, ψ, K_2) then
9:     return FALSE;
10:  end if
11: end if
12: if we.potomni ≠ ∅ then
13:   for w ∈ we.potomni do
14:     if ¬AGsEFRef(w, φ, ψ, I, J) then
15:       return FALSE;
16:     end if
17:   end for
18: end if
19: if we.ref ≠ ∅ then
20:   for w ∈ we.ref.potomni do
21:     if ¬AGsEFRef(w, φ, ψ, I.redukuj(wezel.ref.z_0g − wezel.z_0g)), J) then
22:       return FALSE;
23:     end if
24:   end for
25: end if
26:
27: return TRUE;
```

Procedura realizująca funkcję $AGsEFRef$ sprawdza niezależnie trzy przypadki.

Po pierwsze jeżeli najwcześniejsze wskazanie zegara x_g w we jest późniejsze niż moment $I.b$ to procedura zwraca prawdę, gdyż w we albo dla jego potomnych formuła φ nie ma szans na spełnienie w przedziale I, a przez to spowodowanie przymusu spełnienia ψ (linie 1-3).

Po drugie procedura zwraca fałsz, gdy formuła φ jest spełniona w we, oraz:

- przekrój przedziału I i ograniczeń dla zegara x_g jest niepusty ($I \cap we.Z \neq \emptyset$) i nie istnieje ścieżka przechodząca przez we która w przedziale K_1 spełnia ψ, gdzie $K_1 = [J.a - we.z_{0s}, J.b + I.b]$ (linie 5-6),
- wskazanie x_s jest nie później niż moment $I.b$ i nie istnieje ścieżka przechodząca przez we, która w przedziale K_2 spełnia ψ, $K_2 = [J.a + I.a, J.b + I.b]$ (linie 8-9).

Po trzecie jeżeli we ma węzły potomne to wystarczy, aby jeden z tych węzłów nie spełnił własności $AG_I\varphi \rightarrow EF_J\psi$, żeby procedura zwróciła fałsz.

Po czwarte jeżeli we ma aproksymację w SA_N to dla potomnych węzła $we.ref$ sprawdzana jest badana własność ze zmodyfikowanym przedziałem $I = I.redukuj(wezel.ref.z_{0s} - wezel.z_{0s})$. Gdy własność nie jest spełniona dla przynajmniej jednego węzła potomnego to procedura zwraca $false$. W przeciwnym razie $true$.

W przeciwnym razie procedura zwraca prawdę ponieważ nie ma innych przypadków spełnienia formuły φ.

Kolejna formuła z mocniejszym założeniem co do nagłówka implikacji $AG_I\varphi \rightarrow AF_J\psi$ jest weryfikowana przez funkcję $AGsAFRef(we, \varphi, \psi, I, J)$. Funkcja została zdefiniowana analogicznie jak AGsEFRef za wyjątkiem zmiany drugiego warunku w linni 5 na $\neg EGRef(we, \psi, K_1)$, drugiego warunku w linii 8 na $\neg EGRef(we, \psi, K_2)$ oraz warunku w linii 14 na $\neg AGsAFRef(w, \varphi, \psi, I, J)$, a w linii 21 na $\neg AGsAFRef(w, \varphi, \psi, I.redukuj(wezel.ref.z_{0s} - wezel.z_{0s}), J)$.

5.2 Analiza algorytmu generującego graf stref (TTS)

Przeprowadzono analizę złożonosci czasowej opracowanych algorytmów w kontekscie zastosowań. Jako operacje jednostkowe wybrano wszystkie operacje arytmetyczne oraz operacje porównania i przypisania.

Złożoność czasową pesymistyczną dla algorytmu nr 5 przy zastosowaniu aproksymacji k_x zbadano w zależnosci od liczby miejsc n, liczby tranzycji m oraz liczby k wykonanych iteracji pętli głównej algorytmu (liczby przeanalizowanych węzłów przez algorytm).

Zainicjowanie algorytmu, na co składa się obliczenie stanu s_0 na podstawie stanu s'_0 i dodanie s_0 do zbioru $wait$, ma złożoność liniową $O(n)$.

Założono, że pętla główna jest wykonywana co najwyżej k razy. Oszacowane operacje jednostkowe wykonane w jednej iteracji pętli należy więc przemnożyć przez k.

W obrębie jednej iteracji pętli głównej należy wykonać sprawdzenie $wait \neq \phi$ oraz przypisanie z usunięciem $s = pop(wait)$, a więc złożoność $O(1)$. Następnie sprawdzenie aproksymacji k_x dla węzła s potrzebuje złożoności $O(n^2 \cdot i^2)$, która jest zależna od numeru i iteracji pętli głównej. Dokładnie jest to liczba elementów w zbiorze $wait$ ponieważ strefa każdego węzła ze zbioru $wait$ jest porównywana ze strefą węzła s. Parametr i jest ograniczony przez $1 \geqslant i \geqslant k$. Wybór czynnej tranzycji do uruchomienia ma złożoność $O(m \cdot n)$. Pętla for może mieć co najwyżej m obiegów. Wyznaczenie stanu symbolicznego $PostTd_t(s)$ oraz wyznaczenie stanu symbolicznego s' ma złożoność $O(n^3)$. Podobnie jak w [9] najbardziej kosztowną operacją podczas generowania stanu symbolicznego jest obliczenie postaci kanonicznej przy użyciu algorytmu najkrótszej scieżki *Floyda-Warshalla*. Pozostałe operacje mają złożoność $O(1)$.

Po wykonaniu pętli for stan s zostaje dodany do zbioru $wait$ ze złożonością $O(1)$.

Podsumowując pesymistyczny koszt wygenerowania grafu SA_N w k iteracjach jest rzędu

$$O(n) + O(k) + O(k^3 \cdot n^2) + O(k \cdot m \cdot n^3) + O(k \cdot n)$$

Funkcja H reprezentuje pesymistyczną złożoność wygenerowania reprezentatywnego fragmentu:

$$H(k,m,n) = \frac{2k}{3}(k^2-1)(n^2+n+1) + k \cdot (m \cdot (3n^3+13n^2+22n+10)+6) + n + 1$$

gdzie k to liczba iteracji pętli głównej (liczba analizowanych węzłów), m i n to liczba tranzycji i miejsc modelu systemu.

Interpretacja wyników dla kolejnych k na skali logarytmicznej pokazała, że złożoność przy ustalonej liczbie miejsc i tranzycji, które są ograniczone, nie przekracza $O(exp(k))$. Ponadto ponieważ algorytm zapewnia zatrzymanie w pewnym momencie, to liczba k musi być także ograniczona. Potwierdza to możliwość stosowania algorytmu w praktyce przy racjonalych wartosciach k, co także sprawdzono na konkretnych przykładach przy pomocy opracowanej biblioteki programistycznej.

5.3 Analiza algorytmów weryfikacji własności $EF_I\varphi$, $EG_I\varphi$, $AF_I\varphi$, $AG_I\varphi$

Struktura SA_N może być traktowana jako drzewo wielopoziomowe zwane *drzewem bazowym dla* SA_N którego wierzchołkami są elementy zbioru QA, krawędziami elementy zbioru $\Rightarrow$, a korzeniem $s_0 = \overrightarrow{PostTd_d}(s'_0)$ gdzie $s'_0 \in QA_0$.

Poddrzewem referencyjnym nazwiemy drzewo mające korzeń w dowolnym liściu drzewa bazowego, którego relacja $\Rightarrow$ jest rozszerzona o relację aproksymacji pomiędzy jego węzłami. Formalnie $(s,s') \in \Rightarrow_{ref}$ wtw, gdy:

1. $(s,s') \in \Rightarrow$ lub
2. s to liść, s jest aproksymowane przez s'' i $(s'',s') \in \Rightarrow$.

Analiza złożonosci algorytmów weryfikujących własności wykorzystuje dwie heurystyki:

- h to wysokość drzewa bazowego. Wartość tą można wyznaczyć raz po wygenerowaniu grafu stref prostym algorytmem rekurencyjnym.
- u to maksymalna wysokość poddrzewa referencyjnego wyznaczona w oparcu o prawy koniec przedziału I. Wartość tą można wyznaczyć jako różnicę $u = u_1 - h$. Wartość u_1 reprezentuje wysokość drzewa bazowego wyznaczonego do co najwyżej węzła, którego strefa ma niepuste przecięcie z przedziałem $[0, I.b]$, oraz w którym traktujemy relację $\Rightarrow$ jako $\Rightarrow_{ref}$.

Suma $h+u$ jest więc wysokością fragmentu drzewa bazowego, powstałego bez używania aproksymacji, jaki jest potrzebny do weryfikowania własności do momentu $I.b$.

Złożoność obliczeniową pesymistyczną funkcji EFRef, bez uwzględnienia aproksymacji oraz przy znajomości wysokości h drzewa SA_N można przybliżyć wzorem rekurencyjnym $T(h) = 4 + m \cdot T(h-1)$ dla $h > 0$ oraz $T(h) = 4$ dla $h = 0$. Wartość 4 to

liczba operacji sprawdzenia warunków w funkcji, a $m \cdot T(h-1)$ to koszt m-krotnego wywołania funkcji T dla każdego węzła potomnego.

Ponieważ niektóre liście drzewa bazowego są aproksymowanie z jego węzłami to dla $h = 0$ funkcja T została zmodyfikowana do postaci $T(0) = T^r(u) = 4 + m \cdot T^r(u-1)$ dla $u > 0$ oraz $T^r(u) = 1$ dla $u = 0$.

Zatem złożoność weryfikacji własności $EF_I\varphi$ wyraża się wzorem rekurencyjnym:

$$T_{EF}(h,u) = \begin{cases} 4 + m \cdot T_{EF}(h-1,u) & gdy \quad h > 0 \\ 4 + T^r_{EF}(u) & gdy \quad h = 0 \end{cases} \tag{1}$$

,gdzie

$$T^r_{EF}(u) = \begin{cases} 4 + m \cdot T^r_{EF}(u-1) & gdy \quad u > 0 \\ 1 & gdy \quad u = 0 \end{cases} \tag{2}$$

Do rozwiązania rekurencji zastosowano metodę iteracji. Pesymistyczna złożonosć obliczeniowa weryfikacji własności $EF_I\varphi$ wyraża się wzorem

$$F(h,m,u) = 4\frac{1-m^{h+u+1}}{1-m} \tag{3}$$

gdzie h to wysokość drzewa, m to liczba tranzycji weryfikowanego modelu systemu, a u to wysokość poddrzewa referencyjnego, którego elementy mają część wspólną ograniczeń zegara x_g z przedziałem $[0, I.b]$.

Funkcja F ma złożoność pesymistyczną co najwyżej $O(m^{h+u})$. Tak więc przewiduje się, że w pesymistycznym przypadku w praktyce dla dużych wartości m, h i u weryfikacja może trwać zbyt długo.

Rząd złożoności pesymistycznej weryfikacji formuły $EG_I\varphi$ jest analogiczny co $EF_I\varphi$. Ponieważ weryfikacja pozostałych formuł $AF_I\varphi$ i $AG_I\varphi$ opiera się na negacji $EG_I\varphi$ i $EF_I\varphi$ to ich rząd pesymistycznej złożoności będzie także taki sam.

5.4 Analiza algorytmów weryfikacji własności ograniczonej reakcji

Do zbadania złożonosci algorytmu weryfikującego własność $AG_I\varphi \rightarrow EF_J\psi$ wykorzystuje się trzy heurystyki:

h - wysokość drzewa bazowego,

u - wysokość poddrzewa rekurencyjnego w obrębie przedziału $[0, I.b]$ dla zegara x_g,

w - wysokość poddrzewa rekurencyjnego wyznaczającego pokrycie przedziału $[0, I.b + J.b]$ dla zegara x_g w celu weryfikacji wełasnosci $EF_I\psi$.

Obliczeniowo wartości h i u należy wyznaczyć jak przy analizie własnosci $EF_I\varphi$. Natomiat wartosć parametru w analogicznie jak przy u z tą różnicą, że do przedziału $[0, I.b + J.b]$, aby sprawdzić jak głęboko w pesymistycznym przypadku może być weryfikowana własność $EF_J\psi$.

Złożoność pesymistyczna weryfikacji własności $AG_I\varphi \rightarrow EF_I\psi$ wyraża się wzorem rekurencyjnym:

$$T_{AGpEF}(h,u,w) = \begin{cases} 8 + 2T_{EF}(h,w) + m \cdot (T_{AGpEF}(h-1,u,w)+1) & gdy \quad h > 0 \\ 8 + 2T_{EF}(h,w) + m \cdot (T^r_{AGpEF}(u,w)+1) & gdy \quad h = 0 \end{cases}$$

,gdzie

$$T^r_{AGpEF}(u,w) = \begin{cases} 8 + 2T_{EF}(u,w) + m \cdot (T^r_{AGpEF}(u-1,w)+1) & gdy \quad u > 0 \\ 1 & gdy \quad u = 0. \end{cases}$$

Natomiast po rozwiązaniu rekurencji metodą iteracji, pesymistyczna złożoność obliczeniowa weryfikacji formuły wyraża się wzorem:

$$G(h,m,u,w) = \frac{8um^{w+h+u+3}}{m-1} + \frac{8hm^{w+h+2}}{m-1} + \frac{8m^{w+h+2}}{m-1} + \frac{9m^{h+u+1}}{m-1}$$
$$-\frac{8m^{h+1}}{(m-1)^2} - \frac{8m^{u+1}}{(m-1)^2} - \frac{8m^{2u+1}}{(m-1)^2} - \frac{m}{m-1} - \frac{8}{m-1} + \frac{8}{(m-1)^2}$$

Jest to pesymistyczna złożoność $O(u{\cdot}m^{w+h+u+2})$, gdzie m oznacza liczbę tranzycji, h to wysokość drzewa bazowego, u to maksymalna wysokość pseudodrzewa referencyjnego którego elementy mają część wspólną zegara x_s z przedziałem $[0, I.b]$, oraz analogicznie w dla $[0, I.b + J.b]$.

Podobnie jak dla funkcji F przewiduje się, że w pesymistycznym przypadku w praktyce dla dużych wartości m, h, u i w weryfikacja może trwać zbyt długo.

6 Badania symulacyjne i weryfikacja własności wybranego modelu systemu c-TdPN

Analiza opracowanych algorytmów pokazała, że możliwa jest implementacja i stosowanie w praktyce opracowanej metody dla racjonalnych parametrów określających liczbę miejsc i tranzycji w modelu systemu.

Jako narzędzie symulacji autor wybrał język obiektowy Java 9, oprogramowanie NetBeans IDE 10.0 oraz program VisualVM służący do monitorowania procesów. Obróbka wyników została przygotowana przy pomocy Apache OpenOffice 4.1.5.

Implementacja opracowanej metody, zwana TdPNBib (w skr. Td), stanowi zbiór klas reprezentujących model sieci c-TdPN, strukturę danych reprezentującą przestrzeń stanów modelu c-TdPN oraz weryfikator własności TdPN-TCTL. Całość jest zamknięta w pakiet z załączoną dokumentacją do każdej z klas.

W badaniach porównuje się otrzymane rezultaty z istniejącym rozwiazaniem TAPAAL [13, 29] (w skr. TA) w wersji 3.4.3-win64 (tapaal.net). Narzędzie TAPAAL generuje fragment zachowania wystarczający tylko na potrzeby weryfikacji danej formuły.

Wybrano 4 modele systemów, które krótko scharakteryzowano, przeprowadzono badania przy wykorzystaniu zaimplementowanej metody oraz porównano z istniejącym rozwiązaniem. Każdy model systemu jest rozszerzalny pod względem wybranego parametru co pozwala na skalowanie rozmiaru modelu systemu w trakcie badania.

Dla przejrzystości prezentacji badań autor przedstawia rezultaty dla jednego modelu systemu. Wyniki badań o podobnej formie dla pozostałych systemów umieszczono w załącznikach.

Kolejno przedstawiono opis wybranego modelu systemu synchronizacji dostępu do danych. Opis zawiera cztery warianty rozważanego modelu systemu.

Następnie przedstawiono opis przeprowadzonych badań symulacyjnych:

- Analiza wybranych własności opisanych w logice TdPN-TCTL w poszczególnych wariantach modelu systemu.

- Analiza przydatności miejsc odczytu.

- Analiza porównawcza reprezentatywnych przestrzeni stanów wygenerowanych przez TdPNBib i TAPAAL.

- Badanie eksperymentalne własności $EF_I\varphi$ algorytmu konstruowania modelu TTS z dodatkowym warunkiem momentu stopu.

- Wpływ wybranej aproksymacji na weryfikację własności modelu systemu.

- Analiza wpływu wzrostu parametrów modelu systemu na środowisko symulacyjne do konstruowania reprezentatywnego TTS.

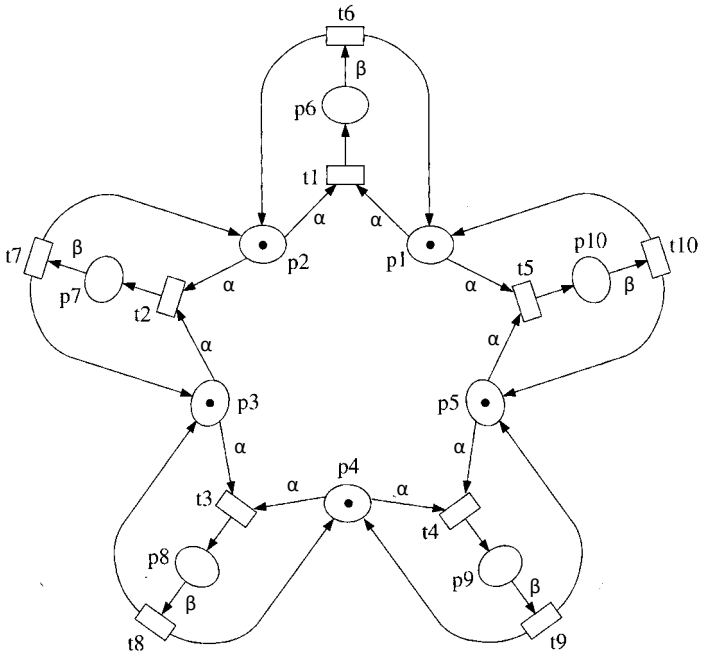
W badaniach złożono ograniczenie co do najwyżej jednego żetonu w miejscu ze względu na rozwiązanie TdPNBib. Weryfikowane formuły w obu rozwiązaniach nie zawierają zagnieżdżeń.

Badania wykonano na komputerach desktopowych:

KOM1 - procesor Intel(R) Core(TM) i7-4700MQ 2.40GHz, 8GB RAM,

KOM2 - procesor Intel(R) Core(TM) i5-4570S 2.90GHz, 16GB RAM.

6.1 Systemy z synchronizacją dostępu do danych z dodatkowymi ograniczeniami



Rysunek 14: Model systemu synchonizacji dostępu do zasobów $N = 5$.
źródło: na podstawie [45] str. 41

Problem ucztujących filozofów jest klasycznym problemem synchronizacji dostępu do danych. Problem ten modelowano w pracy [22] przy zastosowaniu algorytmu [48], [49], a wyniki przedstawiono w prezentacji *„Model of rational agent on the electronic service market"* na warsztatach konferencji IWZ/CMEE 2017 UMCS w Lublinie.

Model klasycznie obejmuje pięciu filozofów ($N = 5$). Liczbę uczonych N można zmieniać w modelu systemu. Każdy i-ty uczony ($i = 1, 2, \ldots, 5$) potrzebuje zestawu zasobów reprezentowanych przez parę miejsc P_i i P_{i+1} dla $i = 1, 2, 3, 4$ oraz P_5 i P_1 dla $i = 5$, aby wykonać akcję wejścia do sekcji krytycznej reprezentowaną przez tranzycje t_i. Akcja zwolnienia zasobów przez i-tego uczonego jest reprezentowana przez tranzycję t_{i+5}. Każdy zasób od momentu pojawienia się w miejscu dostępności może być użyty nie wcześniej niż po upływie $\min \alpha$ jednostek czasu, oraz nie później niż do $\max \alpha$. Uczony może wyjść z sekcji krytycznej nie wcześniej niż po upływie $\min \beta$ od momentu wejścia oraz nie później niż $\max \beta$.

Do tak zdefiniowanego bazowego modelu systemu z parametrem N dodaje się dwa opcjonalne ograniczenia:

(1) Dodaje się inwarianty do miejsc reprezentujących zasoby $inv(x_i) \leqslant \max \alpha$ dla

$i = 1, 2, \ldots, 5$ oraz do miejsc reprezentujących przebywanie w sekcji krytycznej $inv(x_i) \leqslant \max \beta$ dla $i = 6, 7, \ldots, 10$. Symbol x_i oznacza zegar przypisany do miejsca p_i, który reprezentuje upływ czasu od momentu pojawienia się w nim żetonu.

(2) Dodaje się miejsca P_{2N+j} (dla $j = 1, 2, 3, 4, 5$, zawierające po jednym żetonie) wymuszające co najwyżej jednokrotne przeprowadzenie akcji krytycznej przez uczonych. Z każdego miejsca P_{2N+j} wychodzi łuk do tranzycji t_j.

Rozważane dalej warianty zestawiono w Tabeli nr 1.

Tabela 1: Rozważane warianty modelu systemu synchronizacji dostępu do danych

Lp.	Warianty	Opis
1	nr 1	Model bazowy
2	nr 2	Model bazowy z (2)
3	nr 3	Model bazowy z (1)
4	nr 4	Model bazowy z (1) i (2)

6.2 Analiza wybranych własności opisanych w TdPN-TCTL

Do badania wybrano interesujące własności do weryfikacji w modelu systemu w różnych jego wariantach. W badaniu używano komputera KOM1.

Dla typów własności EF,EG,AF i AG możliwe jest badanie stanów reprezentowanych przez formuły φ_i gdzie:

1. Dwóch wybranych konfliktowych uczonych będzie w sekcji krytycznej przeprowadzać akcję krytyczną (φ_1).
2. Tylko jeden wybrany uczony wykona akcję krytyczną (φ_2).
3. Każdy uczony wykona akcję krytyczną tylko raz (φ_3).
4. Wybrany uczony będzie zarówno gotowy do przeprowadzenia akcji krytycznej i będzie w trakcie przeprowadzania akcji krytycznej (φ_4).

W badaniu sprawdza się czy własności $EF_I\varphi_i$,$EG_I\varphi_i$, $AF_I\varphi_i$ i $AG_I\varphi_i$ są spełnione w wybranym wariancie modelu systemu, dla $i = 1, 2, 3, 4$, $N = 2, 3, 4, 5, 6$ oraz $I = [0, \infty)$. W badaniach użyto aproksymację k_x.

6.2.1 Wariant nr 1 modelu systemu

Teoretycznie jedynie własności $EF_I\varphi_2$ i $EF_I\varphi_3$ są spełnione w badanym wariancie nr 1 modelu systemu. Ponadto ze względu na nierozróżnialność formuł φ_2 i φ_3 od formuły reprezentującej stan początkowy modelowanego systemu, własności $EG_I\varphi_j$ i $AF_I\varphi_j$ dla $j = 2, 3$ także są spełnione w wybranym wariancie. Natomiast teoretycznie pozostałe własności nie są spełnione.

Tabela 2: Weryfikacja własności typu $EF_I\varphi_i$ przy pomocy TdPNBib, w zależności od N, przedziału $I = [0, 10]$, w wariancie nr 1 modelu systemu.

N	$EF_I\varphi_1$	$EF_I\varphi_2$	$EF_I\varphi_3$	$EF_I\varphi_4$
2	false	true	true	false
3	false	true	true	false
4	false	true	true	false
5	false	true	true	false
6	?	?	?	?

Ponadto symulacyjna weryfikacja własności $EG_I\varphi_i$ i $AF_I\varphi_i$ przyniosła taki sam efekt jak w Tabeli nr 2. Natomiast weryfikowane własności $AG_I\varphi_i$ nie były spełnione w badanym modelu.

Badanie trwało nie więcej niż dwie godziny. W tym czasie uzyskano co najmniej wszystkie wyniki dla $N = 2, 3, 4, 5$. Badanie symulacyjne potwierdziło rozważania teoretyczne przez symulację dla $N < 6$. Wyjątkowo w tym badaniu użyto komputera KOM2.

6.2.2 Wariant nr 2 modelu systemu

Dodanie dodatkowych miejsc do wariantu bazowego (wariancie nr 2) modelu systemu powoduje możliwość identyfikacji stanu systemu po tym jak każdy z uczonych wykona akcję krytyczną.

Teoretycznie jedynie własności $EF_I\varphi_2$ i $EF_I\varphi_3$ są spełnione w wariancie nr 2.

Tabela 3: Weryfikacja własności typu $EF_I\varphi_i$ przy pomocy TdPNBib, w zależności od N, przedziału $I = [0, \infty]$, w wariancie nr 2 modelu systemu.

N	$EF_I\varphi_1$	$EF_I\varphi_2$	$EF_I\varphi_3$	$EF_I\varphi_4$
2	false	true	true	false
3	false	true	true	false
4	false	true	true	false
5	false	true	true	false
6	false	true	true	false

Ponadto symulacyjna weryfikacja pozostałych własności $EG_I\varphi_i$, $AF_I\varphi_i$ i $AG_I\varphi_i$, dla takich samych wartości parametrów I i N, w wyniku zwróciła fałsz.

Czas generowania wyników tabeli nie przekroczył 8 minut. Badania symulacyjne potwierdziły rozważania teoretyczne.

6.2.3 Wariant nr 3 modelu systemu

Obecność inwariantów nie pozwala na postęp czasu w modelu systemu gdy obecność żetonu w miejscach P_i dla $i = 1, 2, 3, ..., N$ osiągnie maksimum ($inv(x_{P_i}) = x_{P_i}$). Dzięki

temu nie jest możliwe, aby model systemu podczas działania nie wykonał żadnej akcji (nie uruchomił żadnej tranzycji).

Teoretycznie jedynie własności $EF_I\varphi_2$ i $EF_I\varphi_3$ są spełnione w badanym wariancie nr 3 modelu systemu. Ponadto podobnie jak w wariancie nr 1 własności $EG_I\varphi_j$ i $AF_I\varphi_j$ dla $j = 2, 3$ także są spełnione w wybranym wariancie. Natomiast teoretycznie pozostałe własności nie są spełnione.

Tabela 4: Weryfikacja własności typu $EF_I\varphi_i$ przy pomocy TdPNBib, w zależności od N, przedziału $I = [0, 10]$, w wariancie nr 3 modelu systemu.

N	φ_1	φ_2	φ_3	φ_4
2	false	true	true	false
3	false	true	true	false
4	false	true	true	false
5	false	true	true	false
6	-	-	-	-

Czas generowania wyników dla $N < 6$ nie przekroczył 3 minut. Natomiast dla $N = 6$ czas badania przeciągał się. Badania symulacyjne potwierdziły rozważania teoretyczne dla $N < 6$.

6.2.4 Wariant nr 4 modelu systemu

Teoretycznie jedynie własności $EF_I\varphi_2$ i $EF_I\varphi_3$ są spełnione w wariancie nr 4.

Tabela 5: Weryfikacja własności typu $EF_I\varphi_i$ przy pomocy TdPNBib, w zależności od N, przedziału $I = [0, \infty]$, w wariancie nr 4 modelu systemu.

N	$EF_I\varphi_1$	$EF_I\varphi_2$	$EF_I\varphi_3$	$EF_I\varphi_4$
2	false	true	true	false
3	false	true	true	false
4	false	true	true	false
5	false	true	true	false
6	false	true	true	false

Ponadto symulacyjna weryfikacja pozostałych własności $EG_I\varphi_i$, $AF_I\varphi_i$ i $AG_I\varphi_i$, dla takich samych wartości parametrów I i N, w wyniku zwróciła fałsz. Czas generowania wyników do tabeli także nie przekroczył 8 minut. Badania symulacyjne ponownie potwierdziły rozważania teoretyczne.

6.3 Analiza przydatności miejsc odczytu

Badanie ma na celu sprawdzenie w jakim stopniu używanie miejsc odczytu polepsza pracę z modelem systemu i jego modelem zachowania TTS w stosunku do zamiany

łuków czytania na pętle zwykłych łuków. Przewiduje się w badaniu, że różnica nie będzie tak znacząca jak przy modelu zachowania rozgałęzionego procesu czasowego (RPC). W badaniu użyto komputera KOM1 i KOM2.

W związku z tym opracowano dwie modyfikacje wariantu nr 2 modelu systemu w ten sposób, że:

$M1$ Dodano do modelu systemu miejsca odczytu dla każdej tranzycji, która reprezentuje wejście do sekcji krytycznej.

$M2$ Dodano do modelu systemu po jednym miejscu p dla każdej tranzycji t reprezentującej wejście uczonego do sekcji krytycznej. Para elementów p i t jest połączona pętlą łuków w ten sposób, że z miejsca p wychodzi jeden zwykły łuk do t, a drugi wraca z t do p.

Dla obu modyfikacji $M1$ i $M2$ wykonano próby oraz sprawdzono czas generowania i weryfikacji zbioru własności dla każdej próby. Odnotowano także potrzebną pamięć na przechowanie generowanych węzłów danej próby.

Na jedną próbę składa się weryfikacja wybranego zbioru formuł dla wybranego wariantu systemu ($M1$ lub $M2$) skalowanego parametrem $N \in \{2, 3, 4, 5, 6\}$. Wybranym zbiorem formuł może być czteroelementowy zbiór $\{EF_I\varphi_i\}$, $\{EG_I\varphi_i\}$, $\{AF_I\varphi_i\}$ lub $\{AG_I\varphi_i\}$, dla $i = 1, 2, 3, 4$ i $I = [0, \infty]$.

Tabela 6: Porównanie modyfikacji $M1$ i $M2$, wygenerowane przy pomocy KOM1

Własność	$EF_I\varphi_i$			$EG_I\varphi_i$		
Model systemu	**M1**	**M2**	ΔM	**M1**	**M2**	ΔM
Czas średni [s]	756,66	761,62	4,96	752,00	754,95	2,95
Własność	$AF_I\varphi_i$			$AG_I\varphi_i$		
Model systemu	**M1**	**M2**	ΔM	**M1**	**M2**	ΔM
Czas średni [s]	748,83	750,73	1,90	744,80	751,18	6,38

Tabela 7: Porównanie modyfikacji $M1$ i $M2$, wygenerowane przy pomocy KOM2

Własność	$EF_I\varphi_i$			$EG_I\varphi_i$		
Model systemu	**M1**	**M2**	ΔM	**M1**	**M2**	ΔM
Czas średni [s]	549,05	559,74	10,69	550,98	559,07	8,08
Własność	$AF_I\varphi_i$			$AG_I\varphi_i$		
Model systemu	**M1**	**M2**	ΔM	**M1**	**M2**	ΔM
Czas średni [s]	558,64	554,12	-4,52	559,09	560,78	1,69

Średni czas przeprowadzenia badania w każdym porównaniu okazał się krótszy dla modelu systemu, w którym użyto łuków czytania. Wyjątkiem jest formuła typu AF przy zastosowaniu KOM2. Ponadto największa różnica w średnim czasie, na obu komputerach KOM1 i KOM2, przypada dla własności typu EF. Jest to ponad 10 sekund dla modelu systemu z zastosowaniem miejsc odczytu.

Maksymalny rozmiar pamięci jaki został wykorzystany na przechowanie generowanych węzłów w obrębie jednej próby to nie więcej niż 15,5MB.

Po przeprowadzonym badaniu stwierdza się, że dodanie miejsc odczytu do modelu ma następujące zalety:

- szybkość kodowania modelu systemu,
- zwięzłość modelu systemu,
- przejrzystość na schemacie modelu systemu,
- tranzycja korzystająca z miejsca odczytu nie resetuje wieku żetonu,
- modelowanie akcji czytania systemu nie wymaga użycia łuków transportu.

Otrzymane rezultaty potwierdzają sensowność stosowania miejsc odczytu z punktu widzenia zużycia zasobów systemowych.

6.4 Analiza porównawcza reprezentatywnych przestrzeni stanów (TAPAAL)

W badaniu porównywane są narzędzia TAPAAL i TdPNBib (w skr. TA i Td) podczas szukania odpowiedzi na pytania:

- *Jaki jest czas weryfikowania niemożliwej do spełnienia formuły?*
- *Jak dużą reprezentatywną przestrzeń stanów należy wygenerować, aby zweryfikować niemożliwą do spełnienia formułę?*

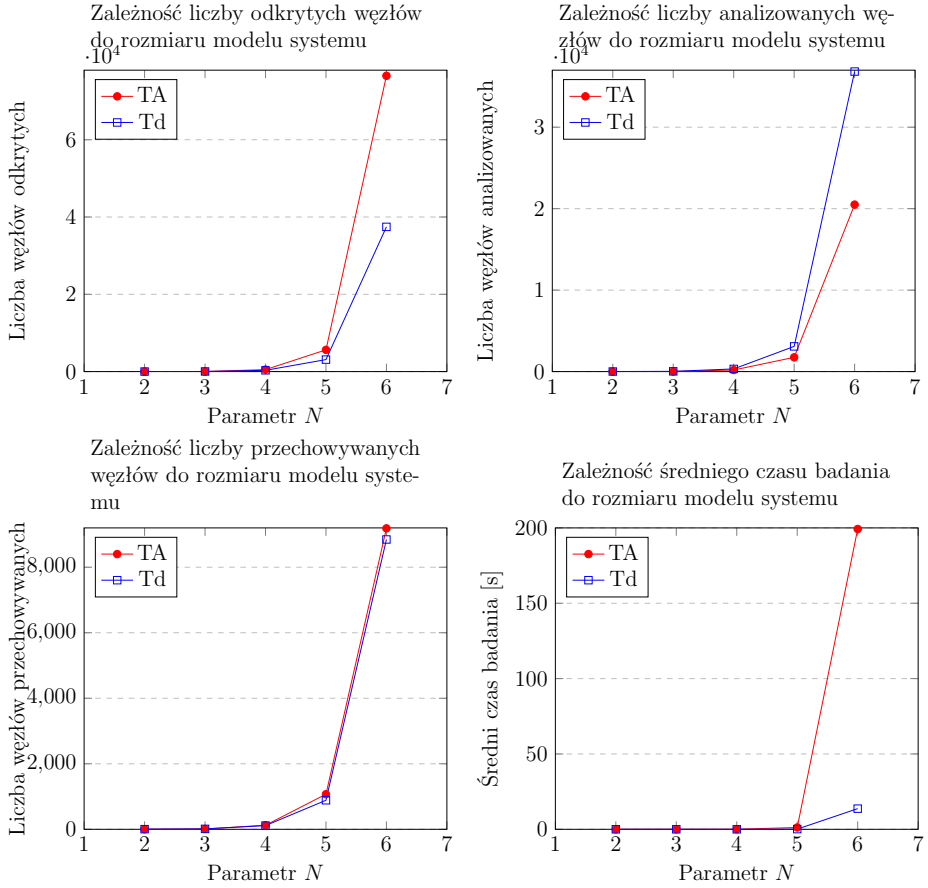
Wykonano test sprawdzający istnienie znakowania, dla którego mamy pewność, że nie istnieje w zachowaniu. Celem takiego badania jest porównanie TAPAAL i TdPNBib podczas sprawdzania czy dane znakowanie (które nie jest możliwe) jest w zachowaniu. Powodem tego jest wymuszenie, aby TAPAAL także wygenerował reprezentatywny fragment przestrzeni stanów. Badanie wykonano dla wariantu nr 1 modelu systemu.

W proponowanym rozwiązaniu (Td) wykorzystano metodę przeglądania w głąb przestrzeni stanów oraz aproksymację LU. W programie TAPAAL wybrano weryfikację przy pomocy silnika Continous Engine verifytapn, w konfiguracji: Search strategy Options: Heuristic Search, Use symmetry reduction, Approximation option: Exact analysis. Powodem wyboru takiej konfiguracji były ustawienia najbardziej zbliżone w obu rozwiązaniach. Obliczenia wykonano na komputerze KOM1.

Tabela 8: Zestawienie informacji o liczbie węzłów po wygenerowaniu przez TA i Td reprezentatywnego fragmentu przestrzeni stanów

	A		B		C		D	
	Odkryte		Analizowane		Przechowywane		Średni czas [s]	
N	TA	Td	TA	Td	TA	Td	TA	Td
2	5	7	3	7	3	4	0,025	0
3	19	28	10	28	10	13	0,026	0
4	509	325	193	325	124	113	0,050	0,003
5	5621	3101	1746	3091	1071	886	1,203	0,124
6	76549	37449	20497	36831	9186	8846	199,27	13,706

Kolumny A,B,C i D odpowiadają kolejno liczbie odkrytych podczas badania węzłów, liczbie przebadanych węzłów, liczbie przechowywanych węzłów w zbiorach pass i wait oraz średni czas badania.



Rysunek 15: Zestawienie zależności liczby odkrytych węzłów, liczby analizowanych węzłów, liczby przechowywanych węzłów oraz średniego czasu badania do rozmiaru modelu.

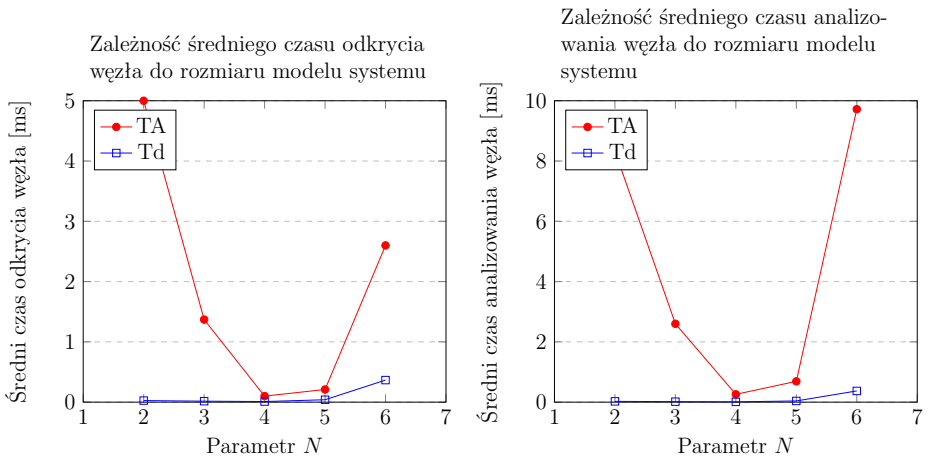
Rozwiązanie Td w porównaniu do TA odkrywa i przechowuje mniej nowych węzłów,

natomiast analizuje więcej węzłów.

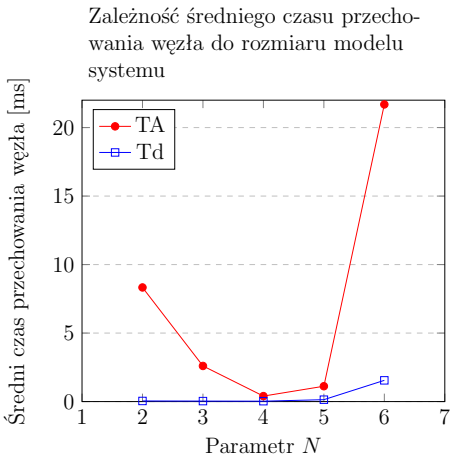
Następnie sprawdzono ile średnio potrzeba czasu na odkrycie węzła, analizowanie węzła oraz przechowanie węzła.

Tabela 9: Zestawienie informacji o średnim czasie [ms] odkrycia węzła, analizowania węzła i przechowania węzła przez TA i Td

	A		B		C	
	Średni czas odkrycia		Średni czas analizowania		Średni czas zapamiętania	
N	TA	Td	TA	Td	TA	Td
2	5,00	0,026	8,33	0,026	8,33	0,045
3	1,37	0,016	2,60	0,016	2,60	0,035
4	0,10	0,010	0,26	0,010	0,40	0,028
5	0,21	0,040	0,69	0,040	1,12	0,139
6	2,60	0,366	9,72	0,372	21,69	1,549



Rysunek 16: Zestawienie zależności średniego czasu odkrycia węzła oraz śrenidego czasu analizowania węzłą do rozmiaru modelu.



Rysunek 17: Zależność średniego czasu przechowania węzła do rozmiaru modelu systemu.

Wraz ze wzrostem parametru N średni czas potrzebny na odkrycie, analizowanie i przechowanie węzła był niemalejący dla Td (za wyjątkiem czasu zapamiętania dla $N = 2$). Natomiast dla rozwiązania TA nieregularny.

Przy przyjętych założeniach Td szybciej odkrywa, analizuje i zapamiętuje węzły. Należy jednak mieć na uwadze, że narzędzie TAPAAL oferuje także weryfikację formuł bez ograniczenia co do jednego żetonu w danym miejscu systemu. Ograniczenie w TA dotyczy sumarycznej liczby żetonów w całym modelu systemu.

6.5 Badanie eksperymentalne własności $EF_I\varphi$ algorytmu konstruowania modelu TTS z dodatkowym warunkiem momentu stopu

Następnie zbadano czy do momentu d wszyscy uczeni wykonają swoją akcję krytyczną. W tym badaniu wybrano wariant nr 2 modelu systemu. Użyto komputera KOM1. Ponadto, aby w rozwiązaniu TAPAAL wymusić weryfikację formuły CTL do momentu d, dodano inwariant do miejsca technicznego ograniczając długość przebywania żetonu w tym miejscu. Spowoduje to weryfikację jedynie w obrębie prefiksu do momentu d.

Tak więc dla $N = 2, 3, 4, 5, 6$ zbadano prawdziwość formuły $EF_{[0,d]}\varphi_N$, gdzie φ_N jest prawdziwa w stanie występującym tuż po tym jak wszyscy uczeni wykonali sekcję krytyczną. Zgodnie z obliczeniami analitycznymi i ustalonymi wcześniej przedziałami α i β, formuła $EF_{[0,d]}\varphi_N$:

- dla $N = 2$ jest spełniona gdy $d \geqslant 4$,
- dla $N = 3$ jest spełniona gdy $d \geqslant 6$,
- dla $N = 4$ jest spełniona gdy $d \geqslant 4$,

- dla $N = 5$ jest spełniona gdy $d \geqslant 6$ oraz

- dla $N = 6$ jest spełniona gdy $d \geqslant 4$.

Przykładowo dla $N = 5$ poczynając od stanu początkowego, aby osiągnąć stan w którym φ_5 jest prawdziwa konieczne jest wykonanie kolejno zdarzeń przedstawionych w tabelce:

Tabela 10: Przykładowy przebieg powodujący stan w którym φ_5 jest spełnione.

Lp.	moment globalny	upływ czasu d lub wykonanie akcji a uruchomienia tranzycji t	Informacja o wejściu/wyjściu uczonego z/do sekcji krytycznej
1	[0,1]	$d_0 = 1$	
2	1	(a_0, t_1)	wejście uczonego nr 1
3	1	$d_1 = 0$	
4	1	(a_1, t_3)	wejście uczonego nr 3
5	[1,2]	$d_2 = 1$	
6	2	(a_2, t_6)	wyjście uczonego nr 1
7	2	$d_3 = 0$	
8	2	(a_3, t_8)	wyjście uczonego nr 3
9	[2,3]	$d_4 = 1$	
10	3	(a_4, t_2)	wejście uczonego nr 2
11	3	$d_5 = 0$	
12	3	(a_5, t_5)	wejście uczonego nr 5
13	[3,4]	$d_6 = 1$	
14	4	(a_6, t_7)	wyjście uczonego nr 2
15	4	$d_7 = 0$	
16	4	(a_7, t_{10})	wyjście uczonego nr 5
17	[4,5]	$d_8 = 1$	
18	5	(a_8, t_4)	wejście uczonego nr 4
19	[5,6]	$d_9 = 1$	
20	6	(a_9, t_9)	wyjście uczonego nr 4

Sumując wartości d_i można zauważyć, że potrzeba co najmniej 6 jednostek czasu, aby każdy z uczonych wykonał akcję krytyczną.

Następnie wygenerowano wyniki weryfikacji przyjętej formuły dla inkrementowanego parametru d dla wszystkich rozważanych N.

Tabela 11: Weryfikacja formuły $EF_{[0,d]}\varphi_N$ przy pomocy TA i Td w zależności od parametrów $d \in \{2,3,4,5,6\}$ i $N \in \{2,3,4,5,6\}$

TA				
N\d	**3**	**4**	**5**	**6**
2	false	**true**	true	true
3	false	false	false	**true**
4	false	**true**	true	true
5	false	false	false	**true**
6	false	**true**	true	true

Td				
N\d	**3**	**4**	**5**	**6**
2	false	**true**	true	true
3	false	false	false	**true**
4	false	**true**	true	true
5	false	false	false	**true**
6	false	**true**	true	true

Przeprowadzone badanie w rezultacie potwierdziło teoretyczne i praktyczne rozważania spełnienia formuły φ_N. Zarówno TA jak i Td wygenerowały zgodnie wyniki dla $N = 2,3,4,5,6$.

6.6 Wpływ wybranej aproksymacji na weryfikację własności modelu systemu

Badanie ma na celu sprawdzenie w jakim stopniu używana aproksymacja pomaga w weryfikacji formuł. W badaniu porównuje się średnie czasy wygenerowania reprezentatywnej przestrzeni stanów, weryfikację wybranej formuły oraz zużytą pamięć na alokację instancji wybranych klas biblioteki TdPNBib.

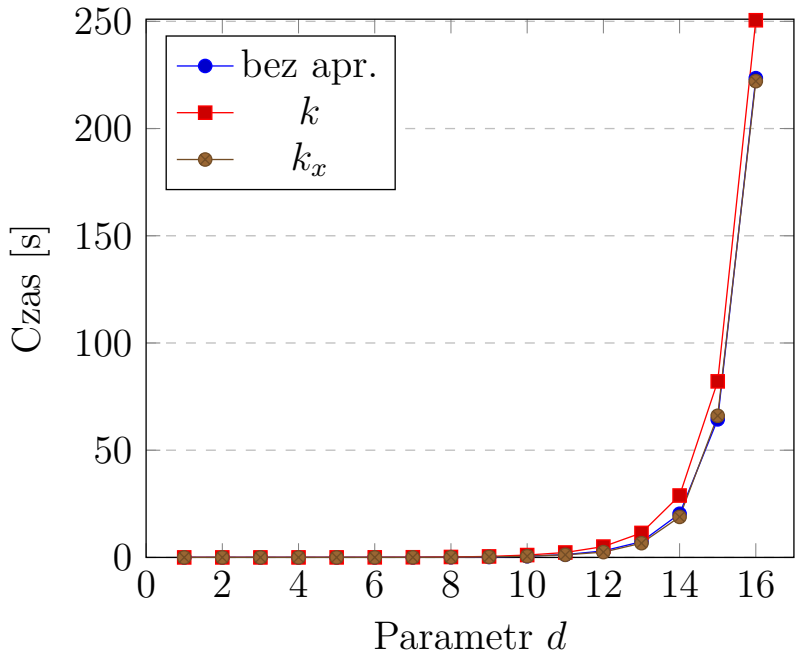
Rozważa się najgorszy przypadek weryfikacji formuły. Wybraną formułą ϕ jest formuła niemożliwa do spełnienia w rozważanym modelu systemu. Do badania wybrano wariant nr 1 modelu systemu z czterema uczonymi ($N = 4$). W badaniu użyto komputera KOM2.

Pojedyncze próba w badaniu stanowi wygenerowanie reprezentatywnego fragmentu przestrzeni stanów do momentu d oraz weryfikację formuły ϕ do momentu d.

Rozpoczęto od porównania średniego czasu badania przypadku bez aproksymacji, a następnie z aproksymacjami k, k_x oraz LU.

Tabela 12: Zależność średniego czasu [s] przeprowadzenia pojedynczego badania od momentu przerwania d.

Moment d	Bez aproksymacji	k	k_x	LU
1	0,001	0,002	0,006	0,006
2	0,000	0,000	0,000	0,000
3	0,001	0,001	0,001	0,001
4	0,002	0,003	0,003	0,004
5	0,004	0,010	0,011	0,006
6	0,012	0,030	0,025	0,007
7	0,031	0,074	0,045	0,006
8	0,088	0,205	0,123	0,009
9	0,217	0,461	0,249	0,022
10	0,568	1,101	0,567	0,072
11	1,300	2,292	1,204	0,238
12	3,039	5,051	2,468	0,848
13	7,337	11,424	6,638	3,145
14	20,359	28,865	18,888	11,570
15	64,381	82,099	66,055	42,989
16	223,449	250,512	222,095	-



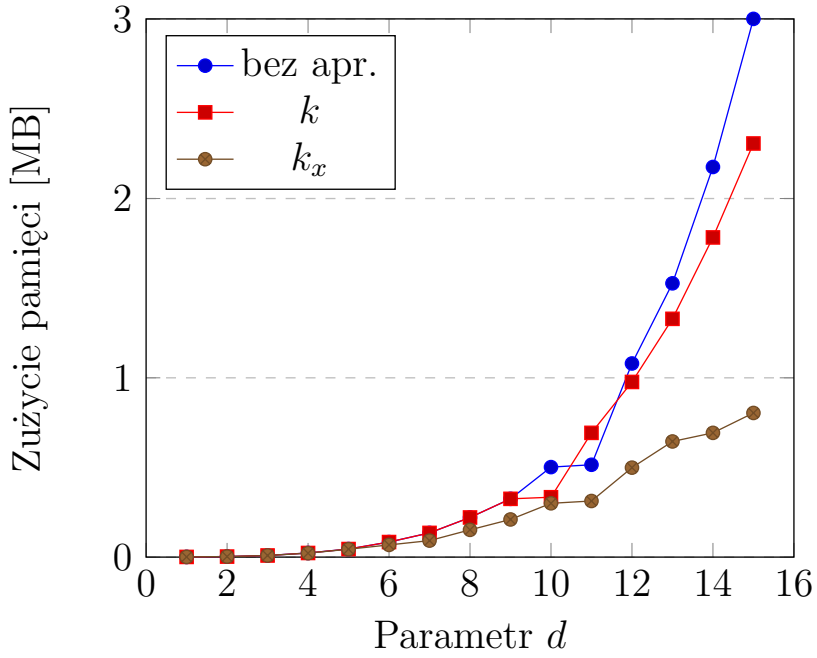
Rysunek 18: Zależność czasu [s] generowania prefiksu zachowania od momentu przerwania d.

Przypadek zastosowania aproksymacji k w stosunku do przypadku bez aproksymacji wymagał średnio więcej czasu. Przypadek aproksymacji k_x i LU wymagał średnio mniej czasu niż przypadek bez aproksymacji.

Następnie sprawdzono jak dużo pamięci potrzeba na sumaryczną alokację obiektów reprezentujących węzeł.

Tabela 13: Zużycie pamięci [MB] dla alokowanych instancji $Wezel$ dla poszczególnych aproksymacji w zależności od momentu przerwania d.

Moment d	Bez aproksymacji	k	k_x
1	0.001	0.001	0.001
2	0.003	0.003	0.003
3	0.009	0.009	0.009
4	0.023	0.023	0.022
5	0.045	0.045	0.044
6	0.084	0.084	0.068
7	0.135	0.135	0.092
8	0.221	0.221	0.152
9	0.325	0.325	0.210
10	0.502	0.334	0.300
11	0.515	0.693	0.313
12	1.080	0.977	0.499
13	1.527	1.329	0.645
14	2.176	1.783	0.693
15	3.002	2.307	0.804



Rysunek 19: Zużycie pamięci [MB] dla alokowanych instancji $Wezel$ dla poszczególnych aproksymacji w zależności od momentu przerwania d

Wykonanie badania dla przypadku bez aproksymacji wymagało najwięcej pamięci (z pominięciem przypadku $d = 11$ w porównaniu z k). Wykonanie badania z aproksymacją k_x zużywało najmniej pamięci w porównaniu do pozostałych przypadków.

6.7 Analiza wpływu wzrostu parametrów modelu systemu na środowisko symulacyjne do konstruowania reprezentatywnego TTS

Badanie ma na celu sprawdzenie jak zmienia się weryfikacja formuł dla modelu systemu, którego parametry przedziałów dostępności są powiększane. Dzięki temu można sprawdzić jak opracowana metoda funkcjonuje w praktyce, gdy przedziały dostępności są szersze. Ponadto dzięki manipulacji szerokością przedziałów dostępności moża rozważyć zastosowanie metody dzielenia przedziałów dostępności w celu wygenerowania dolnego lub górnego przybliżenia reprezentatywnego fragmentu, aby przyspieszyć weryfikację własności [8].

W badaniu użyto wariant nr 2 modelu systemu, w którym znajduje się szesciu filozofów ($N = 6$). Do obliczeń wykorzystano komputer KOM2.

Tabela 14: Średni czas oraz zużycie pamięci [MB] dla poszczególnych wartosci współczynnika powinowactwa a przy zastosowaniu aproksymacji k.

Lp.	a	$a * \alpha$	$a * \beta$	Średni czas [s]
1	10	$[10, 70]$	$[70, 10]$	108,663
2	20	$[20, 140]$	$[140, 20]$	108,630
3	30	$[30, 210]$	$[210, 30]$	109,273
4	40	$[40, 280]$	$[280, 40]$	109,349
5	50	$[50, 350]$	$[350, 50]$	109,289
6	60	$[60, 420]$	$[420, 60]$	109,048
7	70	$[70, 490]$	$[490, 70]$	109,094
8	80	$[80, 560]$	$[560, 80]$	109,192
9	90	$[90, 630]$	$[630, 90]$	109,324
10	100	$[100, 700]$	$[700, 100]$	109,181

$a * \alpha$ i $a * \beta$ to przedziały dostępności modelu systemu nr 2 zależne od parametru a. Różnica w otrzymanych średnich czasach nie przekroczyła 2 sekund. Pojedyncza wybrana próba w badaniu potrzebowała 3,97MB na alokację obiektów typu węzeł.

Zbadano także przypadek użycia aproksymacji k_x oraz LU. W obu przypadkach zarówno średni czas badania jak i zużyta pamięc w zależności od parametru a miały porównywalną zmienność co w badaniu z aproksymacją k.

Zakończenie

W prowadzonych badaniach przyjęto tezę o istnieniu własności modeli c-TdPN, przy których istnieje skończony odcinek początkowy rozwinięcia modelu i dotyczące tego odcinka ograniczenia momentów wykonania tranzycji takie, które jednocześnie wyznaczają wszystkie rozgałęzione procesy czasowe modelowanego systemu. Założono także opracowanie algorytmów rozstrzygania czy dany model systemu ma rzeczone własności oraz wyznaczania odpowiedniego odcinka początkowego i dotyczących go ograniczeń, jeśli istnieją.

W pracy dowiedzono tezę poprzez opracowanie dwóch algorytmów generujących reprezentatywny fragment przestrzeni stanów, zidentyfikowano własności możliwe do opisania w logice modalnej przy pomocy formuł typu: EF, EG, AF i AG oraz opracowano algorytmy je weryfikujące.

Pokazano adaptację sposobu dyskretyzacji przestrzeni stanów dla modeli c-TdPN z dodatkowymi inwariantami. Opisano zidentyfikowane własności w precyzyjnym języku logiki TCTL. Są to kolejno własności bez zagnieżdżeń typu: $EF_I\varphi$, $AF_I\varphi$, $EG_I\varphi$, $AG_I\varphi$, $AG_I\varphi \rightarrow EF_J\psi$ i $AG_I\varphi \rightarrow AF_J\psi$. Opracowano algorytmy weryfikujące powyższe własności. Zbadano złożoność czasową rozwiązania oraz porównano proponowane rozwiązanie z istniejącym narzędziem TAPAAL.

W pracy osiągnięto przede wszystkim:

- opracowano algorytm generowania prefiksu rozgałęzionego procesu czasowego (RPC) reprezentującego zachowanie systemów c-TdPN - algorytm nr 4,
- przedstawiono dowód na generowanie prefiksu zupełnego zachowania przez algorytm nr 4,
- zbadano złożoność czasową dla algorytmu generowania prefiksu rozgałęzionego procesu czasowego (RPC),
- opracowano algorytm generowania prefiksu czasowego systemu tranzycyjnego (TTS) reprezentującego zachowanie systemów c-TdPN w oparciu o istniejący algorytm generowania grafu strefa dla systemów TPN - algorytm nr 5,
- przedstawiono dowód poprawności generowania prefiksu TTS przez algorytm nr 5,
- zbadano złożoność czasową dla algorytmu generowania prefisku czasowego systemu tranzycyjnego (TTS) oraz algorytmów weryfikacji własności,
- zastosowano znane aproksymacje k, k_x, LU_x, aby osiągnąć reprezentatywny fragment zachowania (warunek stopu algorytmu)
- zaproponowano sposób wyznaczania TTS do określonego momentu d,
- zaproponowano algorytmy weryfikacji własności $EF_I\varphi$, $EG_I\varphi$, $AF_I\varphi$, $AG_I\varphi$, $AG_I(\varphi \rightarrow EF_I\psi)$, $AG_I(\varphi \rightarrow AF_I\psi)$,
- zaimplementowano zaproponowany algorytm gerenowania zdyskretyzowanego TTS dla c-TdPN w postaci biblioteki TdPNBib i programów ją stosujących,
- zaimplementowano weryfikację założonych własności,
- porównano narzędzie TdPNBib z istniejącym rozwiązaniem TAPAAL weryfikującym własności opisane przy pomocy formuł CTL,
- przedstawiono wyniki przeprowadzonych badań w postaci analizy porównawczej.

Uzyskano także odpowiedzi na dodatkowe pytania postawione we wstępie.

Jakie opisy zachowania należy rozważać dla modeli c-TdPN?

Wybór dwóch modeli zachowania, rozgałęzionego procesu czasowego (RPC) i czasowego systemu tranzycyjnego (TTS) pozwolił na przeprowadzenie interesujących badań nad zachowaniem systemów czasowych. Model TTS okazał się bardziej przydatny przy implementacji automatycznego narzędzia. Powodem jest możliwość jego dyskretyzacji i reprezentacji stanu symbolicznego tegoż modelu przy pomocy struktury macierzowej. Weryfikacja własności czasowych przebiega przy pomocy zdefiniowanych algorytmów rekurencyjnych.

Model RPC jest modelem pozwalającym ma przechowywanie informacji o bogatszej strukturze. Modelowanie przy pomocy RPC pozwoliło na dokładniejsze opisanie

zachowania mniejszych systemów dla których możliwe okazało się wygenerowanie reprezentatywnego fragmentu bez automatyzacji tego procesu.

Jakie założenia należy przyjąć, aby badać interesujące własności?
Poczynione założenie co do jednego żetonu w miejscu modelu systemu mogłoby zostać osłabione, gdy stosuje się model TTS do opisu zachowania. Pozwoliłoby to na wygodniejszy opis występowania większej liczby obiektów w systemie bez dodawania nowych miejsc. Aczkolwiek zabieg ten skomplikowałby obliczeniowo rozwiązanie oraz utrudniło jego prezentację.

Podczas badań dodano do rozważanego modelu systemu możliwość wymuszania wykonania tranzycji za pomocą inwariantów przypisanych do miejsc systemu. Rozwiązanie to pozwoliło na modelowanie pilności wykonania akcji systemu, którego zachowanie jest reprezentowane przez TTS.

Jakie własności można badać dla modelu c-TdPN?
Przy zastosowaniu RPC możliwe jest dla niedużych systemów wygenerowanie krok po kroku reprezentatywnego fragmentu zachowania i badania osiągalnych stanów, lub wystąpień akcji systemu do przyjętego momentu d. Natomiast przy zastosowaniu TTS zdefiniowano dla modelu c-TdPN logikę pozwalającą na precyzyjny opis własności czasowych przy pomocy formuł $EF_I\varphi$, $EG_I\varphi$, $AF_I\varphi$, $AG_I\varphi$, $AG_I(\varphi \rightarrow EF_I\psi)$, $AG_I(\varphi \rightarrow AF_I\psi)$.

Jaki jest koszt algorytmów je weryfikujących?
Oszacowany koszt algorytmów generujących reprezentatywny fragment zachowania zarówno rozgałęzionego procesu czasowego jak i czasowego systemu tranzycyjnego okazał się wielomianowy. Umożliwiło to implementację algorytmu, który generuje reprezentatywny fragment w racjonalnym obliczeniowo czasie.

Oszacowana złożoność czasowa dla algorytmów weryfikacji modelowej, pokazała, że przy dużej liczbie miejsc oraz dużej przestrzeni stanów badanego systemu, proces weryfikacji może potrwać zbyt długo.

Jaka jest jakość proponowanego rozwiązania w porównaniu do znanego narzędzia TAPAAL? Przy przyjętych założeniach wyniki uzyskane przy pomocy metody zaimplementowane w bibliotece TdPNBib miało zgodne i porównywalne rezultaty weryfikacji jak rozwiązanie uzyskane przy pomocy narzędzia TAPAAL. Weryfikacja formuł modeli systemów, w których rozrost przestrzeni stanów był obszerniejszy, przebiegła szybciej w narzędziu TAPAAL. Powodem przewagi w szybkości jest stosowany inny język programowania.

W dalszym etapie pracy naukowej autor planuje prowadzić badania nad:

- **Osłabieniem założeń dla modelowanego systemu**. Rozszerzenie możliwosci używania wielu żetonów w jednym miejscu modelu systemu pozwoliłoby na wygodniejszy opis występowania większej liczby obiektów w systemie. Optymalizacja implementacji rozwiązania w kierunku mniejszego zużycia wymaganej pamięci oraz szybszego funkcjonowania pozwoliłaby na efektywniejsze działanie procesu weryfikacji.

- **Poszerzeniem języka opisu własności.** Zastosowanie innych rozszerzeń logik modalnych do opisu interesujących własności systemów. Mogą to być np. logi-

ki epistemiczne pozwalające na reprezentację informacji na temat wiedzy jaką posiadają obiekty systemu [38] lub logik ACTL [31]. Z uwagi na brak pilności w modelu c-TdPN do reprezentacji własności systemu zasadny jest wybór także logiki TCTL* do modelowania własności systemu. Jest to logika modalna TCTL rozszerzona o możliwość konstruowania formuł, w których niekoniecznie przed każdym operatorem modalnym wystąpuje kwantyfikator ścieżkowy.Tak więc możliwe jest konstruowanie formuł rozgałęzionej logiki, której podformuły są opisane przy pomocy logiki opartej na liniowej strukturze czasu [40]. Analizę rozszerzenia rozwiązania weryfikacji formuł TCTL o modyfikację opisaną w [16].

- **Rozbudową środowiska modelowania i weryfikacji.** Opracowanie konwersji modelu przygotowanego w narzędziu TAPAAL na model do narzędzia TdPNBib. Ponadto opracowanie narzędzia z interfejsem graficznym, pozwalającym na wizualne tworzenie modeli systemu, pozwoliłoby osiągnąć zwarte rozwiązanie modelowania i symulacji stosujące opracowaną metodę. Opracowanie algorytmów, które weryfikują własność logiki modalnej w sposób on-the-fly [26] przyspieszyłoby weryfikację pojedynczych formuł przy których nie jest potrzebny cały reprezentatywny fragment zachowania modelu systemu.

- **Porównaniem opracowanej metody z kolejnymi istniejącymi rozwiązaniami.** Zbadanie adaptacji (do modeli c-TdPN) rozwiązania enkapsulacji upływu czasu w węzłach przy pomocy grafu agregacyjnego (ang. Timed Aggregate Graph) oraz sposobu zachowania relacji pomiędzy umożliwionymi tranzycjami [33]. Zbadanie rozwiązania rozluźnionego rozwinięcia (ang. relaxed unfolding)[5] w kontekście rozgałęzionych procesów czasowych. Identyfikacja modeli, które mają jednakową/bliską wyrażalność jak systemy c-TdPN [6].

Literatura

[1] P. A. Abdulla and A. Nylén. Timed petri nets and bqos. In *ICATPN*, volume 1, pages 53–72. Springer, 2001.

[2] S. Akshay, B. Genest, and L. Hélouët. Timed petri nets with (restricted) urgency. 2014.

[3] P. Baldan, A. Bruni, A. Corradini, B. König, C. Rodríguez, and S. Schwoon. Efficient unfolding of contextual petri nets. *Theoretical Computer Science*, 449:2–22, 2012.

[4] G. Behrmann, P. Bouyer, K. G. Larsen, and R. Pelánek. Lower and upper bounds in zone-based abstractions of timed automata. *International Journal on Software Tools for Technology Transfer*, 8(3):204–215, 2006.

[5] F. C. V. Benito and L. A. Künzle. Timing analysis of cyclic time petri net using relaxed unfolding and global time technique. In *Distributed Simulation and Real*

Time Applications (DS-RT), 2015 IEEE/ACM 19th International Symposium on, pages 147–154. IEEE, 2015.

[6] B. Berard, F. Cassez, S. Haddad, D. Lime, and O. H. Roux. The expressive power of time petri nets. *Theoretical Computer Science*, 474:1–20, 2013.

[7] B. Berthomieu, D. Lime, O. H. Roux, and F. Vernadat. Reachability problems and abstract state spaces for time petri nets with stopwatches. *Discrete Event Dynamic Systems*, 17(2):133–158, 2007.

[8] S. V. Birch, T. S. Jacobsen, J. J. Jensen, Ch. Moesgaard, N. N. Samuelsen, and J. Srba. Interval abstraction refinement for model checking of timed-arc petri nets. In *International Conference on Formal Modeling and Analysis of Timed Systems*, pages 237–251. Springer, 2014.

[9] H. Boucheneb, G. Gardey, and O. H. Roux. Tctl model checking of time petri nets. *Journal of Logic and Computation*, 19(6):1509–1540, 2009.

[10] P. Bouyer, S. Haddad, and P.-A. Reynier. Timed unfoldings for networks of timed automata. In *International Symposium on Automated Technology for Verification and Analysis*, pages 292–306. Springer, 2006.

[11] P. Bouyer, S. Haddad, and P.-A. Reynier. Timed petri nets and timed automata: On the discriminating power of zeno sequences. *Information and Computation*, 206(1):73–107, 2008.

[12] A. David, L. Jacobsen, M. Jacobsen, and J. Srba. A forward reachability algorithm for bounded timed-arc petri nets. *arXiv preprint arXiv:1211.6194*, 2012.

[13] A. David, L. Jacobsen, Jacobsen M., K.Y. Jørgensen, M.H. Møller, and J. Srba. TAPAAL 2.0: integrated development environment for timed-arc Petri nets. In *Proceedings of the 18th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'12)*, volume 7214 of *LNCS*, pages 492–497. Springer-Verlag, 2012.

[14] C. Daws and S. Tripakis. Model checking of real-time reachability properties using abstractions. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 313–329. Springer, 1998.

[15] D. de Frutos Escrig, V. V. Ruiz, and O. M. Alonso. Decidability of properties of timed-arc petri nets. In *International Conference on Application and Theory of Petri Nets*, pages 187–206. Springer, 2000.

[16] M. E. Esmaili, R. Entezari-Maleki, and A. Movaghar. Improved region-based tctl model checking of time petri nets. *Journal of Computing Science and Engineering*, 9(1):9–19, 2015.

[17] J. Esparza, S. Römer, and W. Vogler. An improvement of mcmillan's unfolding algorithm. *Formal Methods in System Design*, 20(3):285–310, 2002.

[18] P. Filipkowski. Technologie agentowe w komputerowym modelowaniu i symulacjach. *Collegium of Economic Analysis Annals*, (40):229–242, 2016.

[19] P. Filipkowski and M. Horodelski. Zastosowanie rozproszonej platformy modelowania i symulacji do analizy informacji medycznych na bieżąco. *Roczniki Kolegium Analiz Ekonomicznych/Szkoła Główna Handlowa*, (35):149–164, 2014.

[20] P. Filipkowski and M. Horodelski. Modelowanie procesów mnogich/masowych prm przy wykorzystaniu czasowej sieci petriego. *Roczniki Kolegium Analiz Ekonomicznych/Szkoła Główna Handlowa*, (52):45–58, 2018.

[21] P. Filipkowski, M. Horodelski, and K. Polańska. Blockchain w zdecentralizowanej autoryzacji transakcji barterowych. *Research Papers of the Wroclaw University of Economics/Prace Naukowe Uniwersytetu Ekonomicznego we Wroclawiu*, (527), 2018.

[22] P. Filipkowski, M. Horodelski, and E. Skąpska. Modelling of rational agent on the electronic service market. *[po uwagach od członków panelu Expert Systems with Applications, przed złożeniem do recenzji] Expert Systems with Applications*, planowana publikacja w roku 2020.

[23] G. Gardey, O. H. Roux, and O. F. Roux. Using zone graph method for computing the state space of a time petri net. In *International Conference on Formal Modeling and Analysis of Timed Systems*, pages 246–259. Springer, 2003.

[24] A. Geist. *PVM: Parallel virtual machine: a users' guide and tutorial for networked parallel computing*. MIT press, 1994.

[25] A. Giua, F. DiCesare, and M. Silva. Generalized mutual exclusion contraints on nets with uncontrollable transitions. In *Systems, Man and Cybernetics, 1992., IEEE International Conference on*, pages 974–979. IEEE, 1992.

[26] R. Hadjidj and H. Boucheneb. On-the-fly tctl model checking for time petri nets. *Theoretical Computer Science*, 410(42):4241–4261, 2009.

[27] H.-M. Hanisch. Analysis of place/transition nets with timed arcs and its application to batch process control. *Application and Theory of Petri Nets 1993*, pages 282–299, 1993.

[28] M. Horodelski. Modelowanie i automatyczna weryfikacja systemów kryptowalutowych. *[praca po pozytywnej recenzji] Roczniki Kolegium Analiz Ekonomicznych/Szkoła Główna Handlowa*, publikacja planowana w roku 2019.

[29] L. Jacobsen, M. Jacobsen, M. H. Møller, and J. Srba. Verification of timed-arc petri nets. In *International Conference on Current Trends in Theory and Practice of Computer Science*, pages 46–72. Springer, 2011.

[30] A. Janicki. Labtsi(tm)-platforma modelowania i symulacji, red. *Wydawnictwo KUL, Lublin*, 2011.

[31] A. Janowska, W. Penczek, A. Półrola, and A. Zbrzezny. Using integer time steps for checking branching time properties of time petri nets. In *Transactions on Petri Nets and Other Models of Concurrency VIII*, pages 89–105. Springer, 2013.

[32] C. Jard, D. Lime, O. H. Roux, and L.-M. Traonouez. Symbolic unfolding of parametric stopwatch petri nets. *Formal Methods in System Design*, 43(3):493–519, 2013.

[33] K. Klai. Timed aggregate graph: A finite graph preserving event-and state-based quantitative properties of time petri nets. In *Transactions on Petri Nets and Other Models of Concurrency X*, pages 34–54. Springer, 2015.

[34] K. L. McMillan and D. K. Probst. A technique of state space search based on unfolding. *Formal methods in system design*, 6(1):45–65, 1995.

[35] P. Merlin and D. Farber. Recoverability of communication protocols–implications of a theoretical study. *IEEE transactions on Communications*, 24(9):1036–1043, 1976.

[36] P. M. Merlin. A study of the recoverability of computing systems. 1975.

[37] M. L. Minsky. *Computation: finite and infinite machines*. Prentice-Hall, Inc., 1967.

[38] A. Męski, W. Penczek, M. Szreter, B. Woźna-Szcześniak, and A Zbrzezny. Bdd-versus sat-based bounded model checking for the existential fragment of linear temporal logic with knowledge: algorithms and their performance. *Autonomous Agents and Multi-Agent Systems*, 28(4):558–604, 2014.

[39] W. Penczek and A. Pólrola. Advances in verification of time petri nets and timed automata.

[40] W. Penczek and A. Półrola. Abstractions and partial order reductions for checking branching properties of time petri nets. In *International Conference on Application and Theory of Petri Nets*, pages 323–342. Springer, 2001.

[41] C. Rodríguez, S. Schwoon, and P. Baldan. Efficient contextual unfolding. *CONCUR 2011–Concurrency Theory*, pages 342–357, 2011.

[42] V V. Ruiz, D. de Frutos Escrig, and F C. Gómez. On non-decidability of reachability for timed-arc petri nets. In *Petri Nets and Performance Models, 1999. Proceedings. The 8th International Workshop on*, pages 188–196. IEEE, 1999.

[43] J. Srba. Timed-arc petri nets vs. networks of timed automata. In *International Conference on Application and Theory of Petri Nets*, pages 385–402. Springer, 2005.

[44] J. Srba. Comparing the expressiveness of timed automata and timed extensions of petri nets. In *International Conference on Formal Modeling and Analysis of Timed Systems*, pages 15–32. Springer, 2008.

[45] M. Szpyrka. *Sieci Petriego w modelowaniu i analizie systemów współbieżnych.* Wydawnictwa Naukowo-Techniczne, 2008.

[46] L.-M. Traonouez, B. Grabiec, C. Jard, D. Lime, and O. Roux. Symbolic unfolding of parametric stopwatch petri nets. *Automated Technology for Verification and Analysis*, pages 291–305, 2010.

[47] B. Walter. Timed petri-nets for modelling and analyzing protocols with real-time characteristics. In *Protocol Specification, Testing, and Verification*, pages 149–159, 1983.

[48] J. Winkowski. Protocols of accessing overlapping sets of resources. *Information Processing Letters*, 12(5):239–243, 1981.

[49] J. Winkowski. Failure-resistant resource management in a distributed multi-agent system. *Prace IPI PAN*, 2000.

[50] J. Winkowski. Reachability in contextual nets. *Fundamenta Informaticae*, 51(1-2):235–250, 2002.

Spis rysunków

1 Model c-TdPN dla systemu obliczeń rozproszonych. 16
2 Rozgałęziony proces czasowy dla systemu zadań i dwóch komputerów. . . . 18
3 Rozgałęziony proces czasowy modelu TdPN. Przerywana linia oznacza możliwe zdarzenie. 21
4 Model systemu zadań i dwóch węzłów M_1 typu PSwPN.□ 24
5 Przykładowy $\Gamma(M_1)$ dla systemu zadań i dwóch węzłów M_1. 26
6 Prezentacja $dur(B', \delta)$ na półosi czasu. 30
7 M_1 model systemu TPN z Rysunku 4 z wyłączonym węzłem 2 i $a = b = 0$. . 35
8 Model $\Gamma(M_1)$ reprezentujący fragment zachowania systemu M_1 36
9 Model M_2 typu TPN realizujący obliczenia na jednym węźle. 43
10 Graf stanów symbolicznych reprezentujący przestrzeń stanów modelu M_2. . 44
11 Model M_2 rozszerzony o tranzycję t_5. 45
12 Graf stanów symbolicznych reprezentujący przestrzeń stanów modelu M_2 rozszerzonego o tranzycję t_5. 46

13 Wyróżnienie stanów symbolicznych spełniających formułę φ_1 (pozostałe stany spełniają ψ_1). 52
14 Model systemu synchonizacji dostępu do zasobów $N = 5$.
źródło: na podstawie [45] str. 41 83
15 Zestawienie zależności liczby odkrytych węzłów, liczby analizowanych węzłów, liczby przechowywanych węzłów oraz średniego czasu badania do rozmiaru modelu. 89
16 Zestawienie zależności średniego czasu odkrycia węzła oraz śrenidego czasu analizowania węzłą do rozmiaru modelu. 90
17 Zależność średniego czasu przechowania węzła do rozmiaru modelu systemu. 91
18 Zależność czasu [s] generowania prefiksu zachowania od momentu przerwania d. 94
19 Zużycie pamięci [MB] dla alokowanych instancji $Wezel$ dla poszczególnych aproksymacji w zależności od momentu przerwania d 96
20 Schemat modelu systemu autoryzacji dla $N = 3$. 107
21 Fragment sieci reprezentujący działanie i-tego węzła dla $N = 3$, $i = 1$, $j \in \{0, 2\}$. 107
22 Model reprezentujący DM, bez parametryzacji na chwilę obecną. 111
23 Reprezentatywny fragment zachowania modelu systemu N – rozgałęziony proces czasowy 116

Spis tabel

1 Rozważane warianty modelu systemu synchronizacji dostępu do danych 84
2 Weryfikacja własności typu $EF_I\varphi_i$ przy pomocy TdPNBib, w zależności od N, przedziału $I = [0, 10]$, w wariancie nr 1 modelu systemu. 85
3 Weryfikacja własności typu $EF_I\varphi_i$ przy pomocy TdPNBib, w zależności od N, przedziału $I = [0, \infty]$, w wariancie nr 2 modelu systemu. 85
4 Weryfikacja własności typu $EF_I\varphi_i$ przy pomocy TdPNBib, w zależności od N, przedziału $I = [0, 10]$, w wariancie nr 3 modelu systemu. 86
5 Weryfikacja własności typu $EF_I\varphi_i$ przy pomocy TdPNBib, w zależności od N, przedziału $I = [0, \infty]$, w wariancie nr 4 modelu systemu. 86
6 Porównanie modyfikacji $M1$ i $M2$, wygenerowane przy pomocy KOM1 87
7 Porównanie modyfikacji $M1$ i $M2$, wygenerowane przy pomocy KOM2 87
8 Zestawienie informacji o liczbie węzłów po wygenerowaniu przez TA i Td reprezentatywnego fragmentu przestrzeni stanów 89
9 Zestawienie informacji o średnim czasie [ms] odkrycia węzła, analizowania węzła i przechowania węzła przez TA i Td 90
10 Przykładowy przebieg powodujący stan w którym φ_5 jest spełnione. . . 92
11 Weryfikacja formuły $EF_{[0,d]}\varphi_N$ przy pomocy TA i Td w zależności od parametrów $d \in \{2, 3, 4, 5, 6\}$ i $N \in \{2, 3, 4, 5, 6\}$ 93
12 Zależność średniego czasu [s] przeprowadzenia pojedynczego badania od momentu przerwania d. 94

13 Zużycie pamięci [MB] dla alokowanych instancji $Wezel$ dla poszczególnych aproksymacji w zależności od momentu przerwania d. 95
14 Średni czas oraz zużycie pamięci [MB] dla poszczególnych wartosci współczynnika powinowactwa a przy zastosowaniu aproksymacji k. 97
15 Zestawienie zdefiniowanych i weryfikowanych założeń dla systemu . . . 108
16 Zestawienie informacji o liczbie węzłów po wygenerowaniu przez TA i Td reprezentatywnego fragmentu przestrzeni stanów 109
17 Zestawienie informacji o średniej ilości czasu [s] potrzebnej na odkrycie, zbadanie i przechowanie węzła 109
18 Interpretacja miejsc modelu 112
19 Interpretacja tranzycji modelu 112
20 Interpretacja przedziałów dostępności modelu 113

Załączniki

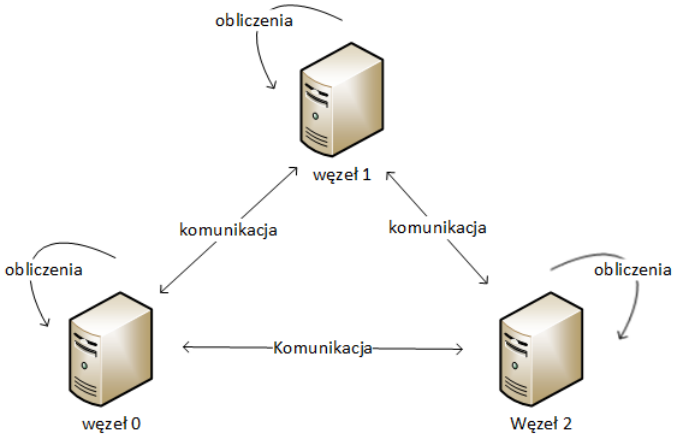
A Systemy transakcyjne i łańcuch bloków

Kolejnym obszarem zastosowania opracowanej metody są systemy tranzakcyjne w skład których zaliczamy także systemy kryptowalutowe oparte o technologię blockchain (ang. łańcuch bloków). Autor rozprawy w publikacji "Blockchain w zdecentralizowanej autoryzacji transakcji barterowych"[21] przedstawił zastosowanie technologii łańcucha bloków do zabezpieczenia transakcji barterowych. Natomiast w pracy pt "Modelowanie i automatyczna weryfikacja systemów kryptowalutowych" (praca po pozytywnej recenzji) [28] prezentuje zastosowanie opracowanej metody w rozprawie do analizy i weryfikacji mechanizmu konsensusu w łańcuchu bloków na przykładzie kryptowaluty bitcoin.

Opracowany model reprezentuje mechanizm konsensusu przy zastosowaniu PoW (ang Proof of Work) dla autoryzacji transakcji systemu kryptowalutowego. System stanowią węzły, nazywane także górnikami, które tworzą i współdzielą pomiędzy sobą elektroniczną księgę rachunkową reprezentowaną przez łańcuch bloków. Każdy górnik komunikuje się z pozostałytmi dostępnymi w sieci. Każdy górnik może wygenerować nowy blok i wysłać informację o nim dla pozostałych górników w celu weryfikacji bloku. Każdy górnik może odebrać informację o nowym bloku i zweryfikować go. Każdy górnik dysponuje mocą obliczeniową, która umożliwia przetwarzanie co najwyżej tyle zadań na ile pozwalają na to jego zasoby.

A.1 Opis modelu systemu

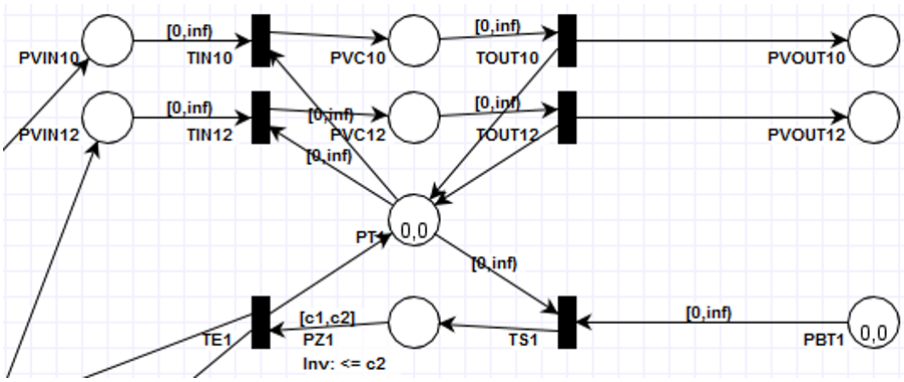
Model systemu zawiera trzy parametry. Parametr N określa liczbę dostępnych węzłów w systemie oraz dwa parametry przedziału $[c1, c2]$ określającego minimalny i maksymalny średni czas wygenerowania kodu hash wraz z rozwiązaniem puzzli dla nowo przygotowanego bloku transakcji.



Rysunek 20: Schemat modelu systemu autoryzacji dla $N = 3$.

Dla $N = 3$ model reprezentuje współpracę trzech węzłów. Techniczny model sieci przygotowano przy pomocy narzędzia TAPAAL. Model jest przechowywany w pliku o strukturze XML.

Każdy i-ty węzeł systemu można scharakteryzować przy pomocy fragmentu czasowej sieci Petriego. Na rysunku 21 przedstawiono fragment trójwęzłowej sieci reprezentujący węzeł nr 1. Węzły w sieci numerowane są od 0.



Rysunek 21: Fragment sieci reprezentujący działanie i-tego węzła dla $N = 3$, $i = 1$, $j \in \{0, 2\}$.

Stan początkowy i-tego węzła jest reprezentowany przez gotowość do pracy (żeton w miejscu PTi) oraz gotowość do wybrania transakcji do spakowania do jednego bloku (żeton w miejscu $PBTi$). Jeden żeton w PTi to możliwość wykonywania jednego zadania (generowanie bloku, weryfikacja bloku) w tym samym czasie. Umieszczenie żetonu w PZi oznacza generowanie nowego bloku zbioru transakcji wybranych przez węzeł i-ty włącznie z rozwiązaniem puzzli. Żeton w miejscu $PVINij$ to wygenerowany nowy blok przez j-ty węzeł przesłany do i-tego węzła, w miejscu $PVCij$ żeton symbolizuje

weryfikację tego bloku, a w miejscu $PVOUTij$ wynik weryfikacji. Przy miejscu PZi występuje ograniczenie, więc żeton w tym miejscu nie może przebywać dłużej niż do momentu $c2$ co oznacza ograniczenie górne czasu generowania bloku przez i-ty węzeł.

Rozpoczęcie generowania bloku (tranzycja TSi) przez i-ty węzeł jest reprezentowane przez tranzycje TSi. TEi kończy generowanie bloku i wysyła dane do pozostałych węzłów według trzeciej grupy mechanizmy propagacji. Tranzycja $TINij$ reprezentuje rozpoczęcie weryfikacji bloku nadesłanego przez j-ty węzeł, a tranzycja $TOUTij$ kończy tą weryfikację.

Tak opracowany model systemu jest zdecentralizowany ponieważ stanowi połączenie równoważnych funkcjonalnie węzłów. Każdy węzeł w zależności od parametru N zawiera $3N$ miejsc oraz $2N$ tranzycji. Stąd $3N^2$ to liczba wszystkich miejsc sieci, a $2N^2$ liczba wszystkich tranzycji sieci. Przyjęto, że czas jaki potrzebuje węzeł na wygenerowanie nowego bloku to około 10 minut.

A.2 Analiza wybranych własności opisanych w TdPN-TCTL

Autor następnie modeluje podstawowe własności jakie powinien spełniać mechanizm konsensusu PoW. Kilka wybranych przedstawiono w tabeli nr 15. Własności te następnie zostały przekształcone do postaci formuł CTL bez zagnieżdżenia, aby możliwa była weryfikacja przy pomocy opracowanej przez autora biblioteki TdPNBib i dla porównania narzędzia TAPAAL.

Tabela 15: Zestawienie zdefiniowanych i weryfikowanych założeń dla systemu

Lp.	Własności	Czas wer.[s]		Wer.
		TA	Td	
1	W każdym przypadku węzeł i-ty nie może przetwarzać więcej zadań niż ma dostępnych wątków (dla $i = 0, 1, 2$).	0,080	2,532	spełnione
2	Istnieje możliwość że jedynie węzeł i-ty wygeneruje nowy blok, a pozostałe węzły zweryfikują jego poprawność (dla $i = 0, 1, 2$).	0,020	0,330	spełnione
3	Istnieje możliwość, że każdy i-ty węzeł wygeneruje nowy blok, a pozostałe węzły zweryfikują jego poprawność (dla $i = 0, 1, 2$).	0,380	0,269	spełnione

Przykładowo formuła TCTL reprezentująca pierwszą własność, zredukowana do formuły CTL ma postać:

$$AG(\varphi(0, 2, 1) \wedge \varphi(1, 2, 0) \wedge \varphi(2, 1, 0))$$

przy czym formuła logiczna $\varphi(i,j,k)$ wyrażająca się wzorem:

$$\begin{aligned}\varphi(i,j,k) = ((|PZi| = 1 \wedge |PTi| = 0 \wedge |PVCij| = 0 \wedge |PVCik| = 0) \vee \\ (|PZi| = 0 \wedge |PTi| = 1 \wedge |PVCij| = 0 \wedge |PVCik| = 0) \vee \\ (|PZi| = 0 \wedge |PTi| = 0 \wedge |PVCij| = 1 \wedge |PVCik| = 0) \vee \\ (|PZi| = 0 \wedge |PTi| = 0 \wedge |PVCij| = 0 \wedge |PVCik| = 1))\end{aligned}$$

oznacza, że i-ty węzeł w tym samym momencie może być albo gotowy do pracy albo może generować nowy blok albo może weryfikować otrzymany nowo wygenerowany blok od j-tego lub od k-tego węzła. Zapis $|P|$ oznacza liczbę żetonów w miejscu P.

A.3 Analiza porównawcza reprezentatywnych przestrzeni stanów (TAPAAL)

Następnie sprawdzono jak długo zajmuje wygenerowanie reprezentatywnego fragmentu przestrzeni stanów i jak dużo odnaleziono węzłów przy pomocy narzędzia TAPAAL (silnik verifytapn bez dodatkowych opcji przyspieszających obliczenia) oraz biblioteki TdPNBib.

Tabela 16: Zestawienie informacji o liczbie węzłów po wygenerowaniu przez TA i Td reprezentatywnego fragmentu przestrzeni stanów

	A		B		C		D	
	Odkryte		Analizowane		Przechowywane		Średni czas[s]	
N	TA	Td	TA	Td	TA	Td	TA	Td
1	3	3	3	4	3	3	0,020	0,061
2	37	55	26	56	26	43	0,110	0,066
3	18705	144265	7079	144254	7079	62893	0,390	276,436
4	12901677	-	8230273	-	8230273	-	2943,810	-

Ze wzgledu na różnicę otrzymanych liczb węzłów sprawdzono ile potrzeba średnio czasu na wygenerowanie jednego węzła z analogicznym podziałem na odkryte, zbadane i przechowywane

Tabela 17: Zestawienie informacji o średniej ilości czasu [s] potrzebnej na odkrycie, zbadanie i przechowanie węzła

N	Węzeł odkryty		Węzeł zbadany		Węzeł przechowywany	
	TA	Td	TA	Td	TA	Td
1	0,0067	0,0203	0,0067	0,0153	0,0067	0,0203
2	0,0030	0,0012	0,0042	0,0012	0,0042	0,0015
3	0,0000	0,0019	0,0001	0,0019	0,0001	0,0044
4	0,0002	-	0,0004	-	0,0004	-

Td potrzebowało średnio więcej czasu na wygenerowanie węzła niż TA, przy czym wyjątkiem jest $N = 2$. Rozwiązanie Td odkryło więcej węzłów w badaniu.

B Procedury Państwowego Ratownictwa Medycznego

Problem modelowania systemów PRM jest aktualnym problemem optymalizującym działanie systemów PRM. Problematykę poruszono na konferencji TIAPISZ'17, gdzie zaprezentowano pomysł na modelowanie i weryfikację własności modelowanego systemu przy pomocy rozgałęzionych procesów. Wyniki opublikowano w pracy pt "Modelowanie procesów mnogich/masowych PRM przy wykorzystaniu czasowej sieci Petriego" [20].

Państwowe Ratownictwo Medyczne (w skr. PRM) jest systemem powołanym dla celów realizacji zadań polegających na zapewnieniu pomocy każdej osobie znajdującej się w stanie nagłego zagrożenia zdrowotnego. Zasady organizacji, funkcjonowania i finansowania systemu oraz zasady zapewnienia edukacji w zakresie udzielania pierwszej pomocy są określone w ustawie o PRM. Głównym dokumentem na którym bazuje funkcjonowanie systemu PRM w danym województwie jest „Wojewódzki plan działania systemu Państwowego Ratownictwa Medycznego", którego wzór opublikowano w rozporządzeniu Ministra ds. Zdrowia z dnia 15 grudnia 2014 . Plan dla województwa jest sporządzany/aktualizowany przez wojewodę i zatwierdzany przed właściwego ministra ds. zdrowia.

Specjalistyczne procedury postępowania, w szczególności wystąpienie zdarzeń mnogich/masowych, są opisane w dodatkowych dokumentach - zaleceniach. Dokument pt. „Zalecenia Konsultanta Krajowego w dziedzinie medycyny ratunkowej dotyczące procedur postępowania na wypadek wystąpienia zdarzenia mnogiego/masowego" (aut. prof. dr hab. n. med. Jerzy Robert Ładny, konsultant krajowy w dziedzinie medycyny ratunkowej) zawiera opis procedur postępowania dla odpowiednich jednostek w systemie Państwowego Ratownictwa Medycznego w razie wystąpienia zdarzeń mających charakter mnogich lub/i masowych (w skr. ZMM).

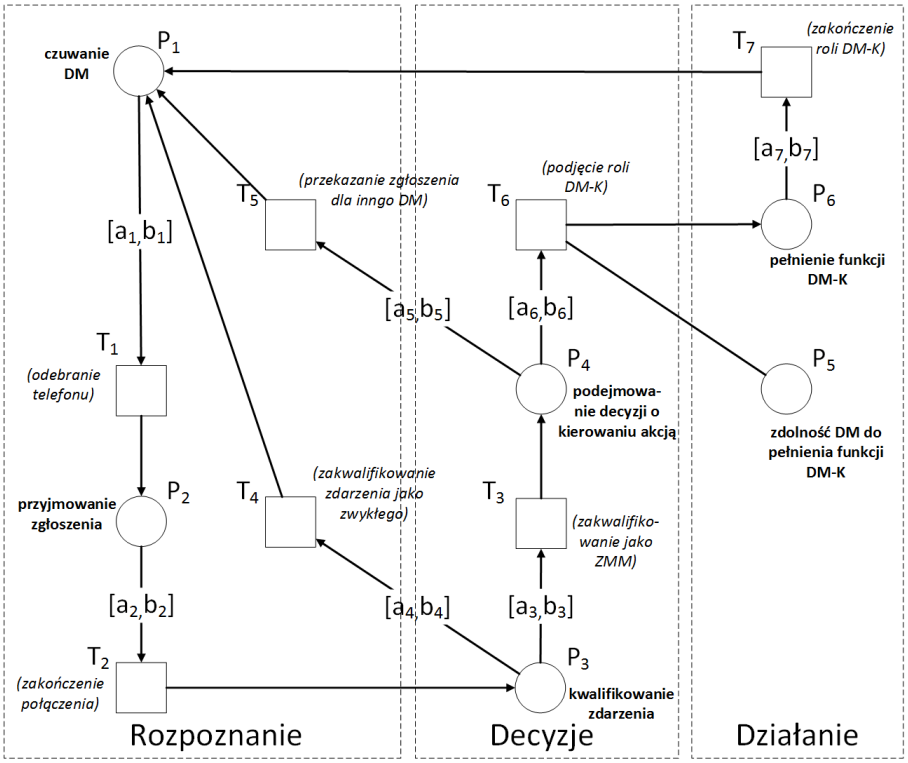
B.1 Opis modelu systemu

Sposób postępowania dyspozytora medycznego (w skr. DM) przy zgłoszeniu mogącym być zdarzeniem mnogim i/lub masowym w którym może podjąć się pełnienia funkcji dyspozytora medycznego kierującego (w skr. DM-K), przedstawiono w 4 punktach:

1. „przyjęcie powiadomienia o zdarzeniu zgodnie z rozporządzeniem (...),
2. zakwalifikowanie zdarzenia jako zdarzenia o potencjalnym charakterze mnogim/masowym,
3. podjęcie decyzji o uruchomieniu postępowania zgodnie z procedurą – zdarzenie mnogie/masowe,
4. przekazanie informacji o uruchomieniu procedury – zdarzenie mnogie/masowe osobie, która będzie pełnić funkcje dyspozytora medycznego kierującego DM-K, w sytuacji gdy dyspozytor medyczny odbierający powiadomienie o zdarzeniu nie będzie pełnił funkcji DM-K."

Wybrana procedura charakterystyczna dla ZMM jest podstawą do opracowania odpowiedniego modelu czasowej sieci Petriego oddającego współdziałanie występujących w niej obiektów, relacji pomiędzy nimi oraz otoczeniem.

Parametrem systemu jest N określający liczbę dyspozytorów medycznych dostępnych w systemie.



Rysunek 22: Model reprezentujący DM, bez parametryzacji na chwilę obecną.

Graf sieci przedstawiony na rysunku 22 reprezentuje działanie jednego DM ($N = 1$). Model systemu został dalej scharakteryzowany przez opis znaczenia miejsc, tranzycji, połączeń z wyróżnionymi przedziałami dostępności pomiędzy nimi oraz obecność żetonów w miejscach.

Tabela 18: Interpretacja miejsc modelu

Lp	S	Opis miejsca	Interpretacja wieku żetonu w miejscu
1	P_1	czuwanie DM	długość czasu czuwania DM
2	P_2	przyjmowanie zgłoszenia o zdarzeniu	długość czasu rozmowy DM ze świadkiem zdarzenia
3	P_3	kwalifikowanie zdarzenia	długość czasu procesu kwalifikowania
4	P_4	podejmowanie decyzji o kierowaniu akcją przez DM	długość czasu podjęcia decyzji o kierowaniu akcją
5	P_5	zdolność DM do funkcji DM-K	długość czasu zdolności DM do funkcji DM-K przed jej rozpoczęciem
6	P_6	kierowanie akcją jako DM-K	długość czasu pełnienia funkcji DM-K przez odbierającego DM

Tabela 19: Interpretacja tranzycji modelu

Lp.	S	Opis tranzycji
1	T_1	odebranie telefonu zdarzenia przez DM
2	T_2	zakończenie przyjmowania zgłoszenia przez DM
3	T_3	zakwalifikowanie zdarzenia jako mnogiego/masowego i uruchomienie postępowania zgodnie z procedurami
4	T_4	zakwalifikowanie zdarzenia jako zwykłego
5	T_5	oddanie funkcji DM-K innemu DM
6	T_6	objęcie funkcji DM-K przez odbierający DM
7	T_7	zakończenie funkcji DM-K przez odbierający DM

Tabela 20: Interpretacja przedziałów dostępności modelu

Lp.	Przedział	Łuk	Opis połączeń
1	$[a_1, b_1]$	(P_1, T_1)	określenie minimalnej a_1 i maksymalnej b_1 dostępności DM do odbierania zgłoszenia od momentu rozpoczęcia dostępności DM przy stanowisku
2	$[a_2, b_2]$	(P_2, T_2)	określenie minimalnego momentu czasu a_2 i maksymalnego b_2 na przyjęcie informacji o zdarzeniu przez DM
3	$[a_3, b_3]$	(P_3, T_3)	określenie minimalnego momentu czasu a_3 i maksymalnego b_3 na podjęcie decyzji o zakwalifikowaniu zgłoszenia jako zdarzenia mnogiego/masowego
4	$[a_4, b_4]$	(P_3, T_4)	określenie minimalnego momentu czasu a_4 i maksymalnego b_4 na podjęcie decyzji o zakwalifikowaniu zgłoszenia jako zwykłego
5	$[a_5, b_5]$	(P_4, T_5)	określenie minimalnego momentu czasu a_5 i maksymalnego b_5 na oddanie funkcji DM-K innemu DM
6	$[a_6, b_6]$	(P_4, T_6)	określenie minimalnego momentu czasu a_6 i maksymalnego b_6 na podjęcie funkcji DM-K przez odbierający DM
7	$[a_7, b_7]$	(P_6, T_7)	określenie minimalnego momentu a_7 po jakim może skończyć się kierowanie akcją przez DM, oraz maksymalnego momentu b_7

Model może być zainicjowany jako:

- DM gotowy do przyjęcia zgłoszenia ze zdolnością ZMM ($m_0(P_1) = 1$, $m_0(P_5) = 1$), lub gotowy bez takiej zdolności ($m_0(P_1) = 1$),
- podejmujący decyzję o kierowaniu akcją ZMM ze zdolnością ($m_0(P_4) = 1$,$m_0(P_6) = 1$) lub z brakiem zdolności ($m_0(P_4) = 1$).

B.2 Określenie parametrów i zainicjowanie modelu systemu

W tak opisanym systemie określono wartości parametrów zgodnie z przyjętymi założeniami:

1. DM jest gotowy do przyjmowania zgłoszenia po upływie $5jc$ od zgłoszenia gotowości oraz co najwyżej do $60jc$ ($[a_1, b_1] = [5, 60]$),
2. czas rozmowy ze świadkiem trwa nie której niż $5jc$ oraz nie dłużej niż $15jc$ ($[a_2, b_2] = [5, 15]$),

3. DM podejmuje decyzję o kwalifikowaniu zdarzenia $5jc$, ale nie dłużej niż $15jc$ ($[a_3, b_3] = [5, 15]$, $[a_4, b_4] = [5, 15]$),

4. DM ma dokładnie $5jc$ na podjęcie decyzji czy będzie pełnił rolę DM-K lub odda funkcję DM-K innemu DM ($[a_5, b_5] = [5, 5]$, $[a_6, b_6] = [5, 5]$), oraz

5. czas pełnienia roli DM-K obejmuje od 120 do $180jc$ ($[a_7, b_7] = [120, 180]$).

6. W momencie 0 DM jest dostępny przy stanowisku i jest zdolny pełnić rolę DM-K co jest reprezentowane przy pomocy funkcji stanu początkowego m_0, gdzie $m_0(P) = 0$ dla $P \in P_1, P_5$ oraz dla pozostałych $m_0(P) = \emptyset$. Oznacza to obecność żetonów w sieci z wiekiem 0 jedynie w miejscach P_1 i P_6.

Symbol jc to skrót od jednostki czasu. Dane dla systemu zostały określone poglądowo i mogą różnić się z rzeczywistymi.

Następnie $N = (P, T, F, C, I, M_0)$ oznacza model systemu c-TdPN, przy czym:

- $P = \{P_1, P_2, P_3, P_4, P_5, P_6\}$,
- $T = \{T_1, T_2, T_3, T_4, T_5, T_6, T_7\}$,
- $F = \{(P_1, T_1), (P_2, T_2), (P_3, T_3), (P_3, T_4), (P_4, T_5), (P_4, T_6), (P_6, T_7),$
  $(T_1, P_2), (T_2, P_3), (T_4, P_1), (T_3, P_4), (T_5, P_1), (T_6, P_6), (T_7, P_1)\}$,
- $C = \{(P_5, T_6)\}$,
- $I = \{((P_1, T_1), [5, 60]), ((P_2, T_2), [5, 15]), ((P_3, T_3), [5, 15]),$
  $((P_3, T_4), [5, 15]), ((P_4, T_5), [5, 5]), ((P_4, T_6), [5, 5]), ((P_6, T_7), [120, 180])\}$,
- $M_0(P_i) = 1$ dla $i \in \{1, 5\}$ oraz $M_0(P_i) = 0$ dla pozostałych.

B.3 Reprezentatywny fragment rozgałęzionego procesu czasowego

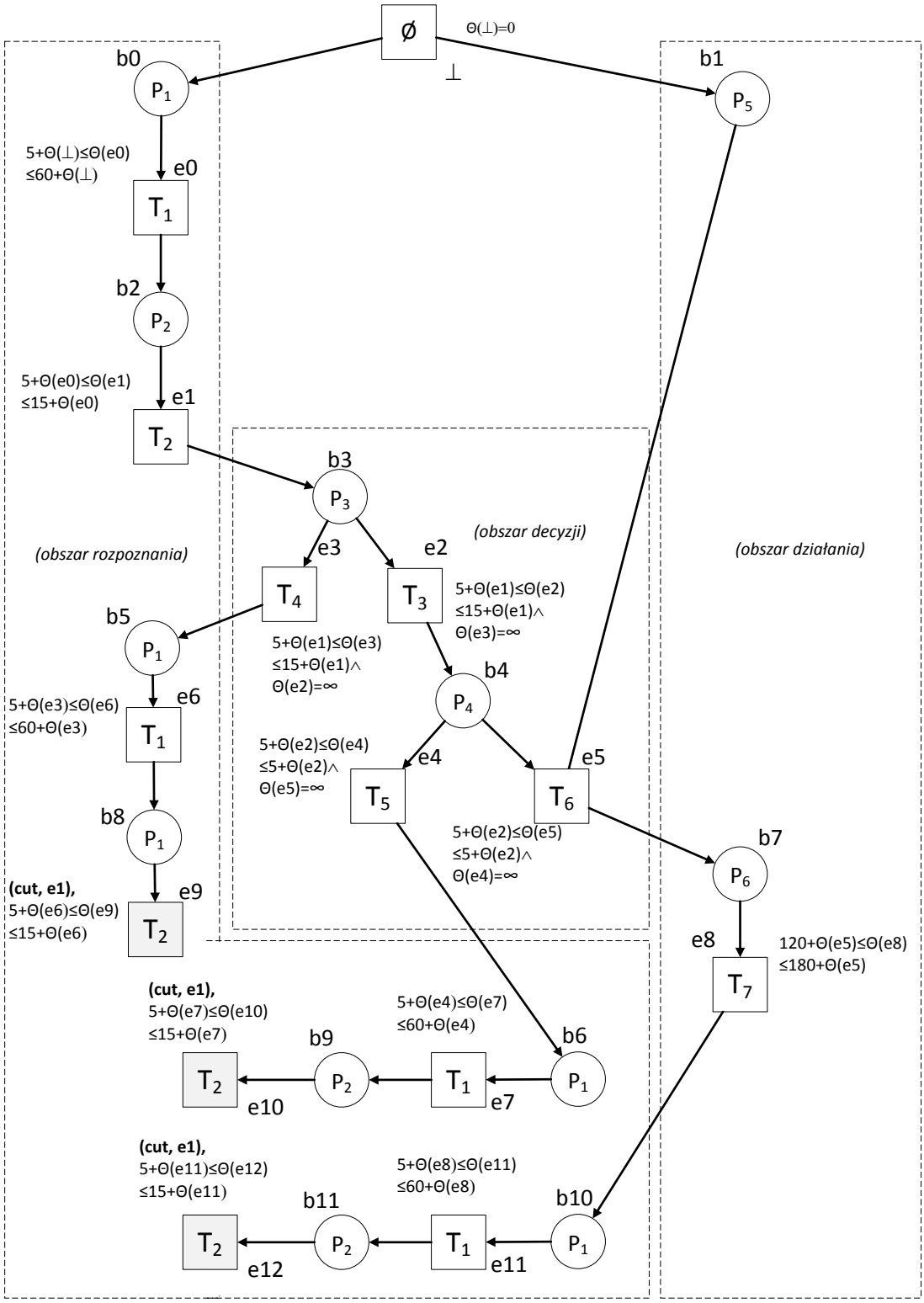
Dla modelu systemu N, przy pomocy algorytmu nr 4, wygenerowano reprezentatywny fragment zachowania (UNF_N, R_N). Składowymi pary są:

- $UNF_N = \{\perp, (b0, \perp), (b1, \perp),$
  $e0, (b2, e0),$
  $e1, (b3, e1),$
  $e2, (b4, e2),$
  $e3, (b5, e3),$
  $e4, (b6, e4),$
  $e5, (b7, e5),$
  $e6, (b8, e6),$
  $e7, (b9, e7),$
  $e8, (b10, e8),$

$e9, e10,$
$e11, (b11, e11),$
$e12\}$

- $R(ei) \in R_N$ dla $i = 0, 1, \ldots, 12$, gdzie:

  - $R(\perp) = \{\theta(\perp) = 0\}$,
  - $R(e0) = \{5 + \theta(\perp) \leqslant \theta(e0) \leqslant 60 + \theta(\perp)\}$,
  - $R(e1) = \{5 + \theta(e0) \leqslant \theta(e1) \leqslant 15 + \theta(e0)\}$,
  - $R(e2) = \{5 + \theta(e1) \leqslant \theta(e2) \leqslant 15 + \theta(e1) \wedge \theta(e3) = \infty\}$,
  - $R(e3) = \{5 + \theta(e1) \leqslant \theta(e3) \leqslant 15 + \theta(e1) \wedge \theta(e2) = \infty\}$,
  - $R(e4) = \{5 + \theta(e2) \leqslant \theta(e4) \leqslant 5 + \theta(e2) \wedge \theta(e5) = \infty\}$,
  - $R(e5) = \{5 + \theta(e2) \leqslant \theta(e5) \leqslant 5 + \theta(e2) \wedge \theta(e4) = \infty\}$,
  - $R(e6) = \{5 + \theta(e3) \leqslant \theta(e6) \leqslant 60 + \theta(e3)\}$,
  - $R(e7) = \{5 + \theta(e4) \leqslant \theta(e7) \leqslant 60 + \theta(e4)\}$,
  - $R(e8) = \{120 + \theta(e5) \leqslant \theta(e8) \leqslant 180 + \theta(e5)\}$,
  - $R(e9) = \{5 + \theta(e6) \leqslant \theta(e9) \leqslant 15 + \theta(e6)\}$,
  - $R(e10) = \{5 + \theta(e7) \leqslant \theta(e10) \leqslant 15 + \theta(e7)\}$,
  - $R(e11) = \{5 + \theta(e8) \leqslant \theta(e11) \leqslant 60 + \theta(e8)\}$,
  - $R(e12) = \{5 + \theta(e11) \leqslant \theta(e12) \leqslant 15 + \theta(e11)\}$,



Rysunek 23: Reprezentatywny fragment zachowania modelu systemu N – rozgałęziony proces czasowy

Graf reprezentatywnego fragmentu przedstawiono na rysunku nr 23.

Zidentyfikowano trzy zdarzenia odcięcia $e9$, $e10$ i $e12$. Przykładowo zdarzenie $e9$ jest powtórnym wykonaniem tranzycji T_2 we wcześniejszym zdarzeniu $e1$. Zdarzenie $e9$ jest zdarzeniem odcięcia ponieważ:

- $\{\lceil e1\rceil < \lceil e9\rceil\}$,
- $l(cut(\lceil e1\rceil)) = l(cut(\lceil e9\rceil)) = \{P_2, P_5\}$,
- $(UNF^1(N), R_1)$ i $(UNF^9(N), R_9)$ mają równoważną przyczynową przyszłość.

przy czym:

- $(UNF^1(N) = \{\bot, (b0, \bot), (b1, \bot), e0, (b2, e0), \}$
- $R_1 = R(\bot), R(e0)$,
- $(UNF^9(N) = \{\bot, (b0, \bot), (b1, \bot),$
  $e0, (b2, e0),$
  $e1, (b3, e1),$
  $e2, (b4, e2),$
  $e3, (b5, e3),$
  $e4, (b6, e4),$
  $e5, (b7, e5),$
  $e6, (b8, e6),$
  $e7, (b9, e7),$
  $e8, (b10, e8)\}$
- $R_9 = R(\bot), R(e0), \ldots, R(e8)$,

Ponadto $cut(UNF^1(N)) = \{b2, b1\}$ oraz $cut(UNF^9(N)) = \{b8, b1\}$. Wystarczy pokazać równość dla wieków odpowiadających sobie warunków:

- $l(b2) = l(b8)$, stąd $age(b2, \theta_1, \{b2, b1\}) = \min\{0, 5\} = 0$ oraz $age(b8, \theta_9, \{b8, b1\}) = \min\{0, 5\} = 0$,
- $l(b1) = l(b1)$, stąd $age(b1, \theta_1, \{b2, b1\}) = \min\{\theta_1(e0), 5\} = 5$ ponieważ $\theta_1(e0) \geqslant 5$ oraz $age(b1, \theta_9, \{b8, b1\}) = \min\{\theta_9(e6), 5\} = 5$ ponieważ $\theta_9(e6) \geqslant 20$.

Stąd $e9$ jest zdarzeniem odcięcia. Analogiczny tok rozumowania można zastosować dla pokazania, że $e10$ i $e12$ są zdarzeniami odcięcia.

W rezultacie trzy zdarzenia odcięcia hamują konstruowanie reprezentatywnego fragmentu zachowania modelu systemu N.

Dysponując modelem z rysunku 4, możliwe jest następnie uzyskanie informacji o zachowaniu modelowanego systemu wraz z przykładem konkretnych momentów wystąpień akcji systemu reprezentowanych przez przebiegi z ograniczeniami momentów wykonań tranzycji w nich zawartych. Dane zestawiono w tabeli nr 4.

B.4 Analiza stanów zapisanych w reprezentatywnym fragmencie do zadanego momentu

Lp	Opis	Przykładowe przebiegi czasowe
1	Do momentu 10 możliwe jest co najwyżej odebranie przez DM informacji o zdarzeniu.	$E_1 = \{\perp, e0, e1\}$, $R_1 = \{R(\perp), R(e0), R(e1)\}$ $R(\perp) = \{\theta_1(\perp) = 0\}$, $R(e0) = \{\theta_1(e0) = 5\}$, $R(e1) = \{\theta_1(e1) = 10\}$
2	Do momentu 15, DM może zakwalifikować odebrane zdarzenie jako zwykłe (a) lub mnogie/masowe (b).	(a) $E_2 = \{\perp, e0, e1, e3\}$, $R_1 = \{R(\perp), R(e0), R(e1), R(e3)\}$ $R(\perp) = \{\theta_1(\perp) = 0\}$, $R(e0) = \{\theta_1(e0) = 5\}$, $R(e1) = \{\theta_1(e1) = 10\}$, $R(e3) = \{\theta_1(e3) = 15 \wedge \theta_1(e2) = \infty\}$ lub (b) $E_3 = \{\perp, e0, e1, e2\}$, $R_1 = \{R(\perp), R(e0), R(e1), R(e2)\}$ $R(\perp) = \{\theta_1(\perp) = 0\}$, $R(e0) = \{\theta_1(e0) = 5\}$, $R(e1) = \{\theta_1(e1) = 10\}$ $R(e3) = \{\theta_1(e2) = 15 \wedge \theta_1(e3) = \infty\}$
3	Do momentu 20, DM może zakwalifikować odebrane zdarzenie jako zwykłe (a) lub przy założeniu, że jest to zdarzenie mnogie/masowe może podjąć się funkcji DM-K (b) lub przekazać pełnienie funkcji DM-K dla innego DM (c).	(a) $E_3 = \{\perp, e0, e1, e3\}$, $R_1 = \{R(\perp), R(e0), R(e1), R(e3)\}$ $R(\perp) = \{\theta_1(\perp) = 0\}$, $R(e0) = \{\theta_1(e0) = 5\}$, $R(e1) = \{\theta_1(e1) = 10\}$ $R(e3) = \{\theta_1(e3) = 15 \wedge \theta_1(e2) = \infty\}$ lub (b) $E_4 = \{\perp, e0, e1, e2, e5\}$, $R_1 = \{R(\perp), R(e0), R(e1), R(e2), R(e5)\}$ $R(\perp) = \{\theta_1(\perp) = 0\}$, $R(e0) = \{\theta_1(e0) = 5\}$, $R(e1) = \{\theta_1(e1) = 10\}$ $R(e2) = \{\theta_1(e2) = 15 \wedge \theta_1(e3) = \infty\}$ $R(e5) = \{\theta_1(e5) = 20 \wedge \theta_1(e4) = \infty\}$ lub (c) $E_5 = \{\perp, e0, e1, e2, e4\}$, $R_1 = \{R(\perp), R(e0), R(e1), R(e2), R(e4)\}$ $R(\perp) = \{\theta_1(\perp) = 0\}$, $R(e0) = \{\theta_1(e0) = 5\}$, $R(e1) = \{\theta_1(e1) = 10\}$, $R(e2) = \{\theta_1(e2) = 15 \wedge \theta_1(e3) = \infty\}$, $R(e4) = \{\theta_1(e4) = 20 \wedge \theta_1(e5) = \infty\}$

Może zdarzyć się także, że do momentu 20 DM rozpocznie odbieranie kolejnego zgłoszenia jeżeli poprzednie zakwalifikuje jako zwykłe w najkrótszym możliwym czasie. Sytuacja ta jest reprezentowana w (UNF_N, R_N) przez zdarzenie $e6$. Aczkolwiek jeżeli DM zakwalifikuje odebranie zdarzenie jako mnogie/masowe, odda kierowanie akcją

innemu DM oraz zrobi to w najkrótszym możliwym czasie (do momentu 20) to także będzie mógł rozpocząć przyjmowanie kolejnego zgłoszenia ($e7$).

Ponadto zbiory warunków $\{b2, b1\}$, $\{b8, b1\}$, $\{b9, b1\}$ i $\{b10, b1\}$, pomijając aspekt czasu modelu, odpowiadają temu samemu stanowi systemu, gdy DM jest w stanie czuwania i jest zdolny pełnić funkcję DM-K.

Zastosowanie algorytmu nr 4 do konstruowania fragmentu rozgałęzionego procesu czasowego umożliwia zapanowanie nad eksplozją stanów poprzez łączenie nakładającej się części strukturalnej przebiegów na grafie (np. $\{e0, e1, e2\} \cap \{e0, e1, e3\} = \{e0, e1\}$). Umożliwia także rozpoznanie powtarzających się połączeń pomiędzy kolejnymi wykonaniami w obrębie przebiegu czasowego.

Mając do dyspozycji skonstruowany model zachowania pokazano przykładowe informacje jakie można uzyskać o modelowanym systemie. W ogólności są to odpowiedzi na pytania:

- Co może zdarzyć się w systemie do pewnego momentu lub pomiędzy pewnymi momentami?

Dla nieskomplikowanych i niedużych modeli możliwe jest formowanie wniosków na podstawie skonstruowanego grafu rozgałęzionego procesu czasowego przedstawiającego początkowy fragment zachowania systemu. Pomimo, iż reprezentacja zachowania przy pomocy rozgałęzionego procesu czasowego rozwiązuje częściowo problem eksplozji stanów to duże modele systemów są uciążliwe do analizowania kartki i ołówka.