

Marcin Szpyrka

sieci Petriego w modelowaniu i analizie systemów współbieżnych

Spis treści

Przedmowa	1
Rozdział 1. Wprowadzenie	3
Rozdział 2. Struktura sieci	9
2.1. Grafy skierowane	9
2.2. Sieci i ich reprezentacja graficzna	13
2.3. Własności strukturalne sieci	15
2.4. Podsumowanie	19
Rozdział 3. Sieci znakowane	21
3.1. Dynamika sieci	21
3.2. Ograniczoność sieci	24
3.3. Żywotność sieci	26
3.4. Charakterystyczne konstrukcje sieciowe	30
3.5. Modelowanie z zastosowaniem sieci znakowanych	35
3.6. Podsumowanie	44
Rozdział 4. Sieci uogólnione	45
4.1. Dynamika sieci	45
4.2. Modelowanie z zastosowaniem sieci uogólnionych	47
4.3. Podsumowanie	50
Rozdział 5. Analiza sieci uogólnionych	51
5.1. Grafy osiągalności	52
5.2. Grafy pokrycia	59
5.3. Grafy silnie spójnych składowych	64

5.4. Symetrie	69
5.5. Trwałe zbiory przejść	77
5.6. Algebraiczna reprezentacja sieci	82
5.7. Niezmienniki miejsc	84
5.8. Niezmienniki przejść	94
5.9. Pułapki i zatrzaski	99
5.10. Metody redukcji sieci	110
5.11. Podsumowanie	116
Rozdział 6. Sieci miejsc i przejść	119
6.1. Dynamika sieci miejsc i przejść	119
6.2. Modelowanie z zastosowaniem sieci miejsc i przejść	121
6.3. Symulowanie PT-sieci za pomocą sieci uogólnionych	123
6.4. Podsumowanie	126
Rozdział 7. Sieci priorytetowe	127
7.1. Dynamika sieci priorytetowych	127
7.2. Modelowanie z zastosowaniem sieci priorytetowych	129
7.3. Podsumowanie	137
Rozdział 8. Sieci czasowe	139
8.1. Proste sieci czasowe	140
8.2. Przedziałowe sieci czasowe	144
8.3. Modelowanie z zastosowaniem sieci czasowych	147
8.4. Podsumowanie	150
Rozdział 9. Analiza sieci czasowych	151
9.1. Grafy osiągalności dla prostych sieci czasowych	151
9.2. Grafy klas stanów	155
9.3. Podsumowanie	160
Rozdział 10. Sieci kolorowane	161
10.1. Wielozbiory	162
10.2. Niehierarchiczne sieci kolorowane	163
10.3. Ograniczoność sieci	168
10.4. Żywotność sieci	172
10.5. Hierarchiczne sieci kolorowane	175
10.6. Modelowanie z zastosowaniem sieci kolorowanych	179
10.7. Podsumowanie	185

Rozdział 11. Analiza sieci kolorowanych	187
11.1. Grafy osiągalności	187
11.2. Grafy osiągalności z klasami równoważności	192
11.3. Niezmienniki sieci kolorowanych	196
11.4. Podsumowanie	204
Rozdział 12. Czasowe sieci kolorowane	205
12.1. Wielozbiory czasowe	205
12.2. Dynamika sieci	207
12.3. Modelowanie z zastosowaniem czasowych sieci kolorowanych	213
12.4. Podsumowanie	218
Rozdział 13. RTCP-sieci	219
13.1. Niekierunkowe RTCP-sieci	219
13.2. Kierunkowe RTCP-sieci	227
13.3. Modelowanie z zastosowaniem RTCP-sieci	239
13.4. Podsumowanie	246
Rozdział 14. Analiza RTCP-sieci	247
14.1. Grafy osiągalności	247
14.2. Grafy pokrycia	251
14.3. Podsumowanie	255
Rozdział 15. Zakończenie	257
Dodatek A. Polsko-angielski słownik terminów	259
Dodatek B. Angielsko-polski słownik terminów	265
Dodatek C. Wybrane narzędzia do konstruowania i analizy sieci Petriego	271
Dodatek D. Lista symboli	273
Literatura	277
Skorowidz	283

Przedmowa

Pomysł napisania tej książki zrodził się kilka lat temu, a konkretnie w 2000 roku, kiedy obroniłem pracę doktorską, której tematyka była związana z kolorowanymi sieciami Petriego. Rok wcześniej ukazała się książka [74], której jestem współautorem. Zawierała ona przegląd klas sieci Petriego niskiego poziomu. Brakowało jednak w literaturze książki, w której przedstawiono by wybrane klasy sieci Petriego niskiego i wysokiego poziomu z zachowaniem jednolitych oznaczeń i uwypukleniem różnic między tymi klasami. Postanowiłem tę lukę wypełnić, choć realizacja tych ambitnych planów się odsuwała, głównie z powodu różnych obowiązków naukowych i dydaktycznych. W tym czasie prowadziłem badania dotyczące systemów czasu rzeczywistego. Interesowały mnie możliwości zastosowania sieci Petriego do modelowania i analizy systemów wbudowanych, a zwłaszcza własności czasowych takich systemów.

Udało mi się w końcu doprowadzić do skutku moje zamierzenia. Książka ta nie zawiera przeglądu wszystkich klas sieci Petriego, nie jest też wyczerpującym studium na temat klas w niej opisanych. Zależało mi bowiem na przedstawieniu zarysu teorii sieci Petriego przez ukazanie różnorodności klas tych sieci, metod ich formalnej analizy oraz możliwości ich zastosowania do modelowania systemów współbieżących i systemów czasu rzeczywistego. Miała to być specyficzna *podróż* przez różne klasy sieci, począwszy od najprostszych, skończywszy na bardzo rozbudowanych z możliwościami hierarchizacji opisu sieci.

Wszystkie omawiane w książce klasy sieci Petriego zostały opisane z podaniem ich formalnej definicji, z przedstawieniem ich najważniejszych własności oraz metod ich analizy. Główny nacisk położyłem na zaprezentowanie przykładów zastosowania poszczególnych klas sieci, zwracając uwagę zwłaszcza na formalną analizę ich własności. W kilku wypadkach ten sam system rozważam przy okazji omawiania różnych klas, tak aby zwrócić uwagę na różne możliwości rozmaitych klas sieci Petriego.

Podziękowania

Chciałbym podziękować kilku osobom, które w różny sposób wpłynęły na powstanie tej książki czy też na jej ostateczny kształt. Przede wszystkim dziękuję prof. Tomaszowi Szmucowi z Katedry Automatyki Akademii Górnictwo-Hutniczej, promotorowi mojej pracy doktorskiej i kierownikowi Laboratorium Informatyki, w którym pracuję. Przez wszystkie lata naszej współpracy mogłem zawsze liczyć na Jego pomoc i cenne uwagi, które często decydowały o kształcie mojej pracy naukowej. Podziękowania kieruję również do prof. Zbigniewa Suraja, promotora mojej pracy magisterskiej i kierownika Katedry Informatyki Uniwersytetu Rzeszowskiego. Był pierwszą osobą, dzięki której zacząłem interesować się sieciami Petriego.

Wyrazy wdzięczności kieruję również do recenzentów tej książki – prof. Andrzeja Adamskiego (Akademia Górnictwo-Hutnicza w Krakowie), prof. Mariana Adamskiego (Uniwersytet Zielonogórski) i prof. Jana Magotta (Politechnika Wrocławskiego). Skorzystałem z ich cennych uwag, co pozwoliło mi uniknąć wielu niedociągnięć.

Chciałbym również podziękować moim kolegom, którzy często wspierali mnie w pracy naukowej. Wielu z nich było pierwszymi czytelnikami tej książki, a ich trafne spostrzeżenia miały wpływ na jej ulepszenie. Do osób, którym jestem winien szczególną wdzięczność, należą: prof. Antoni Ligęza, dr inż. Konrad Kułakowski, dr inż. Grzegorz J. Nalepa (wszyscy z AGH Kraków), dr inż. Sławomir Samolej (Politechnika Rzeszowska) i mgr inż. Piotr Matyasiak (AGH Kraków).

Szczególnie goraco dziękuję moim najbliższym, żonie Agnieszce oraz dzieciom, Klaudii i Mateuszowi, głównie za ich cierpliwość i wyrozumiałość, kiedy podczas pisania tej książki stawałem się na długie godziny *nieobecny*, jak również za okazaną mi pomoc. Moja żona należała do pierwszych czytelników tej książki, a jej liczne konstruktywne uwagi miały duży wpływ na ostateczny kształt tego opracowania.

Kraków, styczeń 2007

Wprowadzenie

Sieci Petriego są graficznym i matematycznym narzędziem, stosowanym w wielu różnych dziedzinach nauki. Charakteryzuje je intuicyjny graficzny język modelowania, wspierany przez zaawansowane metody formalnej analizy ich własności. Naturalnym zjawiskiem w sieciach Petriego jest współbieżność wykonywanych akcji; dlatego też są one najczęściej postrzegane jako matematyczne narzędzie służące do modelowania systemów współbieżnych (zob. prace: [32], [70], [74] i [93]). Sieci, których definicję rozbudowano o model czasu, mogą być stosowane do modelowania systemów czasu rzeczywistego (zob. prace: [11], [42], [75], [76] i [77]).

Teoria sieci Petriego wzięła początek od pracy K. A. Petriego [51] opublikowanej w 1962 roku, a swą nazwę – od nazwiska autora tej pracy. W wyniku ponad czterdziestoletniego rozwoju tej teorii powstało wiele różnych klas sieci i rozszerzono zakres ich zastosowań. W zależności od potrzeb definicję sieci zmieniano i dostosowywano do rozważanego problemu. Na wprowadzanie takich modyfikacji pozwala niezwykła *elastyczność* tej teorii. Dzięki różnorodności znanych obecnie klas sieci można stosunkowo łatwo dobrać klasę najwłaściwszą dla danej dziedziny zastosowań. Różnorodność ta jednak utrudnia opracowanie jednolitych metod analizy różnych klas sieci.

Sieć Petriego jest przedstawiana jako graf dwudzielny. Może ona mieć strukturę hierarchiczną, znacznie ułatwiającą modelowanie złożonych systemów. Model w postaci sieci Petriego jest nie tylko graficzną reprezentacją danego systemu. Możliwa jest bowiem symulacja pracy sieci (najczęściej wspomagana przez odpowiednie narzędzia komputerowe), dzięki której model taki staje się wirtualnym prototypem reprezentowanego systemu.

Oprócz reprezentacji graficznej i możliwości symulacji pracy, na uwagę zasługuje rozbudowana teoria, która jest podstawą formalnej analizy własności sieci Petriego. Różnorodność klas sieci powoduje, że metody ich analizy znacznie się różnią. Dla sieci o najprostszej strukturze istnieje najszerszy wachlarz tych metod i są one stosunkowo łatwe w użyciu. W przypadku sieci najbardziej rozbudowanych

możliwości formalnej analizy są zazwyczaj uboższe lub trudniejsze do zastosowania. W książce występuje kilkanaście przykładów modeli w postaci sieci Petriego. Omówienie każdego modelu zawiera wybrane wyniki jego analizy.

Dla kogo jest ta książka

Niniejsza książka jest przeznaczona przede wszystkim dla tych osób, które chcą stosować sieci Petriego w badaniach naukowych lub w praktyce inżynierskiej. Jej lektura powinna ułatwić wybór klasy sieci najbardziej odpowiadającej ich wymaganiom. Książka może też posłużyć wykładowcom przedmiotów, które częściowo dotyczą sieci Petriego. Najliczniejsze grono czytelników będą zapewne stanowić studenci, zarówno studiów inżynierskich i magisterskich, jak i doktoranckich, na których są wykładane elementy teorii sieci Petriego. Mam nadzieję, że książka pomoże poszerzyć ich wiedzę z tego zakresu.

Pisząc tę książkę, przyjąłem założenie, że czytelnik zna elementy matematyki wyższej wykładanej na pierwszym roku studiów technicznych. Dotyczy to zwłaszcza następujących działów matematyki: algebry zbiorów, algebry liniowej (w tym rachunku macierzowego) i teorii grafów.

Przyjęta konwencja notacyjna

Różnorodność istniejących klas sieci Petriego, a także narzędzi komputerowych umożliwiających ich użytkowanie, spowodowała, że nie istnieje jednolita notacja odnosząca się do zarówno graficznej reprezentacji sieci, jak i oznaczeń jej poszczególnych elementów. Notacje graficzne sieci różnią się w zależności od wybranego oprogramowania, ale różnice te nie są zbyt zasadnicze. Sytuacja wygląda inaczej, w przypadku formalnego opisu sieci, gdyż nawet w obrębie jednej klasy stosuje się różne notacje. W książce próbowałem wprowadzić jednolitą notację, zarówno przedstawiając graf sieci, jak i formułując definicje.

Internet zapewnia dostęp do wielu narzędzi komputerowych, umożliwiających modelowanie z zastosowaniem sieci Petriego. Nie istnieje jednak oprogramowanie służące jednocześnie do modelowania wszystkich omawianych klas sieci, toteż żadnego w szczególny sposób nie polecam. Dodatek C zawiera informacje o narzędziach użytych w omawianych przykładach.

Struktura książki

Książka składa się z piętnastu rozdziałów, w których opisałem dziewięć klas sieci Petriego. Na początku omawiam klasy najprostsze, a następnie klasy będące ich pochodnymi, zazwyczaj o bardziej rozbudowanej definicji.

Rozdział 2 zawiera krótkie wprowadzenie do teorii grafów oraz przegląd własności strukturalnych sieci Petriego. Większość zdefiniowanych w nim pojęć zachowuje ważność przy analizie dowolnej klasy sieci. W rozdziale tym pominąłem całkowicie aspekt dynamiki sieci. Zasady pracy sieci Petriego, które są zależne od ich typu, przedstawiłem w kolejnych rozdziałach, omawiając poszczególne klasy sieci.

Literatura: [19], [23], [48], [51], [56], [58], [70], [73], [74].

Rozdział 3 zawiera opis *sieci znakowanych*, będących najprostszą klasą sieci Petriego. Zdefiniowałem w nim podstawowe pojęcia dotyczące własności dynamicznych sieci. Wyróżniłem dwie grupy tych własności – związane z ograniczonością miejsc sieci i żywotnością jej przejść. Właściwości te występują w większości klas sieci przedstawionych w książce. W rozdziale tym omówiłem podstawowe zasady konstruowania modeli w postaci sieci Petriego, a także przykłady modelowania z zastosowaniem sieci znakowanych.

Literatura: [3], [14], [19], [23], [41], [48], [50], [51], [52], [56], [70], [72], [74], [75], [76].

Rozdział 4 dotyczy podstawowej klasy sieci niskiego poziomu, tzw. *sieci uogólnionych*. Umożliwiają one usuwanie lub dodawanie do jednego miejsca większej liczby znaczników podczas wykonania pojedynczego przejścia. Uzyskuje się to przez przypisanie łukom wag, które są interpretowane jako ich krotność. Rozdział zawiera również przykłady modelowania z zastosowaniem tej klasy sieci.

Literatura: [17], [19], [48], [50], [52], [56], [70], [72], [74], [80], [89], [91].

Rozdział 5 zawiera opis najczęściej spotykanych metod analizy sieci uogólnionych. Stanowi on główną część książki, poświęconą metodom analizy sieci Petriego. Opisywane w następnych rozdziałach metody analizy innych klas sieci są zazwyczaj porównywane z metodami przedstawionymi w rozdz. 5.

Prezentowane metody analizy można podzielić na kilka grup. Pierwszą stanowią metody oparte na analizie zbioru osiągalnych znakowań, który jest reprezentowany w postaci grafu skierowanego. Reprezentacja taka może być zarówno pełnym grafem osiągalności, w którym każde osiągalne znakowanie sieci jest reprezentowane przez jeden węzeł, jak również grafem zredukowanym, w którym jeden węzeł reprezentuje niepusty zbiór osiągalnych znakowań.

Druga grupa metod analizy jest związana z algebraiczną reprezentacją sieci. Strukturę sieci uogólnionych można przedstawić wzajemnie jednoznacznie w postaci dwóch macierzy o współczynnikach całkowitych. Reprezentacja ta służy do wyznaczenia tzw. macierzy incydencji, na której podstawie oblicza się wartości niezmienników sieci. Wyrażają one stałe własności modelu i można je stosować zarówno przy analizie ograniczości sieci (niezmienniki miejsc), jak i jej żywotności (niezmienniki przejść).

Ponadto rozdział zawiera również opisy metod, w których korzysta się z istnienia specyficznych zbiorów miejsc, określanych mianem zatrasków i pułapek, a także metod redukujących rozmiar sieci. Analiza istnienia pułapek i zatrasków pozwala zidentyfikować podzbiory miejsc, które zawsze pozostają znakowane albo zawsze pozostają czyste. Metody redukcji pozwalają zmniejszyć liczbę elementów sieci z zachowaniem wybranych jej własności, takich jak: żywotność sieci, bezpieczeństwo, brak znakowań martwych w zbiorze znakowań osiągalnych sieci itp.

Literatura: [2], [4], [6], [12], [14], [18], [20], [23], [28], [30], [32], [37], [42], [44], [45], [48], [50], [54], [56], [57], [61], [62], [63], [64], [66], [67], [68], [69], [70], [72], [74], [85], [86], [87], [88], [92].

Rozdział 6 zawiera opis sieci *miejsc i przejść* (PT-sieci), które są jedną z najczęściej spotykanych w literaturze klas sieci Petriego. W porównaniu z sieciami uogólnionymi w sieciach miejsc i przejść definiuje się funkcję ograniczającą pojemność miejsc. Rozdział zawiera również przykład modelowania z zastosowaniem tej klasy sieci.

Literatura: [48], [50], [51], [52], [56], [70], [74].

Rozdział 7 zawiera opis sieci *priorytetowych* oraz przykłady modelowania z ich zastosowaniem. W porównaniu z sieciami miejsc i przejść, w definicji sieci priorytetowych występuje dodatkowo funkcja priorytetów określona na zbiorze przejść. Użycie priorytetów pozwala na (częściowo) deterministyczny wybór przejścia do wykonania w sytuacjach konfliktowych.

Literatura: [19], [24], [43], [50], [55].

Rozdział 8 dotyczy dwóch klas sieci czasowych. Wprowadzenie modelu czasu do sieci Petriego umożliwia stosowanie ich do modelowania systemów czasu rzeczywistego. W rozdziale tym omawiam dwie klasy: *proste sieci czasowe* i *przedziałowe sieci czasowe*. Pierwszą wybrałem ze względu na prostotę oraz intuicyjność przedstawionych w niej rozwiązań, a druga jest jedną z najczęściej cytowanych klas czasowych sieci Petriego. Rozdział ten zawiera również przykłady modelowania z zastosowaniem tych klas sieci.

Literatura: [1], [5], [6], [7], [8], [9], [11], [15], [16], [20], [21], [25], [26], [27], [46], [47], [53], [54], [57], [59], [60], [64], [67], [68], [75], [87], [88], [93].

Rozdział 9 dotyczy metod analizy czasowych sieci Petriego. Omówiłem w nim grafy osiągalności dla prostych sieci czasowych i grafy klas dla sieci przedziałowych. Wykorzystanie tych metod pozwala analizować wszystkie podstawowe własności sieci, w tym również własności czasowe.

Literatura: [1], [5], [6], [7], [9], [20], [53], [57], [64], [67], [68], [75], [85], [86], [87], [92].

Rozdział 10 zawiera opis *sieci kolorowanych*. Są one jedną z najbardziej popularnych klas sieci Petriego wysokiego poziomu. Zasadniczą różnicą między sieciami uogólnionymi a sieciami kolorowanymi jest możliwość definiowania w sieciach kolorowanych różnych typów znaczników, które odpowiadają typom danych występującym w popularnych językach programowania. Rozdział zawiera również omówienie mechanizmów konstruowania hierarchicznych sieci kolorowanych oraz przykłady modelowania z zastosowaniem tej klasy sieci.

Literatura: [13], [29], [31], [32], [33], [34], [38], [65], [75].

Rozdział 11 zawiera omówienie metod analizy sieci kolorowanych. Konieczność uwzględnienia różnych typów znaczników powoduje, że metod przedstawionych w rozdz. 5 nie można bezpośrednio stosować do sieci wysokiego poziomu. W rozdziale tym przedstawiłem grafy osiągalności, grafy relacyjne oraz niezmienikи miejsc i przejść stosowane do analizowania sieci kolorowanych.

Literatura: [13], [32], [35], [36], [39], [40], [57], [84], [85].

Rozdział 12 zawiera opis czasowych kolorowanych sieci Petriego. Stanowią one rozszerzenie sieci kolorowanych o model czasu. W porównaniu z sieciami nieczasowymi, znacznik – oprócz wartości odpowiedniego typu – ma przypisaną również tzw. *pieczętkę czasową*, której wartość decyduje o jego dostępności. W rozdziale tym omówiłem podstawowe pojęcia związane z dynamiką czasowych sieci kolorowanych, a także zamieściłem przykłady modelowania z zastosowaniem tej klasy sieci.

Literatura: [13], [32], [35], [49].

Rozdział 13 dotyczy podklasy sieci kolorowanych, zwanych RTCP-sieciami. Ta opracowana przeze mnie podklasa sieci została przystosowana do modelowania wbudowanych systemów czasu rzeczywistego. W porównaniu z sieciami kolorowanymi RTCP-sieci charakteryzują się m.in. zupełnie odmiennym modelem czasu, występowaniem priorytetów przejść i zmienionym podejściem do tworzenia mo-

deli hierarchicznych. Wprowadzone zmiany miały ułatwić i przyspieszyć tworzenie modeli oraz umożliwić bezpośrednie modelowanie elementów typowych dla programowania współbieżnego, takich jak przeterminowania, priorytety zadań itp. Rozdział zawiera definicję niehierarchicznych RTCP-sieci oraz postaci kanonicznej dla modeli hierarchicznych, a także przykłady modelowania z zastosowaniem tej klasy sieci.

Literatura: [10], [22], [32], [71], [77], [81], [83].

Rozdział 14 dotyczy metod analizy charakterystycznych dla RTCP-sieci. Omówiłem w nim grafy osiągalności dla tej klasy sieci oraz przedstawiłem koncepcję pokrywania się stanów i grafy pokrycia. Jedną z zasadniczych cech RTCP-sieci jest fakt, że w wielu przypadkach można dla nich zbudować skończony graf pokrycia, umożliwiający weryfikację większości własności modelu, w tym również zależności czasowych.

Literatura: [32], [77], [79], [81].

Rozdział 15 zawiera podsumowanie zagadnień rozważanych w książce.

Dodatek A zawiera polsko-angielski słownik najważniejszych pojęć związanych z teorią sieci Petriego.

Dodatek B zawiera angielsko-polski słownik najważniejszych pojęć związanych z teorią sieci Petriego.

Dodatek C zawiera opis wybranych narzędzi komputerowych, wspierających konstruowanie i analizę sieci Petriego.

Dodatek D zawiera spis symboli występujących w tekście książki.

Wszelkie opinie i uwagi na temat książki proszę kierować pod adresem mszpyrka@agh.edu.pl.



Struktura sieci

Jedną z zasadniczych cech sieci Petriego jest ich stosunkowo prosta i intuicyjna reprezentacja graficzna, wspólna dla wszystkich klas sieci. Sieci Petriego są przedstawiane w postaci grafów skierowanych, zawierających dwa rodzaje węzłów (grafy dwudzielne) nazywanych miejscami i przejściami. Tych dwóch rodzajów węzłów używa się odpowiednio do reprezentowania stanów i aktywności modelowanego systemu.

Łuki w grafie sieci mogą łączyć tylko węzły różnych typów. Ze względu na charakter tych połączeń definiuje się sieci o rozmaitych własnościach strukturalnych, m. in.: sieci czyste, sieci proste, sieci swobodnego wyboru, grafy synchronizacji, maszyny stanowe itd. (zob. prace: [19], [48], [56], [70] i [74]).

Rozdział ten zawiera krótkie wprowadzenie do teorii grafów oraz omówienie własności strukturalnych sieci Petriego. Większość zdefiniowanych pojęć zachowuje ważność dla dowolnej klasy sieci. W tym rozdziale pominiemy całkowicie aspekt dynamiki sieci. Zasadami pracy sieci Petriego, które zależą od ich typu, zajmiemy się w następnych rozdziałach przy okazji omawiania poszczególnych klas sieci.

2.1. Grafy skierowane

W tym podrozdziale omówimy podstawowe pojęcia związane z grafami skierowanymi, których używa się do opisu i analizy sieci Petriego. Bardziej szczegółowe wprowadzenie do teorii grafów można znaleźć na przykład w pracach [58] i [90].

Definicja 2.1. *Grafem skierowanym (grafem)* nazywamy uporządkowaną trójkę postaci $\mathcal{G} = (V, A, \gamma)$, przy czym:

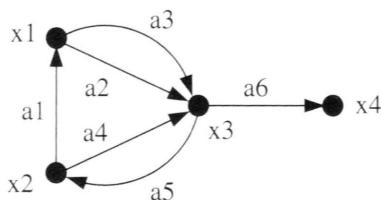
- (1) V jest zbiorem węzłów grafu.
- (2) A jest zbiorem łuków (krawędzi) grafu, takim że $V \cap A = \emptyset$.
- (3) $\gamma: A \rightarrow V \times V$ jest funkcją zaczepienia, która każdemu łukowi przypisuje uporządkowaną parę węzłów.

Jeżeli zbiory węzłów i łuków grafu \mathcal{G} są zbiorami skończonymi, to graf \mathcal{G} nazywamy *grafem skończonym*. ■

Graf skierowany jest najczęściej przedstawiany w postaci wykresu, który składa się z punktów reprezentujących elementy zbioru V oraz strzałek reprezentujących elementy zbioru A , takich że jeżeli $x, y \in V$, $a \in A$ i $\gamma(a) = (x, y)$, to strzałka odpowiadająca łukowi a prowadzi od punktu oznaczonego przez x do punktu oznaczonego przez y . Węzeł x nazywamy wówczas *węzłem początkowym (początkiem)*, a węzeł y , *węzłem końcowym (końcem)* łuku a .

Na rysunku 2.1 przedstawiono przykład grafu skierowanego $\mathcal{G} = (V, A, \gamma)$, dla którego:

- $V = \{x_1, x_2, x_3, x_4\}$;
- $A = \{a_1, a_2, a_3, a_4, a_5, a_6\}$;
- $\gamma(a_1) = (x_2, x_1)$, $\gamma(a_2) = (x_1, x_3)$, $\gamma(a_3) = (x_1, x_3)$, $\gamma(a_4) = (x_2, x_3)$, $\gamma(a_5) = (x_3, x_2)$, $\gamma(a_6) = (x_3, x_4)$.



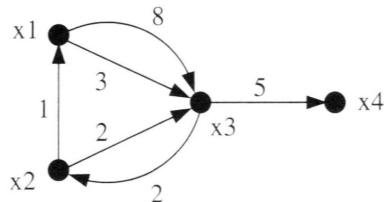
Rys. 2.1. Graf skierowany

Powyższa definicja dopuszcza istnienie więcej niż jednego łuku prowadzącego od węzła x do węzła y . Grafy takie nazywa się niekiedy *multigrafami*. W rozważanym grafie węzły x_1 i x_3 są połączone dwoma łukami prowadzącymi od x_1 do x_3 . Jeżeli założymy, że funkcja zaczepienia jest funkcją różnowartościową, to graf można zdefiniować jako parę $\mathcal{G} = (V, A)$, przy czym $A \subseteq V \times V$. Będziemy odtąd *grafem* nazywać zarówno graf zdefiniowany jako trójka $\mathcal{G} = (V, A, \gamma)$, jak i przedstawiony jako para $\mathcal{G} = (V, A)$. Z kontekstu będzie jednoznacznie wynikać, czy dopuszczalne są łuki wielokrotne.

Tworząc wykres grafu skierowanego, pomija się zazwyczaj oznaczenia łuków. Często natomiast łuki grafu mają przypisane etykiety, których interpretacja zależy od interpretacji grafu.

Definicja 2.2. Graf skierowany $\mathcal{G} = (V, A, \gamma)$ nazywamy *grafem etykietowanym nad zbiorem etykiet* L , jeżeli łuki grafu \mathcal{G} mają przypisane etykiety ze zbioru L . ■

Przykład grafu etykietowanego nad zbiorem \mathbb{N} przedstawiono na rys. 2.2.



Rys. 2.2. Graf etykietowany

Definicja 2.3. Niech $\mathcal{G} = (V, A, \gamma)$ będzie grafem skierowanym. Dla dowolnego węzła $x \in V$ elementy zbioru

$$In(x) = \{y \in V : \exists a \in A \wedge \gamma(a) = (y, x)\} \quad (2.1)$$

nazywamy *poprzednikami* węzła x , natomiast elementy zbioru:

$$Out(x) = \{y \in V : \exists a \in A \wedge \gamma(a) = (x, y)\} \quad (2.2)$$

nazywamy *następnikami* węzła x . Zbiory $In(x)$ i $Out(x)$ nazywamy odpowiednio *zbiorem poprzedników* i *zbiorem następników* węzła x . ■

Jeżeli zbiór poprzedników węzła $x \in V$ jest zbiorem pustym, to x nazywamy *węzłem początkowym*, jeżeli natomiast zbiór następników jest zbiorem pustym, to x nazywamy *węzłem końcowym* grafu \mathcal{G} . Jeżeli zbiory poprzedników i następników węzła x są zbiorami pustymi, to x nazywamy *węzłem izolowanym*.

Definicję 2.3 można rozszerzyć na zbiór węzłów sieci. Jeżeli $V' \subseteq V$, to zbiory $In(V')$ i $Out(V')$ definiuje się następująco:

$$In(V') = \bigcup_{x \in V'} In(x), \quad (2.3)$$

$$Out(V') = \bigcup_{x \in V'} Out(x). \quad (2.4)$$

Definicja 2.4. Niech $\mathcal{G} = (V, A, \gamma)$ będzie grafem skierowanym.

- (1) Ciąg łuków $a_1, a_2, \dots, a_n \in A$ nazywamy *drogą nieskierowaną* w grafie \mathcal{G} , jeżeli istnieją węzły $x_1, x_2, \dots, x_{n+1} \in V$, takie że $\gamma(a_i) = (x_i, x_{i+1})$ lub $\gamma(a_i) = (x_{i+1}, x_i)$ dla dowolnego $i = 1, 2, \dots, n$. O drodze nieskierowanej mówimy wówczas, że prowadzi od węzła x_1 do węzła x_{n+1} .
- (2) Ciąg łuków $a_1, a_2, \dots, a_n \in A$ nazywamy *drogą (skierowaną)* w grafie \mathcal{G} , jeżeli istnieją węzły $x_1, x_2, \dots, x_{n+1} \in V$, takie że $\gamma(a_i) = (x_i, x_{i+1})$ dla dowolnego $i = 1, 2, \dots, n$. O drodze mówimy wówczas, że prowadzi od węzła x_1 do węzła x_{n+1} .
- (3) Drogę nazywamy *skończoną*, jeżeli dla pewnego $n \in \mathbb{N}$ nie ma już określonego a_{n+1} . Droga ma wówczas długość n .
- (4) Drogę nazywamy *pustą*, jeżeli jest skończona i ma długość 0.
- (5) Drogę nazywamy *cyklem*, jeżeli jest skończona, ma długość n oraz $x_{n+1} = x_1$.

Rozważmy graf przedstawiony na rys. 2.1. Graf ten zawiera jeden węzeł końcowy $x4$. Przykładem drogi w tym grafie jest ciąg łuków $a1, a3, a6$. Droga ta ma długość 3 i jednoznacznie wyznacza ciąg węzłów $x2, x1, x3, x4$. Przykładem cyklu jest droga $a4, a5$. Ponieważ graf zawiera cykle, więc można wskazać również drogę nieskończoną, na przykład $a4, a5, a4, a5, \dots$.

Definicja 2.5. Niech $\mathcal{G} = (V, A, \gamma)$ będzie grafem skierowanym.

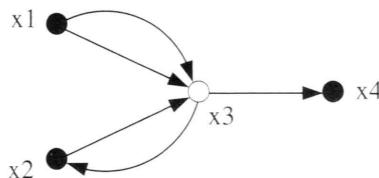
- (1) Graf \mathcal{G} nazywamy *grafem acyklicznym*, jeżeli nie zawiera cykli.
- (2) Graf \mathcal{G} nazywamy *grafem spójnym*, jeżeli dla dowolnych dwóch węzłów x i y istnieje w grafie droga nieskierowana od x do y .
- (3) Graf \mathcal{G} nazywamy *grafem silnie spójnym*, jeżeli dla dowolnych dwóch węzłów x i y istnieje w grafie droga skierowana od x do y .

Graf przedstawiony na rys. 2.1 nie jest grafem acyklicznym. Jest on grafem spójnym, ale nie jest silnie spójny, ponieważ nie istnieje droga od węzła $x4$ do żadnego innego węzła tego grafu.

Definicja 2.6. Graf skierowany $\mathcal{G} = (V, A, \gamma)$ nazywamy *grafem dwudzielnym*, jeżeli zbiór V jest sumą dwóch rozłącznych zbiorów V_1, V_2 i dowolny łuk grafu \mathcal{G} łączy węzły należące do różnych zbiorów, tzn.

$$\forall a \in A : \gamma(a) \in (V_1 \times V_2) \cup (V_2 \times V_1). \quad (2.5)$$

Przykład grafu dwudzielnego przedstawiono na rys. 2.3. Węzły różnych typów oznaczono odmiennymi kolorami.



Rys. 2.3. Graf dwudzielny

Wszystkie pojęcia przedstawione w tym podrozdziale można zdefiniować analogicznie dla grafów bez łuków wielokrotnych.

2.2. Sieci i ich reprezentacja graficzna

Jak już wspomniałem na początku do tego rozdziału, sieci Petriego są przedstawiane jako grafy dwudzielne. W większości klas sieci omawianych w tej książce nie mogą występować łuki wielokrotne i definicje takich klas sieci są oparte na grafach skierowanych, przedstawionych jako pary $\mathcal{G} = (V, A)$. Grafy z łukami wielokrotnymi służą do opisu grafów osiągalności (zob. rozdz. 5) oraz sieci kolorowanych (zob. rozdz. 10).

Definicja 2.7. Uporządkowaną trójkę postaci $\mathcal{N} = (P, T, A)$ nazywamy *siecią*, jeżeli są spełnione warunki:

- (1) P jest niepustym zbiorem *miejsc*.
- (2) T jest niepustym zbiorem *przejścia* (*tranzycji*), takim że $P \cap T = \emptyset$.
- (3) $A \subseteq (P \times T) \cup (T \times P)$ jest zbiorem *łuków sieci*.

Dla sieci niskiego poziomu zbiór A nazywany jest również *relacją przepływu*. ■

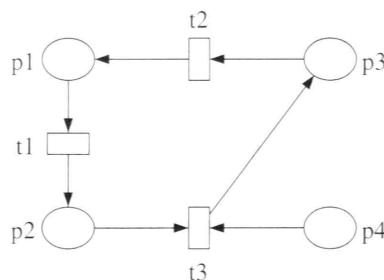
Będziemy odtąd rozważać tylko *sieci skończone*, tzn. takie, których zbiory miejsc i przejść są zbiorami skończonymi.

Dla sieci przyjmuje się następującą interpretację. Sieć przedstawia się jako graf dwudzielny $\mathcal{G} = (V, A)$, którego węzłami są elementy zbiorów P i T , tzn. $V = P \cup T$, łukami zaś, pary relacji A . Etykiety łuków nie mają bezpośredniego związku

z pojęciami omawianymi w tym rozdziale. Interpretacja etykiet łuków dotyczy sieci uogólnionych, którymi zajmiemy się w rozdz. 4.

Miejsca sieci są reprezentowane graficznie za pomocą elips, przejścia zaś, za pomocą prostokątów. Relacja przepływu jest reprezentowana za pomocą strzałek łączących odpowiednio elipsy z prostokątami, i prostokąty z elipsami. Na rysunku 2.4 przedstawiono przykład sieci $\mathcal{N} = (P, T, A)$, dla której:

- $P = \{p1, p2, p3, p4\}$;
- $T = \{t1, t2, t3\}$;
- $A = \{(p1, t1), (t1, p2), (p2, t3), (p4, t3), (t3, p3), (p3, t2), (t2, p1)\}$.



Rys. 2.4. Graficzna reprezentacja sieci

Wszystkie pojęcia zdefiniowane dla grafów skierowanych obowiązują również w odniesieniu do sieci, tzn. można mówić o drogach w grafie sieci (lub krótko – o drogach w sieci), cyklach, sieciach spójnych, sieciach silnie spójnych itd. Bezpośrednio z definicji sieci wynika, że poprzednikami i następcikami miejsc są przejścia i na odwrót.

Przy przyjętej definicji sieci, droga w odpowiadającym jej grafie jest jednoznacznie wyznaczona zarówno przez ciąg jej łuków, jak i ciąg węzłów. W celu uproszczenia opisu drogi w grafie sieci, zamiast ciągu łuków, np. $(p1, t1), (t1, p2), (p2, t3), (t3, p3)$, będzie występował ciąg węzłów: $p1, t1, p2, t3, p3$.

Dla sieci¹ przedstawionej na rys. 2.4 przykładem drogi nieskierowanej jest ciąg węzłów $p1, t1, p2, t3, p4$, przykładem drogi jest ciąg $p4, t3, p3, t2$, a przykładem cyklu, ciąg $p1, t1, p2, t3, p3, t2, p1$. Sieć zawiera jeden węzeł początkowy – miejsce $p4$.

¹ Mówiąc o graficznej reprezentacji sieci, będziemy równoważnie używać terminów: *sieć* i *graf sieci*.

2.3. Własności strukturalne sieci

W tym podrozdziale ograniczymy nasze rozważania do takich sieci, których graf jest grafem spójnym. W przypadku sieci, których grafy nie są spójne, ich poszczególne fragmenty można rozważać niezależnie.

Niech dana będzie sieć $\mathcal{N} = (P, T, A)$. Ze względu na charakter połączeń elementów sieci wyróżnia się kilka typów sieci.

Definicja 2.8. Parę węzłów x i y nazywamy *ciasną pętlą*, jeżeli węzeł y jest jednocześnie węzłem wejściowym i wyjściowym węzła x , tzn. jeżeli $(x, y) \in A$ i $(y, x) \in A$. Sieć \mathcal{N} nazywamy *czystą*, jeżeli nie zawiera ciasnych pętli. ■

Definicja 2.9. Sieć \mathcal{N} nazywamy *prostą*, jeżeli różne jej węzły mają różne zbiory wejściowe lub wyjściowe, tzn.:

$$\forall x, y \in V: x \neq y \Rightarrow (In(x) \neq In(y) \vee Out(x) \neq Out(y)). \quad (2.6)$$

Definicja 2.10. Sieć \mathcal{N} nazywamy *maszyną stanową*, jeżeli każde jej przejście ma dokładnie jedno miejsce wejściowe i dokładnie jedno miejsce wyjściowe, tzn., gdy jest spełniony warunek²:

$$\forall t \in T: |In(t)| = 1 \wedge |Out(t)| = 1. \quad (2.7)$$

Sieć \mathcal{N} nazywamy *grafem synchronizacji*, jeżeli każde jej miejsce ma dokładnie jedno przejście wejściowe i dokładnie jedno przejście wyjściowe, tzn., gdy jest spełniony warunek:

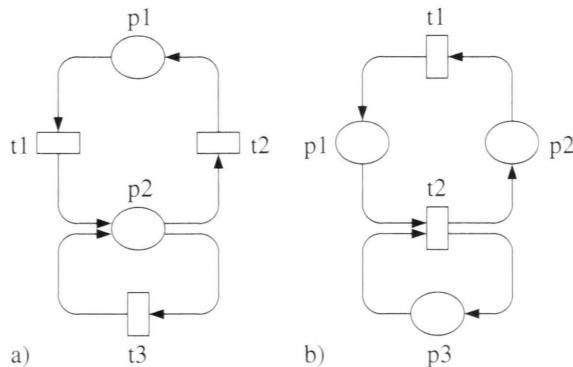
$$\forall p \in P: |In(p)| = 1 \wedge |Out(p)| = 1. \quad (2.8)$$

Jeśli dodatkowo graf sieci jest grafem silnie spójnym, to mówimy o *silnie spójnej maszynie stanowej* lub *silnie spójnym grafie synchronizacji*.

Maszyna stanowa i graf synchronizacji są pojęciami dualnymi. Graf dualny do maszyny stanowej \mathcal{N} (tzn. sieć, która powstaje z sieci \mathcal{N} , gdy miejsca zostaną

² Symbol $|X|$ oznacza moc zbioru X .

zamienione rolami z przejściami) jest zawsze grafem synchronizacji i na odwrót. Na rysunku 2.5 przedstawiono parę grafów dualnych. Jeżeli na przykład z sieci 2.5 b) usuniemy miejsce p_3 i powiązane z nim łuki, to otrzymana sieć będzie jednocześnie grafem synchronizacji i maszyną stanową.



Rys. 2.5. Maszyna stanowa a) i graf synchronizacji b)

Definicja 2.11. Sieć \mathcal{N} nazywamy *siecią swobodnego wyboru*, jeżeli dla dowolnych dwóch miejsc sieci \mathcal{N} , które mają wspólne przejście wyjściowe, przejście to jest ich jedynym przejściem wyjściowym, tzn., gdy jest spełniony warunek:

$$\begin{aligned} \forall t \in T \quad \forall p_i, p_j \in P: \\ (p_i \neq p_j \wedge t \in Out(p_i) \cap Out(p_j)) \Rightarrow \\ (Out(p_i) = Out(p_j) = \{t\}). \end{aligned} \quad (2.9)$$

Sieć \mathcal{N} nazywamy *rozszerzoną siecią swobodnego wyboru*, jeżeli dowolne dwa miejsca sieci \mathcal{N} , które mają wspólne przejście wyjściowe, mają równe zbiory przejść wyjściowych, tzn., gdy jest spełniony warunek:

$$\begin{aligned} \forall p_i, p_j \in P: (Out(p_i) \cap Out(p_j) \neq \emptyset) \Rightarrow \\ (Out(p_i) = Out(p_j)). \end{aligned} \quad (2.10)$$

Sieć \mathcal{N} nazywamy *rozszerzoną siecią prostą*, jeżeli dla dowolnych dwóch miejsc sieci \mathcal{N} , które mają wspólne przejście wyjściowe, zbiór przejść wyjściowych jednego z tych miejsc zawiera się w zbiorze przejść wyjściowych drugiego, tzn., gdy jest spełniony warunek:

$$\begin{aligned} \forall p_i, p_j \in P: (Out(p_i) \cap Out(p_j) \neq \emptyset) \Rightarrow \\ ((Out(p_i) \subseteq Out(p_j)) \vee (Out(p_j) \subseteq Out(p_i))). \end{aligned} \quad (2.11)$$

Między pojęciami przedstawionymi w definicji 2.11 zachodzi relacja porządku. Jeżeli sieć jest siecią swobodnego wyboru, to jest ona również rozszerzoną siecią swobodnego wyboru. Z kolei, jeżeli sieć jest rozszerzoną siecią swobodnego wyboru, to jest ona również rozszerzoną siecią prostą. Sieć przedstawiona na rys. 2.4 jest siecią swobodnego wyboru.

Definicja 2.12. Sieć $\mathcal{N}' = (P', T', A')$ nazywamy *podsiecią* sieci $\mathcal{N} = (P, T, A)$ jeżeli są spełnione warunki:

- (1) $P' \subseteq P$;
- (2) $T' \subseteq T$;
- (3) $A' = A \cap ((P' \times T') \cup (T' \times P'))$.

Ostatni warunek oznacza, że w sieci \mathcal{N}' występują wszystkie te łuki sieci \mathcal{N} , które łączą elementy zbiorów P' i T' . O podsieci \mathcal{N}' mówimy, że jest wyznaczona przez zbiory P' i T' .

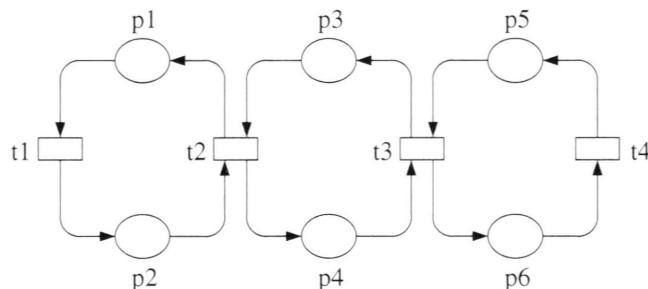
Jeżeli dodatkowo spełniony jest warunek:

$$T' = In(P') \cup Out(P'), \quad (2.12)$$

to mówimy, że podsieć jest *wyznaczona* przez zbiór miejsc P' , natomiast gdy:

$$P' = In(T') \cup Out(T'), \quad (2.13)$$

to mówimy, że podsieć jest *wyznaczona* przez zbiór przejść T' . ■



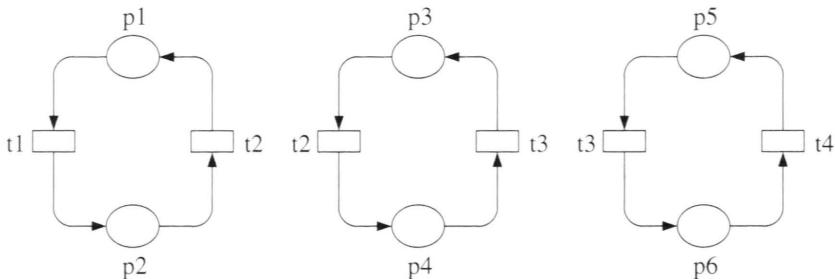
Rys. 2.6. Sieć rozkładalna na maszyny stanowe

Rozważmy sieć \mathcal{N} przedstawioną na rys. 2.6. Przykładem jej podsieci jest sieć $\mathcal{N}_1 = (P_1, T_1, A_1)$, dla której: $P_1 = \{p1, p2\}$, $T_1 = \{t1, t2\}$ oraz $A_1 = \{(p1, t1), (t1, p2), (p2, t2), (t2, p1)\}$. Jest to podsieć wyznaczona przez zbiór miejsc P_1 .

Definicja 2.13. Niech $\mathcal{N} = (P, T, A)$ będzie siecią i niech $\mathcal{N}' = (P', T', A')$ będzie podsiecią sieci \mathcal{N} .

- (1) Podsieć \mathcal{N}' nazywamy *składową* sieci \mathcal{N} , jeżeli jest ona podsiecią wyznaczoną przez zbiór P' .
- (2) Składową \mathcal{N}' nazywamy *MS-składową*, jeżeli \mathcal{N}' jest maszyną stanową, natomiast *SSMS-składową*, jeżeli \mathcal{N}' jest silnie spójną maszyną stanową. ■

Rozważana podsieć $\mathcal{N}_1 = (P_1, T_1, A_1)$ jest SSMS-składową sieci \mathcal{N} przedstawionej na rys. 2.6. Dla sieci \mathcal{N} istnieją dwie inne SSMS-składowe, wyznaczone przez zbiory miejsc $P_2 = \{p3, p4\}$ i $P_3 = \{p5, p6\}$ (zob. rys. 2.7). Inne podsieci, na przykład podsieć wyznaczona przez zbiór miejsc $P_4 = \{p1, p2, p3, p4\}$, są składowymi sieci \mathcal{N} , ale nie są SSMS-składowymi, gdyż nie są one maszynami stanowymi.



Rys. 2.7. Rozkład sieci na maszyny stanowe

Definicja 2.14. Sieć $\mathcal{N} = (P, T, A)$ jest *pokryta przez podsieci* $\mathcal{N}_i = (P_i, T_i, A_i)$, $i = 1, 2, \dots, k$, jeżeli spełnione są warunki:

$$P = \bigcup_{i=1}^k P_i, \quad T = \bigcup_{i=1}^k T_i, \quad A = \bigcup_{i=1}^k A_i. \quad (2.14)$$

Sieć \mathcal{N} nazywamy *rozkładalną na maszyny stanowe*, jeżeli istnieją silnie spójne maszyny stanowe $\mathcal{N}_i = (P_i, T_i, A_i)$, $i = 1, 2, \dots, k$, które tworzą jej pokrycie. ■

Sieć na rysunku 2.6 jest rozkładalna na maszyny stanowe. Rozkład sieci \mathcal{N} możemy zobaczyć na rys. 2.7.

2.4. Podsumowanie

Omówiliśmy w tym rozdziale podstawowe pojęcia dotyczące graficznej reprezentacji sieci Petriego. Sieci te są przedstawiane jako grafy dwudzielne, można zatem w odniesieniu do nich stosować pojęcia charakterystyczne dla grafów skierowanych, a więc mówić o drogach w sieci, cyklach, sieciach spójnych, sieciach silnie spójnych itd. Jednocześnie ze względu na charakter połączeń między węzłami sieci definiuje się własności strukturalne typowe dla sieci Petriego, na przykład: sieci czyste, sieci proste, sieci swobodnego wyboru, grafy synchronizacji, maszyny stanowe itp. Zarówno przedstawiona reprezentacja graficzna, jak i własności strukturalne są wspólne dla wszystkich klas sieci.

Sieci znakowane

Wprowadzona w poprzednim rozdziale definicja pozwala scharakteryzować strukturę sieci, ale nie daje możliwości stworzenia opisu jej zachowania. Aby można było w pełni modelować systemy współbieżne, musimy rozszerzyć tę definicję o znakowanie, które reprezentuje stan modelowanego systemu i ulega zmianie w trakcie symulacji pracy sieci. Znakowanie jest informacją o liczbie znaczników (żetonów) znajdujących się w poszczególnych miejscach. Zmiana znakowania następuje w wyniku wykonywania przejść, które powoduje zarówno usunięcie pewnej liczby znaczników znajdujących się w miejscach sieci, jak i dodanie nowych.

W tym rozdziale zajmiemy się jedną z najprostszych klas sieci, tzw. sieciami znakowanymi (zob. prace: [48], [56] i [70]), które posługują nam do omówienia własności dynamicznych. W zależności od tego, czy rozważa się zbiór osiągalnych znakowań czy też częstotliwość wykonywania poszczególnych przejść, wyróżnia się własności związane z ograniczonością miejsc i żywotnością przejść sieci. Omówimy ponadto podstawowe zasady konstruowania modeli w postaci sieci Petriego, a także kilka przykładów modelowania z zastosowaniem sieci znakowanych. W dalszym ciągu przyimionik *znakowana* często będzie pomijany, lecz nie powinno to prowadzić do nieporozumień.

3.1. Dynamika sieci

Stan sieci jest reprezentowany przez tzw. *znakowanie*, które zmienia się w wyniku *wykonywania przejść*. W porównaniu z definicją 2.7, w sieciach znakowanych wprowadza się dodatkową funkcję, która definiuje znakowanie początkowe.

Definicja 3.1. Uporządkowaną czwórkę postaci $\mathcal{N} = (P, T, A, M_0)$ nazywamy *siecią znakowaną*, jeżeli są spełnione warunki:

$$(1) \quad \mathcal{N} = (P, T, A) \text{ jest siecią.}$$

- (2) $M_0: P \rightarrow \mathbb{Z}_+$ jest funkcją określona na zbiorze miejsc, zwaną *znakowaniem początkowym* sieci \mathcal{N} . ■

Znakowanie początkowe jest funkcją, która każdemu miejscu sieci przyporządkowuje liczbę całkowitą nieujemną, interpretowaną jako liczba znaczników (żetonów) umieszczonych wstępnie w tym miejscu. Znaczniki są reprezentowane graficznie za pomocą kropek umieszczanych wewnątrz elips symbolizujących miejsca lub, gdy liczba znaczników jest duża, za pomocą etykiet zawierających informację o liczbie znaczników.

Definicja 3.2. Znakowaniem sieci $\mathcal{N} = (P, T, A, M_0)$ nazywamy dowolną funkcję

$$M: P \rightarrow \mathbb{Z}_+. \quad (3.1) \quad \blacksquare$$

Znakowanie sieci ulega zmianie w wyniku tzw. *wykonywania przejścia (odpalania przejścia)*. Wykonanie przejścia powoduje usunięcie znacznika z każdego z jego miejsc wejściowych i dodanie nowego znacznika do każdego z jego miejsc wyjściowych. Przejście można wykonać, jeżeli jest ono *aktywne*.

Definicja 3.3. Przejście $t \in T$ jest *aktywne* przy znakowaniu M (jest M -aktywne), jeżeli każde z jego miejsc wejściowych zawiera co najmniej jeden znacznik. ■

Jeżeli przejście t jest aktywne przy znakowaniu M , to nowe znakowanie M' uzyskanie w wyniku wykonania przejścia t jest określone w sposób następujący:

$$M'(p) = \begin{cases} M(p) - 1, & \text{gdy } p \in In(t) - Out(t), \\ M(p) + 1, & \text{gdy } p \in Out(t) - In(t), \\ M(p) & \text{w pozostałych przypadkach.} \end{cases} \quad (3.2)$$

Jeżeli przejście t jest aktywne przy znakowaniu M i w wyniku jego wykonania otrzymujemy znakowanie M' , to fakt ten zapisujemy następująco:

$$M \xrightarrow{t} M'. \quad (3.3)$$

Jeżeli istnieje przejście aktywne przy znakowaniu M' , to możemy je wykonać, otrzymując kolejne znakowanie itd. Niech dane będą przejścia $t_1, t_2, \dots, t_k \in T$ i znakowania M, M_1, M_2, \dots, M_k sieci \mathcal{N} , takie że:

$$M \xrightarrow{t_1} M_1 \xrightarrow{t_2} M_2 \xrightarrow{t_3} \cdots \xrightarrow{t_k} M_k. \quad (3.4)$$

O ciągu przejść $\alpha = t_1, t_2, \dots, t_k$ mówimy, że prowadzi od znakowania M do znakowania M_k ; fakt ten zapisujemy jako:

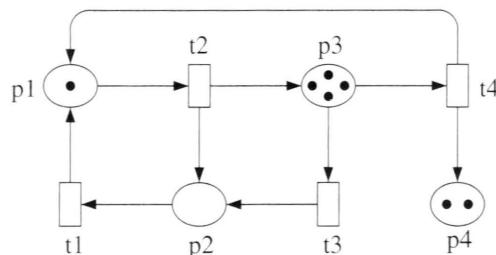
$$M \xrightarrow{\alpha} M_k. \quad (3.5)$$

Dowolne znakowanie sieci N , które możemy otrzymać ze znakowania M w wyniku wykonania skończonego ciągu przejść, nazywamy *znakowaniem osiągalnym* ze znakowania M . Zbiór wszystkich znakowań osiągalnych ze znakowania M oznaczamy symbolem $\mathcal{R}(M)$. Zbiór znakowań osiągalnych ze znakowania początkowego oznaczamy symbolem $\mathcal{R}(M_0)$.

Z każdym znakowaniem M' osiągalnym ze znakowania M jest związany ciąg przejść, których wykonanie prowadzi od znakowania M do M' . Zbiór wszystkich ciągów przejść, które można wykonać, rozpoczynając od znakowania M , będziemy oznaczać symbolem $\mathcal{L}(M)$. Zbiór $\mathcal{L}(M)$ zawiera zarówno skończone, jak i nieskończone ciągi przejść.

Jeżeli przyjmiemy, że zbiór miejsc sieci N jest zbiorem uporządkowanym, $P = \{p_1, p_2, \dots, p_n\}$, to dowolne znakowanie można przedstawić jako wektor:

$$M = (M(p_1), M(p_2), \dots, M(p_n)). \quad (3.6)$$



Rys. 3.1. Sieć znakowana

Na rysunku 3.1 przedstawiono przykład sieci znakowanej N o znakowaniu początkowym $M_0 = (1, 0, 4, 2)$. Przy znakowaniu początkowym aktywne są przejścia t_2 , t_3 i t_4 . Wykonanie tych przejść prowadzi odpowiednio do następujących znakowań: $M_2 = (0, 1, 5, 2)$, $M_3 = (1, 1, 3, 2)$ i $M_4 = (2, 0, 3, 3)$.

3.2. Ograniczoność sieci

Pojęcie ograniczoności sieci wiąże się z pojęciem bezpieczeństwa spotykanym w opisach systemów współbieżnych. Definicja bezpieczeństwa przyjmowania w teorii systemów współbieżnych oznacza niemożliwość wystąpienia niepożądanych stanów podczas działania systemu. Można to nieformalnie wyrazić w ten sposób, że *nic złego nigdy się nie zdarzy* (zob. pracę [41]). W przypadku sieci Petriego niepożądane stany dotyczą najczęściej nieograniczonego wzrostu liczby znaczników w pewnych miejscach sieci.

Podczas pracy sieci może się zmienić znakowanie, a tym samym liczba znaczników w poszczególnych jej miejscach. W zależności od charakteru tych zmian definiuje się kilka przedstawionych poniżej własności sieci:

Niech dana będzie sieć znakowana $\mathcal{N} = (P, T, A, M_0)$.

Definicja 3.4. Miejsce $p \in P$ nazywamy *znakowanym*, jeżeli istnieje znakowanie $M \in \mathcal{R}(M_0)$, takie że $M(p) > 0$.

Zbiór miejsc $P' \subseteq P$ nazywamy *czystym*, jeżeli przy dowolnym znakowaniu $M \in \mathcal{R}(M_0)$ żaden znacznik nie leży w miejscu należącym do tego zbioru. W przeciwnym przypadku zbiór P' nazywamy *znakowanym*. ■

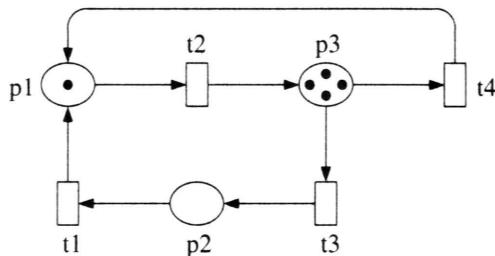
Definicja 3.5. Miejsce p nazywamy *ograniczonym*, jeżeli jest spełniony warunek:

$$\exists k \in \mathbb{N} \quad \forall M \in \mathcal{R}(M_0): M(p) \leq k, \quad (3.7)$$

tzn. przy dowolnym znakowaniu osiągalnym ze znakowania początkowego liczba znaczników w miejscu p jest nie większa niż k . Miejsce takie nazywamy również *k-ograniczonym*, a liczbę k , *górним ograniczeniem* miejsca p . Miejsce 1-ograniczone nazywamy miejscem *bezpiecznym*. Miejsce p nazywamy *nieograniczonym*, jeżeli nie jest ograniczone. Sieć \mathcal{N} nazywamy *k-ograniczoną*, jeżeli wszystkie jej miejsca są *k-ograniczonymi*; sieć 1-ograniczoną nazywamy *bezpieczną*. ■

Wszystkie miejsca sieci przedstawionej na rys. 3.1 są nieograniczone. Na rysunku 3.2 przedstawiono pewną modyfikację tej sieci. Zmodyfikowana sieć przy znakowaniu początkowym $M_0 = (1, 0, 4)$ jest siecią 5-ograniczoną, a przy znakowaniu $M_0 = (1, 0, 0)$, siecią bezpieczną.

Twierdzenie 3.1. Każda maszyna stanowa \mathcal{N} znakowana k znacznikami jest *k-ograniczona*. ■



Rys. 3.2. Ograniczona sieć znakowana

Dowód. Powyższe twierdzenie wynika z definicji 2.10. Wykonanie dowolnego przejścia w sieci będącej maszyną stanową powoduje jedynie *przesunięcie* znacznika z jego miejsca wejściowego do wyjściowego. Całkowita liczba znaczników nie ulega zmianie, mogą one co najwyżej zgromadzić się w jednym miejscu. \square

Wniosek 3.1. Jeżeli sieć znakowana N jest rozkładalna na maszyny stanowe, to jest ograniczona. \blacksquare

Definicja 3.6. Sieć N nazywamy *zachowawczą*, jeżeli jest spełniony warunek:

$$\forall M \in \mathcal{R}(M_0): \sum_{p \in P} M(p) = \sum_{p \in P} M_0(p), \quad (3.8)$$

tzn. jeżeli łączna liczba znaczników w sieci pozostaje stała dla każdego znakowania osiągalnego ze znakowania początkowego.

Sieć N nazywamy *zachowawczą względem wektora wag* (w_1, w_2, \dots, w_n) , przy czym $n = |P|$ oraz $w_i > 0$, dla $i = 1, 2, \dots, n$, jeżeli spełniony jest warunek:

$$\forall M \in \mathcal{R}(M_0): \sum_{i=1}^n w_i M(p_i) = \sum_{i=1}^n w_i M_0(p_i), \quad (3.9)$$

tzn. jeżeli łączna ważona liczba znaczników w sieci pozostaje stała dla każdego znakowania osiągalnego ze znakowania początkowego. \blacksquare

Pojęcie sieci zachowawczej jest mocniejsze, niż pojęcie sieci ograniczonej. Żądamy tutaj, by nie tylko nie następował nieograniczony wzrost liczby znaczników w miejscach sieci, ale by łączna ich liczba (ewentualnie ważona) była stała. Jeżeli wektor wag można określić tylko dla pewnego podzbioru $P' \subset P$, to mówimy, że sieć N jest *częściowo zachowawcza*.

Bezpośrednio z definicji maszyny stanowej wynika poniższy wniosek.

Wniosek 3.2. Jeżeli sieć \mathcal{N} jest maszyną stanową, to jest ona siecią zachowawczą. ■

Definicja 3.7. Znakowanie początkowe M_0 nazywamy *odtwarzalnym*, jeżeli istnieje różne od M_0 znakowanie $M \in \mathcal{R}(M_0)$, z którego znakowanie początkowe jest ponownie osiągalne. Sieć \mathcal{N} nazywamy *odtwarzalną*, jeżeli znakowanie początkowe jest odtwarzalne. Sieć \mathcal{N} nazywamy *odwracalną*, jeżeli znakowanie początkowe jest osiągalne z każdego znakowania $M \in \mathcal{R}(M_0)$.

Znakowanie M sieci \mathcal{N} nazywamy *znakowaniem własnym*, jeżeli jest osiągalne z dowolnego znakowania $M' \in \mathcal{R}(M_0)$. ■

Twierdzenie 3.2. Sieć \mathcal{N} jest odwracalna wtedy i tylko wtedy, gdy wszystkie znakowania $M \in \mathcal{R}(M_0)$ są jej znakowaniami własnymi. ■

Dowód. Niech sieć \mathcal{N} będzie siecią odwracalną i niech $M_1, M_2 \in \mathcal{R}(M_0)$. Istnieją więc skończone ciągi przejść α_1, α_2 , takie że

$$M_1 \xrightarrow{\alpha_1} M_0 \wedge M_0 \xrightarrow{\alpha_2} M_2. \quad (3.10)$$

Ciąg przejść $\alpha_1\alpha_2$ prowadzi od znakowania M_1 do M_2 . Wynika stąd, że dla dowolnych znakowań sieci $M_1, M_2 \in \mathcal{R}(M_0)$ znakowanie $M_2 \in \mathcal{R}(M_1)$, a zatem wszystkie znakowania sieci są jej znakowaniami własnymi. Dowód w drugą stronę wynika bezpośrednio z definicji znakowania własnego sieci. □

Sieć przedstawiona na rys. 3.1 nie jest siecią odtwarzalną, natomiast sieć przedstawiona na rys. 3.2 jest siecią odwracalną.

3.3. Żywotność sieci

Pojęcie żywotności można nieformalnie opisać następująco: *coś dobrego kiedyś się zdarzy*. Żywotność jest kojarzona z ciągłością działania. Przedstawione w poprzednim podrozdziale zagadnienia dotyczyły własności miejsc sieci zależnych od jej dynamiki. Zmiany znakowań sieci następują w wyniku wykonywania przejść. Zaczynając od znakowania początkowego, wykonywane mogą być wszystkie przejścia lub tylko część z nich. Istnieje również możliwość osiągnięcia znakowania, przy którym nie będzie aktywne żadne przejście.

W zależności od tego, które przejścia i jak często są wykonywane, definiuje się kilka własności związanych z żywotnością sieci. Sieć nazywamy żywą, jeżeli dla

każdego znakowania osiągalnego ze znakowania początkowego, wychodząc od tego znakowania, można wykonać każde przejście sieci. Oznacza to całkowity brak możliwości zablokowania pracy w jakiekolwiek części sieci. Jest to *najsielszta* własność żywotności, jaką można zdefiniować dla sieci. Zazwyczaj wystarczy, że są spełnione znacznie słabsze warunki.

Definicja 3.8. Niech t będzie przejściem sieci $\mathcal{N} = (P, T, A, M_0)$.

- (0) Przejście t nazywamy *martwym* ($\mathcal{L}0$ -żywym), jeżeli przejście t nie występuje w żadnym ciągu przejść należącym do zbioru $\mathcal{L}(M_0)$.
- (1) Przejście t nazywamy *potencjalnie wykonalnym* ($\mathcal{L}1$ -żywym), jeżeli istnieje ciąg przejść należący do zbioru $\mathcal{L}(M_0)$, w którym przejście t występuje co najmniej jeden raz.
- (2) Przejście t nazywamy $\mathcal{L}2$ -żywym, jeżeli dla dowolnej liczby naturalnej k istnieje ciąg przejść należący do zbioru $\mathcal{L}(M_0)$, w którym przejście t występuje co najmniej k razy.
- (3) Przejście t nazywamy $\mathcal{L}3$ -żywym, jeżeli istnieje ciąg przejść należący do zbioru $\mathcal{L}(M_0)$, w którym przejście t występuje nieskończonie wiele razy.
- (4) Przejście t nazywamy żywym ($\mathcal{L}4$ -żywym), jeżeli przejście t jest potencjalnie wykonalne dla każdego znakowania $M \in \mathcal{R}(M_0)$.

Sieć \mathcal{N} nazywamy $\mathcal{L}k$ -żywą, jeżeli każde przejście tej sieci jest $\mathcal{L}k$ -żywe dla $k = 0, 1, 2, 3, 4$. Sieć \mathcal{N} nazywamy dokładnie $\mathcal{L}k$ -żywą, jeżeli jest ona $\mathcal{L}k$ -żywa, ale nie jest $\mathcal{L}(k+1)$ -żywa. Sieć \mathcal{N} nazywamy strukturalnie żywą, jeżeli istnieje znakowanie początkowe M_0 , przy którym sieć ta jest żywą. ■

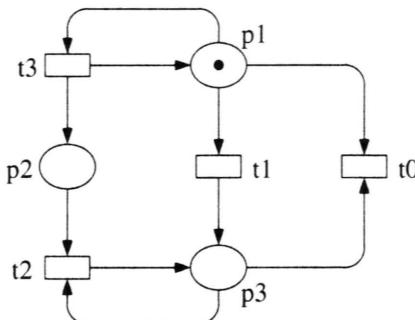
Definicje związane z żywotnością sieci odnoszą się również do jej znakowań. Znakowanie M nazywamy żywym, jeżeli sieć $\mathcal{N} = (P, T, A, M)$ jest żywa. Analogicznie znakowanie M nazywamy martwym, jeżeli sieć $\mathcal{N} = (P, T, A, M)$ jest martwa.

Prawdziwe są poniższe zależności:

$$\mathcal{L}4 \Rightarrow \mathcal{L}3 \Rightarrow \mathcal{L}2 \Rightarrow \mathcal{L}1, \quad (3.11)$$

to znaczy, że jeżeli przejście (sieć) jest $\mathcal{L}4$ -żywe, to jest również $\mathcal{L}3$ -żywe itd.

Rozważmy przykład sieci znakowanej przedstawionej na rys. 3.3. Przejścia $t0$, $t1$, $t2$ i $t3$ są odpowiednio dokładnie $\mathcal{L}0$, $\mathcal{L}1$, $\mathcal{L}2$, i $\mathcal{L}3$ -żywe. Rozpoczynając od znakowania początkowego, możemy dowolną liczbę razy wykonać przejście $t3$, dopóki nie wykonamy przejścia $t1$. Możliwe jest jednokrotne wykonanie przejścia $t1$, po czym zostanie wykonane przejście $t2$ tyle razy, ile znaczników zgromadziło



Rys. 3.3. Przejścia o różnym stopniu żywotności

się wcześniej w miejscu p_2 . W wyniku tego sieć osiągnie martwe znakowanie $M = (0, 0, 1)$. Przejście t_0 nigdy nie będzie aktywne. Jeżeli będziemy wykonywać tylko przejście t_3 , to otrzymamy nieskończony ciąg wykonania tego przejścia.

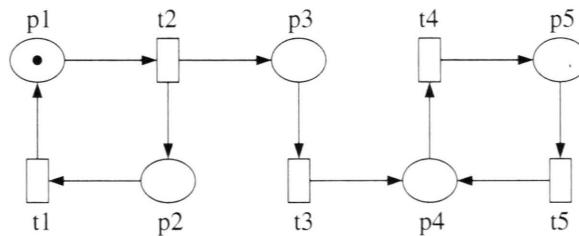
Definicja 3.9. Miejsce $p \in P$ nazywamy żywym, jeżeli dla dowolnego znakowania $M \in \mathcal{R}(M_0)$ istnieje znakowanie $M' \in \mathcal{R}(M)$, takie że $M'(p) > 0$. Sieć N nazywamy żywą ze względu na miejsca, jeżeli wszystkie jej miejsca są żywe. ■

Innymi słowy, żywotność miejsca oznacza, że zawsze ma ono szansę ponownie zawierać znaczniki, natomiast żywotność przejścia oznacza, że zawsze ma ono szansę ponownie być aktywne. Kiedy przejście nie jest przejściem żywym, wtedy jeszcze nie oznacza to, że jest ono martwe. Może się zdarzyć, że przy zadanym znakowaniu początkowym przejście będzie mogło być wykonane kilka razy, a dopiero później straci możliwość bycia aktywnym. Martwe przejścia można usunąć z sieci, nie ryzykując zmiany jej zachowania.

Podobnie jak w przypadku przejść, jeżeli sieć nie jest żywa, to nie znaczy, że jest ona martwa. Jest to związane z faktem, że poszczególne jej przejścia mogą mieć różne poziomy żywotności.

Rozważmy sieć pokazaną na rys. 3.4. Przy przedstawionym znakowaniu początkowym sieć jest żywa i jest żywa ze względu na miejsca. Ponadto, miejsca p_1 i p_2 są bezpieczne, a pozostałe miejsca są nieograniczone.

Przy znakowaniu początkowym $M_0 = (0, 0, 3, 0, 1)$ rozważana sieć nie jest ani żywa, ani martwa. Żywe są przejścia t_4 i t_5 , martwe zaś, t_1 i t_2 . Przejście t_3 nie jest przejściem martwym, bo można je trzykrotnie wykonać (jest $\mathcal{L}1$ -żywe). Zbiór miejsc $P' = \{p_1, p_2\}$ jest zbiorem czystym, miejsce p_3 jest 3-ograniczone, a miejsca p_4 i p_5 są 4-ograniczone. Tylko miejsca p_4 i p_5 są żywe.



Rys. 3.4. Sieć żywa i żywa ze względu na miejsca

Twierdzenie 3.3. Jeżeli sieć znakowana N jest silnie spójną maszyną stanową, której zbiór miejsc jest znakowany, to jest żywa. ■

Dowód. Z twierdzenia 3.1 wynika, że liczba znaczników w maszynie stanowej nie ulega zmianie. Ponieważ graf sieci jest silnie spójny, to dla dowolnej pary węzłów istnieje droga od jednego z nich do drugiego. Każde przejście w dowolnym znakowaniu sieci znajduje się więc w znakowanym cyklu, a zatem jest potencjalnie wykonalne. □

Jeżeli przy pewnym znakowaniu M aktywne są dwa przejścia, ale wykonanie dowolnego z nich powoduje utratę aktywności przez drugie, to mówimy, że przejścia te pozostają ze sobą w *konflikcie dynamicznym* (lub krótko – w *konflikcie*). Przejścia pozostają natomiast w *konflikcie statycznym*, jeżeli po umieszczeniu po jednym znaczniku w każdym z miejsc sieci, przejścia te będą w konflikcie dynamicznym. Jeżeli w sieci nie może dojść do konfliktu dynamicznego, to sieć nazywamy *trwałą*.

Rozważmy ponownie sieć przedstawioną na rys. 3.2. Jeżeli znakowanie miejsca p_3 będzie równe jeden, to między przejściami t_3 i t_4 wystąpi konflikt dynamiczny. Przejścia te pozostają również w konflikcie statycznym.

Klasifikacja żywotności przedstawiona w definicji 3.8 nie odzwierciedla dobrze regularności działania sieci, i to nawet w odniesieniu do sieci żywych. Szansa na ponowne wykonanie przejścia nie oznacza, że przejście to kiedykolwiek będzie wykonane. Może się zdarzyć, że ze względu na występujące konflikty przejście jest nieskończonym wiele razy aktywne, ale nigdy nie zostaje wykonane. Dla sieci wprowadza się pojęcie *sprawiedliwości*, które oznacza żądanie równomiernego wykonywania poszczególnych jej fragmentów. To żądanie nakłada silne wymagania na strukturę sieci, która nie powinna dopuszczać do *zapętlania* jej fragmentów.

Definicja 3.10. Ciąg przejść $\alpha \in \mathcal{L}(M_0)$ nazywamy *sprawiedliwym*, jeżeli jest skończony lub jest nieskończony i każde przejście $t \in T$ występuje nieskończonie wiele razy w ciągu α . Sieć N nazywamy *sprawiedliwą*, jeżeli każdy ciąg przejść $\alpha \in \mathcal{L}(M_0)$ jest sprawiedliwy. ■

Sieci przedstawione na rys. 3.1, 3.2, 3.3 i 3.4 nie są sieciami sprawiedliwymi. W każdym przypadku można podać nieskończony ciąg przejść, który nie jest sprawiedliwy, na przykład ciąg $\alpha = t_2, t_4, t_2, t_4 \dots$ dla sieci na rys. 3.1.

3.4. Charakterystyczne konstrukcje sieciowe

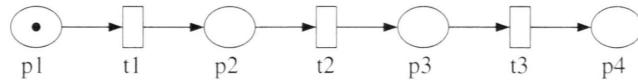
Łatwość i poglądowość opisu za pomocą sieci Petriego sprawiły, że znalazły one szerokie zastosowanie w modelowaniu (na różnych poziomach abstrakcji) systemów, w których występują procesy współbieżne. Zastosowania sieci wykraczają poza obszar informatyki i są spotykane w elektronice, automatyce, biologii, biochemii, medycynie, ochronie środowiska¹, przemyśle itp. Modelowanie z zastosowaniem sieci Petriego pozwala na przedstawienie aspektów systemu powiązanych z: współbieżnością działania, współdzieleniem zasobów, komunikacją między procedami, synchronizacją procesów itp.

Podrozdział ten stanowi wprowadzenie do zagadnień związanych z praktycznym modelowaniem systemów przy użyciu sieci Petriego. Omówimy w nim typowe konstrukcje spotykane w sieciach, które mają związek z modelowaniem charakterystycznych właściwości systemów współbieżnych.

Program sekwencyjny można określić jako ciąg wykonywalnych instrukcji zapisanych w jakimś języku programowania wraz z deklaracjami danych, które program ten ma przetwarzać (zob. pracę [3]). Instrukcje te są kolejno wykonywane przez komputer zgodnie z ich porządkiem. Poprawność takich programów można badać dwojako, wykonując odpowiednio przygotowane testy albo formalnie dowodząc poprawności zastosowanych algorytmów. Sposób modelowania sekwencyjności wykonywania przejść, które mogą być interpretowane jako instrukcje programu, przedstawiono na rys. 3.5.

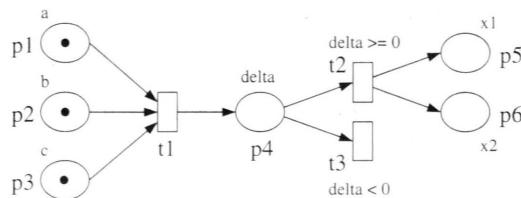
Na rysunku 3.6 przedstawiono sieć będącą reprezentacją algorytmu rozwiązywania równań kwadratowych w zbiorze liczb rzeczywistych. Obok miejsc i przejść zamieszczono komentarze, opisujące znaczenie poszczególnych elementów tej sieci.

¹ Obszerny przegląd literatury poświęconej różnorodnym zastosowaniom sieci Petriego można znaleźć w pracy [89].



Rys. 3.5. Modelowanie sekwencyjności wykonywania przejść

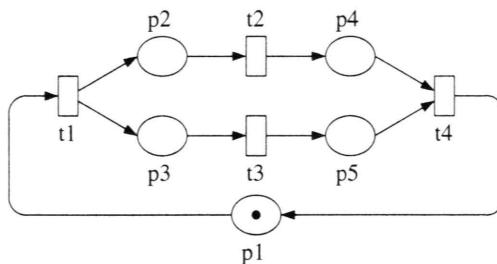
Znaczniki w miejscach p_1 , p_2 i p_3 reprezentują współczynniki równania kwadratowego. Jeżeli jego wyróżnik będzie miał wartość nieujemną, to w wyniku wykonania przejść t_1 i t_2 pojawią się w miejscach p_5 i p_6 znaczniki wskazujące na istnienie rozwiązań równania. Zapis ten jest zapisem uproszczonym, gdyż stosując sieci niskiego poziomu, nie można wyrazić wartości liczbowych (oprócz przypadków, w których liczbę znaczników w danym miejscu sieci interpretujemy jako liczbę całkowitą).



Rys. 3.6. Algorytm rozwiązywania równań kwadratowych

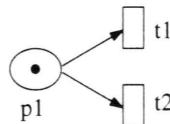
Program współbieżny można traktować jako zbiór programów sekwencyjnych (podczas wykonywania zwanych *procesami*), które są wykonywane *abstrakcyjnie równolegle*. Równoległość wykonywania dwóch procesów oznacza, że przedziały określające czas ich wykonania przynajmniej częściowo się nakładają (mogą się nałożyć), tzn. wykonanie jednego z nich rozpoczyna się przed zakończeniem drugiego. Abstrakcyjność równoległości oznacza, że program może być wykonywany przez maszynę jednoprocesorową, a moc obliczeniową procesora wspólnie dzielą poszczególne procesy.

Dowodzenie poprawności programów współbieżnych jest zadaniem o wiele bardziej skomplikowanym, niż dowodzenie poprawności programów sekwencyjnych. Wyniki działania takiego programu zależą nie tylko od danych wejściowych, ale również od sposobu *nałożenia się* fragmentów wykonywanych współbieżnie, a dokładniej mówiąc, od sposobu przeplatania się ciągów wykonania atomowych instrukcji procesów sekwencyjnych. Korzystając z sieci Petriego, można takie programy analizować w sposób formalny. Rysunek 3.7 może być traktowany jako model programu, w którym przejścia t_2 i t_3 mogą być wykonane współbieżnie. Przejście t_1 rozdziela obliczenia na dwa współbieżne procesy.

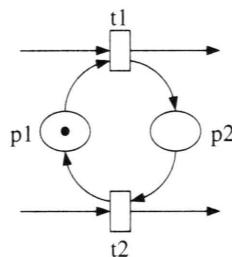


Rys. 3.7. Modelowanie współbieżności wykonywania przejść

Systemy modelowane z zastosowaniem sieci Petriego często działają w sposób niedeterministyczny. Dzieje się tak w sytuacjach konfliktowych. Dla przejść t_1 i t_2 na rys. 3.8 wybór przejścia, które ma być wykonane, jest wyborem niedeterministycznym. Wprowadzenie determinizmu w tego typu sytuacjach wymaga na przykład dodania priorytetów przejść (zob. rozdz. 7). Na rysunku 3.9 przedstawiono konstrukcję, która eliminuje niedeterminizm i wymusza naprzemienne wykonywanie przejść t_1 i t_2 .



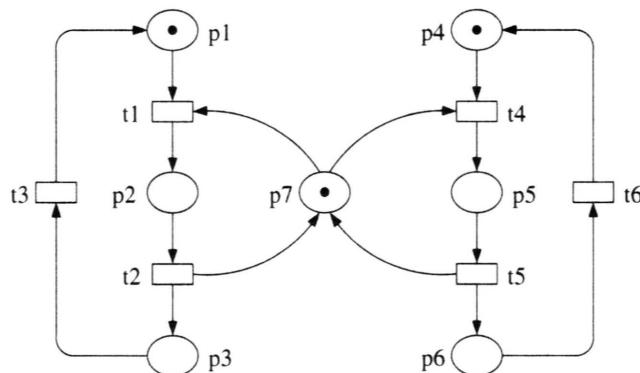
Rys. 3.8. Sytuacja konfliktowa



Rys. 3.9. Modelowanie naprzemienności wykonywania przejść

Program współbieżny składa się z co najmniej dwóch procesów wykonywanych równolegle. Procesy te mogą rywalizować o wspólne zasoby (dane, zasoby sprzętowe itp.). Zasoby takie określane są jako *zasoby współdzielone*. Ta część procesu,

w której korzysta się z zasobu współdzielonego, jest nazywana *sekcją krytyczną* procesu. Problem współzawodnictwa procesów o zasób, z którego w danej chwili może korzystać tylko jeden proces, nazywany jest *wzajemnym wykluczaniem*. Program współbieżny musi zapewnić możliwość wzajemnego wykluczania, tzn. instrukcje z sekcji krytycznych dwu lub większej liczby procesów nie mogą być przeplatane. W celu zapewnienia wzajemnego wykluczania stosuje się mechanizmy zwane *semaforami i monitorami*².



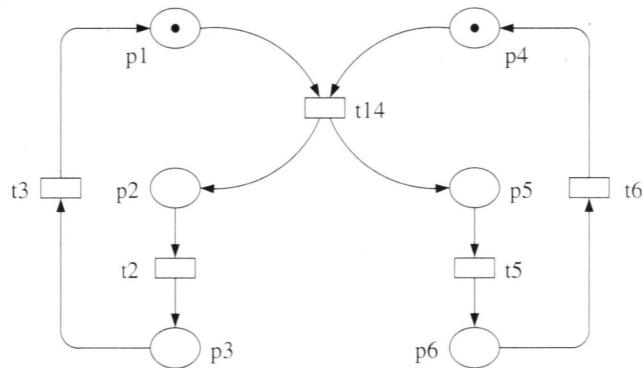
Rys. 3.10. Modelowanie wzajemnego wykluczania

Na rysunku 3.10 przedstawiono dwa współbieżne procesy reprezentowane przez podsieci wyznaczone odpowiednio przez zbiory miejsc $\{p_1, p_2, p_3\}$ i $\{p_4, p_5, p_6\}$. Przyjmijmy, że rywalizują one o wspólny zasób, a znaczniki w miejscach p_2 i p_5 oznaczają przebywanie odpowiednio pierwszego (lewego) i drugiego (prawego) procesu w sekcji krytycznej. Użycie dodatkowego miejsca p_7 , które pełni w tym modelu funkcję semafora, gwarantuje, że w sekcji krytycznej znajdzie się w danej chwili co najwyżej jeden proces.

Procesy współbieżnie mogą nie tylko rywalizować o zasoby współdzielone, ale również przesyłać sobie dane lub synchronizować działanie. *Komunikacja* między procesami występuje wtedy, kiedy dwa procesy, z których jeden odgrywa rolę *nadawcy*, a drugi *odbiorcy*, porozumiewają się w celu przesłania danych. Komunikację nazywamy *synchroniczną*, jeżeli nadawca żąda, by odbiorca był gotów do odebrania komunikatu (nadawanie i odbiór odbywają się jednocześnie). Przy komunikacji synchronicznej mamy do czynienia z *synchronizacją* procesów, tzn. uporządkowaniem ich w czasie. Jeżeli jeden z procesów jako pierwszy dotrze do punktu synchronizacji, to musi czekać na drugi proces.

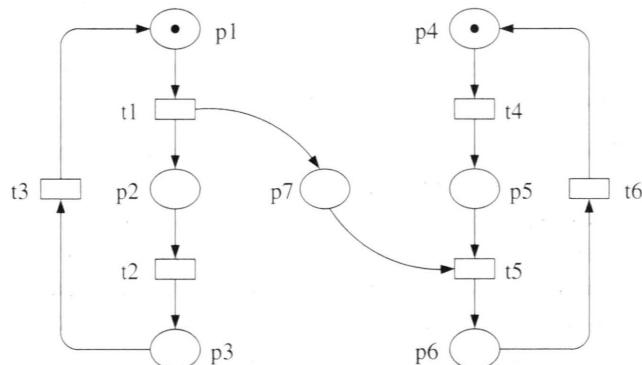
² Mechanizmy takie spotykamy na przykład w języku Ada 2005.

Rozważmy procesy przedstawione na rys. 3.11. Przejście t_{14} synchronizuje obydwa procesy. Kiedy jeden z procesów wcześniej dotrze do miejsca synchronizacji, wtedy musi zaczekać na drugi proces.



Rys. 3.11. Modelowanie komunikacji synchronicznej procesów

Drugim rodzajem komunikacji jest *komunikacja asynchroniczna*. Komunikację nazywamy asynchroniczną, jeżeli nadawca może wysłać wiadomość i kontynuować pracę bez względu na stan odbiorcy. Przy komunikacji asynchronicznej konieczne jest *buforowanie komunikatów*. Przykład komunikacji asynchronicznej przedstawiono na rys. 3.12. Miejsce p_7 pełni funkcję bufora do przekazywania informacji od pierwszego do drugiego procesu (bufor ten nie przechowuje informacji o kolejności, w jakiej umieszczono w nim komunikaty).



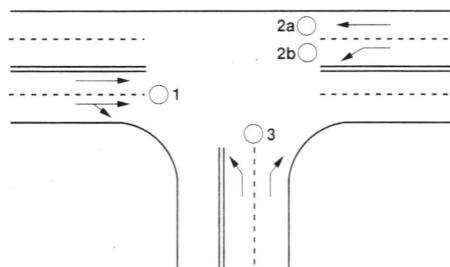
Rys. 3.12. Modelowanie komunikacji asynchronicznej procesów

3.5. Modelowanie z zastosowaniem sieci znakowych

W tym podrozdziale omówimy przykłady zastosowania sieci znakowych do modelowania systemów współbieżnych. Będą to dwie wersje modelu sygnalizacji świetlnej, cztery wersje sieci modelujących problem pięciu filozofów i model prostego protokołu komunikacyjnego.

3.5.1. Model sygnalizacji świetlnej

Zanalizujmy system sygnalizacji świetlnej na skrzyżowaniu przedstawionym na rys. 3.13. Cztery sygnalizatory umieszczone na skrzyżowaniu oznaczono symbolami 1, 2a, 2b i 3. Przyjmijmy dla uproszenia, że są wyświetlane tylko dwa sygnały: czerwony i zielony. Listę wszystkich dopuszczalnych stanów przedstawiono w tab. 3.1. Każdemu ze stanów będzie odpowiadać jedno znakowanie tworzonej sieci.



Rys. 3.13. Schemat skrzyżowania dróg

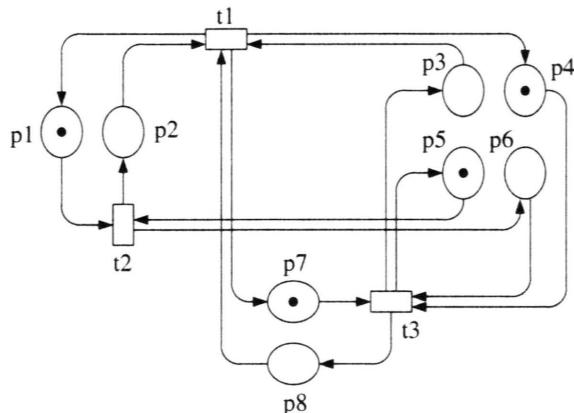
Tabela 3.1. Dopuszczalne stany systemu

Stan	1	2a	2b	3
1	zielony	zielony	czerwony	czerwony
2	czerwony	zielony	zielony	czerwony
3	czerwony	czerwony	czerwony	zielony

Rozwiązanie 1

Przyjmijmy, że każdy sygnalizator będzie reprezentowany przez dwa miejsca: jedno – reprezentujące sygnał zielony, drugie – sygnał czerwony. Obecność znacznika

w danym miejscu będzie oznaczać wyświetlanie reprezentowanego przez to miejsce sygnału (przyjmujemy, że patrząc od strony nadjeżdżającego pojazdu, dolne światło jest światłem zielonym, a górne czerwonym). Sieć znakowaną N_1 , modelującą omawiany system przedstawiono na rys. 3.14.



Rys. 3.14. Model sygnalizacji świetlnej (wersja 1)

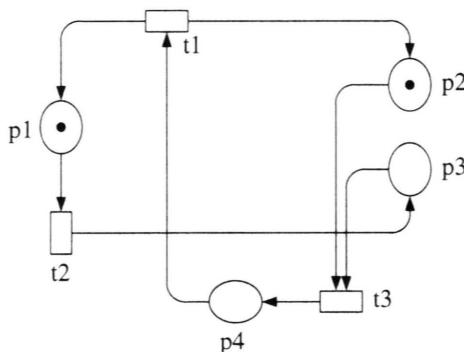
Rozważana sieć jest grafem synchronizacji i jest siecią żywą, bezpieczną, żywą ze względu na miejsca i odwracalną. Cyklicznie są osiągane trzy znakowania $M_0 = (1, 0, 0, 1, 1, 0, 1, 0)$, $M_1 = (0, 1, 0, 1, 0, 1, 1, 0)$ i $M_2 = (0, 1, 1, 0, 1, 0, 0, 1)$, które odpowiadają stanom systemu opisany w tab. 3.1. Pomijając skończone ciągi przejść, istnieje dokładnie jeden nieskończony ciąg przejść, w którym każde przejście sieci występuje nieskończenie wiele razy, a zatem sieć jest sprawiedliwa. Łatwo zauważać, że w trakcie pracy sieci nie ulega zmianie liczba znaczników, tzn. sieć jest zachowawcza. Przedstawione rozwiązańe można zoptymalizować pod względem liczby węzłów sieci.

Rozwiązanie 2

Ten sam system można zamodelować, stosując tylko jedno miejsce dla każdego sygnalizatora. Obecność znacznika w takim miejscu będzie oznaczać, że świeci się sygnał zielony, brak zaś znacznika – wyświetlanie sygnału czerwonego. Rozwiązanie takie przedstawiono na rys. 3.15.

W drugim rozwiązańiu (rys. 3.15) sieć również jest grafem synchronizacji, jest żywą, bezpieczną, żywą ze względu na miejsca i odwracalną. Analogicznie jak

w rozwiążaniu poprzednim są osiągane cyklicznie trzy stany: $M_0 = (1, 1, 0, 0)$, $M_1 = (0, 1, 1, 0)$ i $M_2 = (0, 0, 0, 1)$, a zatem sieć jest także sprawiedliwa.

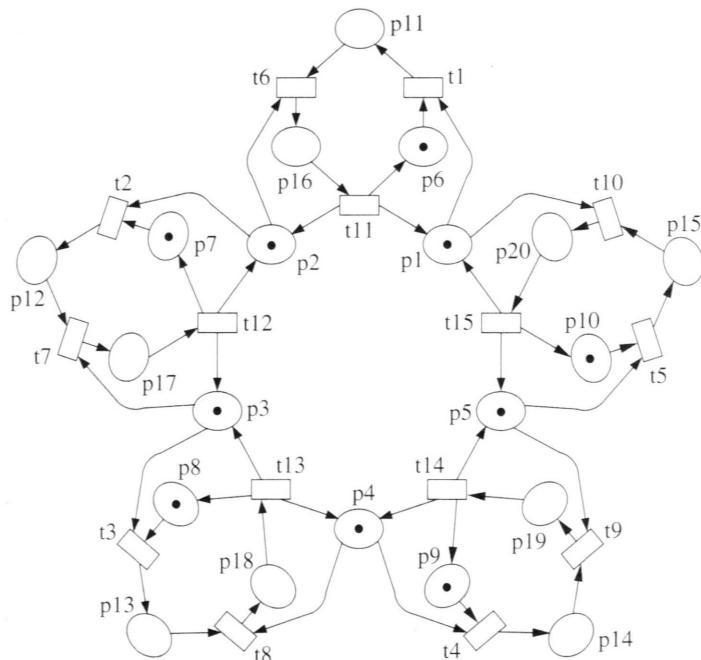


Rys. 3.15. Model sygnalizacji świetlnej (wersja 2)

W tym przypadku liczba znaczników w sieci nie jest stała. Sieć nie jest więc zachowawcza. Istnieje jednak możliwość wskazania wektora wag i pokazania, że sieć jest zachowawcza względem tego wektora. Wykonanie przejścia t_3 powoduje usunięcie dwóch znaczników (z miejsc p_2 i p_3) oraz dodanie tylko jednego znacznika do miejsca p_4 . W przypadku wykonania przejścia t_1 zachodzi sytuacja odwrotna – z sieci jest usuwany jeden znacznik, ale są dodawane dwa znaczniki. Dla miejsca p_4 należy więc przyjąć wagę dwa razy większą niż dla pozostałych miejsc. Przyjmując wektor wag $w = (1, 1, 1, 2)$, można pokazać, że sieć jest zachowawcza względem wektora w . Ważona łączna liczba znaczników wynosi wówczas 2 i jest stała dla wszystkich znakowań osiągalnych ze znakowania początkowego.

3.5.2. Problem pięciu filozofów

Jako kolejny przykład rozważymy znany z literatury problem pięciu filozofów. Filozofowie w odosobnionym gronie wykonują dwie czynności: *myślenie* i *jedzenie*. Posiłki są spożywane przy wspólnym stole z pięcioma talerzami i pięcioma widelcami. Na środku stołu stoi miska ze stale donoszonym spaghetti. Do jedzenia filozof potrzebuje dwóch widelców. Może on podnieść jedynie widelce znajdujące się po jego prawej i lewej stronie.



Rys. 3.16. Sieć modelująca problem pięciu filozofów

Rozwiązanie 1

W pierwszym rozwiążaniu przyjmijmy założenie, że filozof najpierw podnosi lewy widelec, a później prawy. Sieć modelującą opisany problem przedstawiono na rys. 3.16. Miejsca i przejścia sieci tak ponumerowano, że rozważając je modulo 5, otrzymujemy elementy odnoszące się do jednego filozofa. Znaczenie poszczególnych elementów sieci dotyczących pierwszego filozofa jest następujące:

- Miejsca p_1 i p_2 reprezentują odpowiednio lewy i prawy widelec, znacznik w danym miejscu oznacza, że widelec leży na stole.
- Miejsca p_6 , p_{11} i p_{16} reprezentują różne stany, w których może znaleźć się filozof. Znaczniki w tych miejscach oznaczają odpowiednio: filozof rozmyśla i jest gotów podnieść lewy widelec; filozof podniósł już lewy widelec i może podnieść prawy (o ile ten jest dostępny); filozof podniósł oba widelce i je, może on również odłożyć widelce na stół.

- Przejścia $t1$, $t6$ i $t11$ reprezentują odpowiednio czynności wykonywane przez filozofa: podniesienie lewego widelca, podniesienie prawego widelca i odłożenie obu widelców na stół.

Przedstawiona sieć jest siecią bezpieczną, ale żadne przejście nie jest żywe. Sieć N jest dokładnie Ł3-żywa. Dla każdego przejścia można wskazać nieskończony ciąg przejść, w którym jest ono wykonywane nieskończonymi wielokrotnie (np. dla przejścia $t1$ jest to ciąg $t1, t6, t11, t1, t6, t11 \dots$). Można również wskazać sprawiedliwy ciąg przejść (wszyscy filozofowie kolejno posilają się), jednak nie wszystkie ciągi przejść są sprawiedliwe, a zatem sieć nie jest sprawiedliwa.

Sieć nie jest siecią zachowawczą, ale jest siecią zachowawczą względem wektora wag. Przykładem takiego wektora wag jest wektor:

$$w = (1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3).$$

Łączna ważona liczba znaczników wynosi wówczas 10 i jest stała dla wszystkich znakowań osiągalnych ze znakowania początkowego.

Sieć nie jest żywa, ponieważ w zbiorze znakowań osiągalnych znajduje się znakowanie martwe:

$$M = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0).$$

Znakowanie to odpowiada sytuacji, w której wszyscy filozofowie jednocześnie podnoszą lewe widelce, powodując zablokowanie systemu. W kolejnych prezentowanych rozwiązańach podjęto próbę wyeliminowania tego błędu.

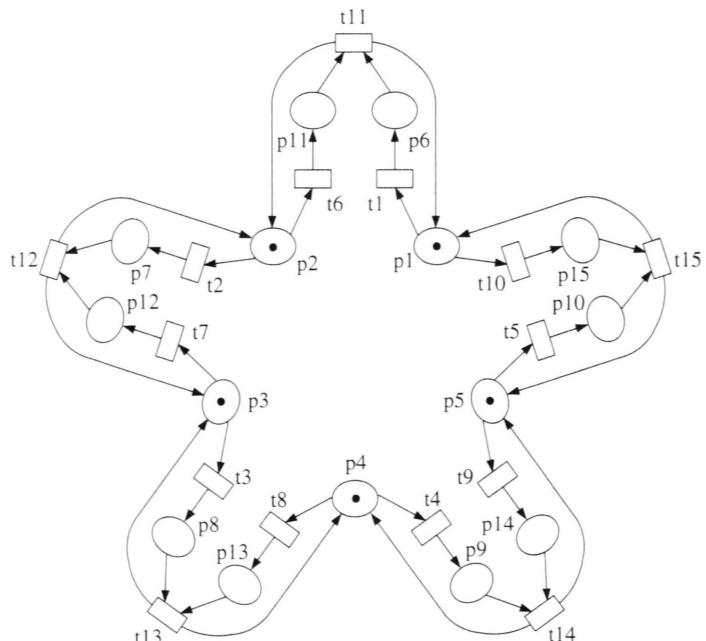
Rozwiązanie 2

Można podać kilka rozwiązań, które zapobiegają wystąpieniu blokady w systemie pięciu filozofów. Jedną z możliwości jest wymuszenie, by jeden (ustalony) filozof, najpierw podniósł widelec prawy, a później lewy. Jeżeli wybranym filozofem byłby filozof pierwszy, to aby osiągnąć wspomniany efekt, należałoby zamienić łuki $(t1, p11)$, $(p11, t6)$, $(t6, p16)$, $(p16, t11)$, $(t11, p6)$ i $(p6, t1)$ na łuki przeciwnie skierowane oraz przyjąć znakowanie początkowe $M_0(p6) = 0$ i $M_0(p16) = 1$.

Tak zmodyfikowana sieć N' nie osiąga stanu martwego. Sieć N' jest odwracalna, żywa, bezpieczna i żywa ze względu na miejsca. Podobnie jak w poprzednim rozwiązańiu – nie jest siecią sprawiedliwą.

Rozwiązanie 3

W poprzednich rozwiązańach przyjęto założenie dotyczące kolejności podnoszenia widelców przez filozofów. Na rysunku 3.17 przedstawiono rozwiązanie, w którym zrezygnowano z tego założenia. Filozof może podnieść widelce w dowolnej kolejności (możliwe jest również podniesienie dwóch widelców jednocześnie).



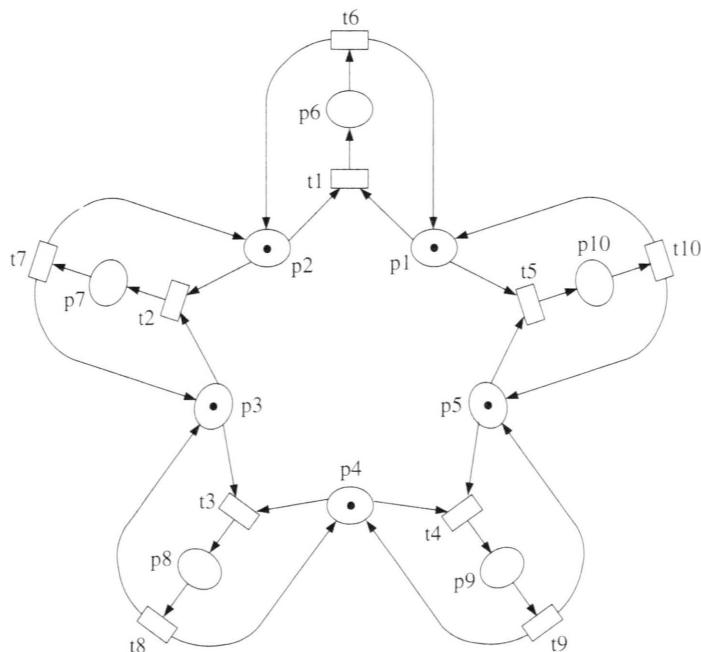
Rys. 3.17. Sieć modelująca problem pięciu filozofów (wersja 3)

Wszystkie znaczniki w sieci reprezentują widelce, przy czym znaczniki znajdujące się w miejscach od p_1 do p_5 reprezentują widelce znajdujące się na stole, a znaczniki w pozostałych miejscach sieci, widelce w posiadaniu filozofów. Znacznik w miejscu p_6 oznacza na przykład, że pierwszy filozof ma swój lewy widelec. Przejścia mają takie samo znaczenie, jakie miały w modelu przedstawionym na rys. 3.16.

Przedstawiona sieć jest bezpieczna, odtwarzalna i zachowawcza, a ponadto sieć jest dokładnie $\mathcal{L}3$ -żywa. Żadne z przejść nie jest przejściem żywym. Zbiór $\mathcal{R}(M_0)$ zawiera aż 32 martwe znakowania (odpowiadają one sytuacjom, gdy każdy filozof podniósł dokładnie jeden widelec). Sieć nie jest siecią sprawiedliwą, chociaż można wskazać sprawiedliwy ciąg przejść (wszyscy filozofowie kolejno się posilają).

Rozwiązanie 4

W poprzednim rozwiążaniu filozofowie mogli podnosić widełce w dowolnej kolejności. Jeżeli każdy z nich podniósł jeden widelec, to system się blokował. Można wyeliminować blokadę systemu, jeżeli przyjmiemy założenie, że każda osoba może sięgnąć po widełce tylko wówczas, gdy oba widełce (prawy i lewy) znajdują się na stole. W takiej sytuacji filozof może wziąć widełce, ale tylko oba jednocześnie. Rozwiązanie takie przedstawiono na rys. 3.18.



Rys. 3.18. Sieć modelująca problem pięciu filozofów (wersja 4)

Podobnie jak w poprzednich przykładach znaczniki znajdujące się w miejscach od $p1$ do $p5$ reprezentują widełce znajdujące się na stole. W przypadku pierwszego filozofa, znaczenie poszczególnych węzłów sieci jest następujące: przejście $t1$ reprezentuje podniesienie widełeców przez filozofa, a przejście $t6$ odłożenie widełeców na stół. Stan filozofa jest reprezentowany przez znakowanie miejsca $p6$. Znacznik w tym miejscu oznacza posilającego się filozofa, puste miejsce reprezentuje filozofa, który myśli. Analogiczne znaczenie mają węzły sieci reprezentujące pozostałych czterech filozofów.

Rozważana sieć jest odwracalna, żywa, żywa ze względu na miejsca i bezpieczna. Zbiór znakowań osiągalnych zawiera tylko 11 elementów. Sieć jest zachowawcza względem wektora wag $w = (1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2)$.

Sieć nie jest siecią sprawiedliwą. Brak sprawiedliwości pozostaje nadal wadą tego systemu. Pomimo że wyeliminowano możliwość zablokowania systemu, może się tak zdarzyć, iż dany filozof nigdy nie będzie się posiadać. Do rozważanego modelu można wprowadzić sprawiedliwość, dodając konstrukcję wymuszającą posilanie się filozofów w określonej kolejności. Rozwiążanie takie nie jest optymalne, gdyż z góry narzuca stałą częstotliwość posilania się filozofów (np. jednakową dla wszystkich).

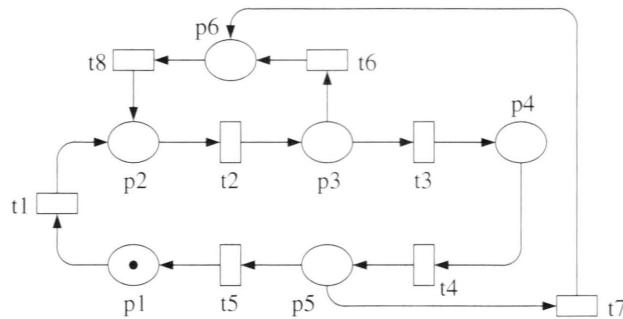
Można podać inne rozwiązania rozważanego problemu. Jedno z nich polega na dodaniu do sieci na rys. 3.17 przejść umożliwiających odłożenie podniesionego widełka (jeżeli drugi widelec nie jest dostępny), eliminując możliwość zablokowania systemu. Innym sposobem wyeliminowania blokady w rozważanym systemie jest wprowadzenie założenia, że filozofowie siadają do stołu tylko wtedy, gdy są głodni, przy czym jednocześnie co najwyżej czterech z nich może usiąść przy stole.

3.5.3. Model protokołu komunikacyjnego

Jako ostatni przykład rozważmy model prostego protokołu komunikacyjnego, który składa się z dwóch modułów: *nadawcy* i *odbiorcy*. *Nadawca* wysyła komunikaty i oczekuje na potwierdzenie odbioru. Po odebraniu potwierdzenia *nadawca* jest gotów wysłać kolejny komunikat. *Odbiorca* po odebraniu komunikatu wysyła potwierdzenie.

Przyjmijmy, że zarówno sam komunikat, jak i potwierdzenie odebrania, mogą zginąć podczas transmisji. Jeżeli *odbiorca* nie otrzyma potwierdzenia w określonym przedziale czasu, transmisja komunikatu będzie ponowiona. Sieci znakowane uniemożliwiają modelowanie czasu w sposób liczbowy. Wystąpienie przeterminowania będzie więc oznaczane przez umieszczenie znacznika w jednym z miejsc sieci. Niemożliwe jest również modelowanie prawdopodobieństwa, z jakim może dojść do przeterminowania. Model rozważanego systemu przedstawiono na rys. 3.19.

Przy dowolnym znakowaniu rozważana sieć zawiera dokładnie jeden znacznik (sieć jest zachowawcza). Zbiór $\mathcal{R}(M_0)$ jest zbiorem sześcioelementowym, a zatem modelowany system może znaleźć się w jednym z sześciu stanów, w zależności od tego, w którym miejscu znajduje się znacznik:



Rys. 3.19. Model protokołu komunikacyjnego

- p_1 – gotowość systemu do rozpoczęcia kolejnego cyklu pracy (poprawnie zakończył się poprzedni cykl),
- p_2 – gotowość nadawcy do wysłania wiadomości,
- p_3 – wiadomość w drodze od nadawcy do odbiorcy,
- p_4 – gotowość odbiorcy do wysłania potwierdzenia,
- p_5 – potwierdzenie w drodze od odbiorcy do nadawcy,
- p_6 – zagubienie wiadomości lub potwierdzenia (dojdzie do przeterminowania).

Przejścia sieci reprezentują następujące czynności:

- t_1 – wygenerowanie nowej wiadomości,
- t_2 – wysłanie wiadomości przez nadawcę,
- t_3 – odebranie wiadomości przez odbiorcę,
- t_4 – wysłanie potwierdzenia przez odbiorcę,
- t_5 – odebranie potwierdzenia przez nadawcę,
- t_6 – zagubienie wiadomości,
- t_7 – zagubienie potwierdzenia,
- t_8 – wysłanie sygnału o konieczności ponowienia transmisji.

Przedstawiona sieć jest maszyną stanową. Jest ona żywa, bezpieczna, żywa ze względu na miejsca i odwracalna, lecz nie jest sprawiedliwa. W zbiorze $\mathcal{L}(M_0)$ można wskazać sprawiedliwe ciągi przejść, ale nie wszystkie ciągi mają tę właściwość, na przykład nie jest sprawiedliwy ciąg przejść $\alpha = t_1, t_2, t_6, t_8, t_2, t_6, t_8 \dots$. W wypadku tego systemu nie należy traktować braku sprawiedliwości jako wady modelu.

3.6. Podsumowanie

W tym rozdziale omówiliśmy podstawowe pojęcia dotyczące dynamiki sieci Petriego. Stan modelowanego systemu jest reprezentowany przez znakowanie, które zmienia się w wyniku wykonywania przejść. Wykonanie dowolnego przejścia powoduje usunięcie znacznika z każdego z jego miejsc wejściowych i dodanie nowego znacznika do każdego z jego miejsc wyjściowych. Przejścia można wykonać, o ile są one aktywne, przy czym, jeżeli nie istnieją konflikty, to kilka przejść może zostać wykonanych współbieżnie.

Podczas pracy sieci mogą ulegać zmianom znakowania, a tym samym liczba znaczników w poszczególnych jej miejscach. Ze względu na osiągane stany oraz częstotliwość wykonywania poszczególnych przejść definiuje się m.in. pojęcia, takie jak: ograniczoność miejsc, żywotność przejść oraz ograniczoność, zachowawczość, odtwarzalność, odwracalność, żywotność i sprawiedliwość sieci. Własności te są definiowane analogicznie dla dowolnej klasy sieci.

Sieci Petriego są postrzegane najczęściej jako formalizm służący do modelowania systemów współbieżnych. W stosunkowo prosty sposób pozwalają one modelować zjawiska, takie jak: współbieżność procesów, wzajemne wykluczanie, komunikację synchroniczną i asynchroniczną itp.

Sieci uogólnione

Wykonanie przejścia w sieciach znakowanych powoduje usunięcie pojedynczych znaczników z miejsc wejściowych wykonywanego przejścia i dodanie pojedynczych znaczników do jego miejsc wyjściowych. Podstawowym rozszerzeniem pojęcia sieci znakowanych są sieci uogólnione, które umożliwiają usuwanie lub dawanie do jednego miejsca większej liczby znaczników przy wykonaniu pojedynczego przejścia (zob. prace: [48], [56], [70]). Uzyskuje się to dzięki przypisaniu łukom wag, interpretowanych jako ich krotność. Rozszerzenie to jest równoznaczne z możliwością łączenia określonej pary węzłów sieci kilkoma łukami. Wynikiem takiego postępowania jest możliwość zabierania z danego miejsca (lub przesyłania do niego) pewnej skończonej liczby znaczników, równej krotności odpowiedniego łuku.

W tym rozdziale omówimy formalną definicję sieci uogólnionych oraz przykłady modelowania z ich zastosowaniem. Sieci znakowane są szczególnym przypadkiem sieci uogólnionych, w których wszystkie wagi łuków są równe 1. Większość przedstawionych w rozdz. 3 definicji własności sieci znakowanych dotyczy również sieci uogólnionych. Zmienia się tylko definicja aktywności przejścia i sposób wyznaczania znakowania powstałego w wyniku wykonania przejścia.

4.1. Dynamika sieci

Możliwość przesyłania większej liczby znaczników przy odpalaniu pojedynczego przejścia ma wpływ na aktywność danego przejścia oraz zmianę poszczególnych znakowań sieci.

Definicja 4.1. Uporządkowaną piątkę postaci $\mathcal{N} = (P, T, A, W, M_0)$ nazywamy *siecią uogólnioną*, jeżeli są spełnione warunki:

- (1) $\mathcal{N} = (P, T, A)$ jest siecią.

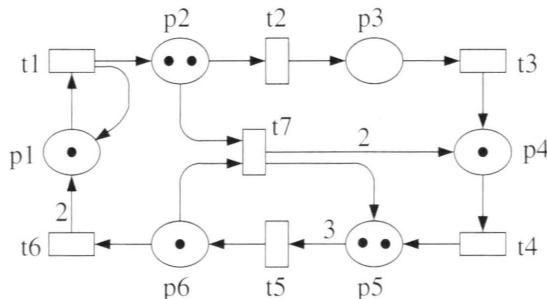
- (2) $W: A \rightarrow \mathbb{N}$ jest funkcją wag łuków. Funkcja W przyporządkowuje każdemu łukowi sieci liczbę naturalną, interpretowaną jako waga (*krotność*) łuku.
(3) $M_0: P \rightarrow \mathbb{Z}_+$ jest znakowaniem początkowym sieci \mathcal{N} . ■

Wagi łuków reprezentujemy graficznie za pomocą etykiet umieszczanych przy odpowiednich łukach. Przy opisie sieci są pomijane wagi równe 1. Wagę łuku (x, y) ($x, y \in V$) będziemy oznaczać symbolem $W(x, y)$ ¹.

Definicja 4.2. Przejście $t \in T$ jest *aktywne* przy znakowaniu M (jest M -aktywne), jeżeli każde z jego miejsc wejściowych zawiera co najmniej tyle znaczników, ile wynosi waga łuku prowadzącego od tego miejsca do przejścia t . ■

Jeżeli przejście t jest M -aktywne, to nowe znakowanie M' uzyskane w wyniku wykonania przejęcia t jest określone w sposób następujący:

$$M'(p) = \begin{cases} M(p) - W(p, t), & \text{gdy } p \in In(t) - Out(t), \\ M(p) + W(t, p), & \text{gdy } p \in Out(t) - In(t), \\ M(p) - W(p, t) + W(t, p), & \text{gdy } p \in In(t) \cap Out(t), \\ M(p) & \text{w pozostałych przypadkach.} \end{cases} \quad (4.1)$$



Rys. 4.1. Sieć uogólniona

Rozważmy przykład sieci uogólnionej przedstawionej na rys. 4.1. Przy znakowaniu początkowym $M_0 = (1, 2, 0, 1, 2, 1)$ aktywne są przejścia $t1, t2, t4, t6$ i $t7$. Wykonanie każdego z tych przejść prowadzi odpowiednio do znakowań $M_1 = (1, 3, 0, 1, 2, 1)$, $M_2 = (1, 1, 1, 1, 2, 1)$, $M_4 = (1, 2, 0, 0, 3, 1)$, $M_6 = (3, 2, 0, 1, 2, 0)$ i $M_7 = (1, 1, 0, 3, 3, 0)$. Przejścia $t3$ i $t5$ nie są aktywne, ponieważ odpowiednio miejsce $p3$ jest puste, a miejsce $p5$ zawiera zbyt mało znaczników.

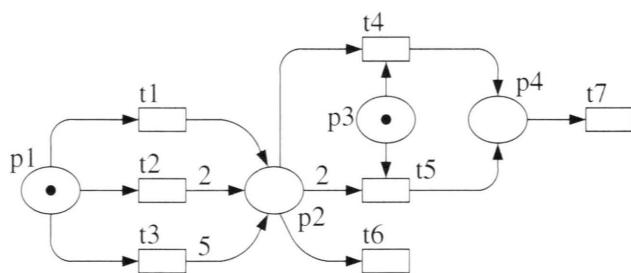
¹ Formalnie zapis powinien mieć postać $W((x, y))$, ale dla uproszczenia podwójne nawiasy będą pomijane.

4.2. Modelowanie z zastosowaniem sieci uogólnionych

W tym podrozdziale omówimy trzy przykłady zastosowania sieci uogólnionych do modelowania systemów współbieżnych: model dystrybutora napojów, model problemu czytelników i pisarzy oraz model systemu producent-konsument.

4.2.1. Model dystrybutora napojów

Rozważmy przykład dystrybutora napojów. Przyjmijmy dla uproszczenia, że są tylko dwa napoje: herbata w cenie 1 zł i kawa w cenie 2 zł. Dystrybutor przyjmuje monety o nominałach 1 zł, 2 zł i 5 zł oraz wydaje resztę. Założymy ponadto, że istnieje możliwość wrzucenia tylko jednej monety. Po wrzuceniu monety klient może wybrać rodzaj napoju lub zrezygnować i z powrotem odebrać uiszoną kwotę. Sieć na rysunku 4.2 modeluje pojedynczy cykl pracy dystrybutora.



Rys. 4.2. Model dystrybutora napojów

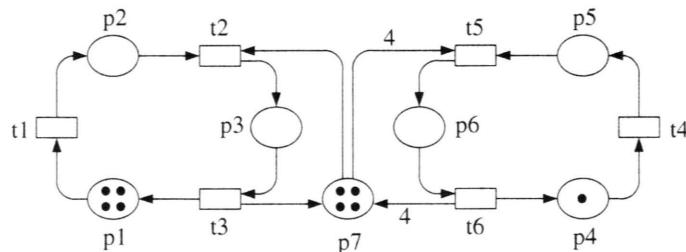
Znaczenie poszczególnych elementów sieci N jest następujące:

- Znacznik w miejscu $p1$ oznacza gotowość dystrybutora do pracy.
- Przejścia $t1$, $t2$ i $t3$ reprezentują wrzucenie monety do dystrybutora, odpowiednio o nominałach 1 zł, 2 zł i 5 zł. Wartość wrzuconej monety jest reprezentowana przez odpowiednią liczbę znaczników, które będą dodane do miejsca $p2$.
- Przejścia $t4$ i $t5$ reprezentują wybór napoju – odpowiednio herbaty i kawy. Miejsce $p3$ służy do zagwarantowania, że będzie wybrany tylko jeden napój. Wydawany napój jest reprezentowany przez znacznik dodawany do miejsca $p4$. Odbiór napoju przez klienta reprezentuje przejście $t7$.
- Przejście $t6$ reprezentuje rezygnację z zakupu napoju i odebranie monety lub odebranie reszty po zakupie napoju. Warto zwrócić uwagę, że przejście to musi zostać wykonane dokładnie tyle razy, ile złotówek należy odebrać.

Zbiór znakowań osiągalnych sieci $\mathcal{R}(M_0)$ jest zbiorem 17-elementowym. Sieć jest dokładnie $\mathcal{L}1$ -żiva, osiąga dwa martwe znakowania (interpretowane jako zakończenie obsługi klienta): $M_1 = (0, 0, 0, 0)$ i $M_2 = (0, 0, 1, 0)$ (w drugim przypadku klient zrezygnował z zakupu napoju). Aby była możliwa obsługa kolejnego klienta, sieć należałoby tak zmodyfikować, by znakowanie początkowe było odwracalne. Miejsce p_2 sieci jest 5-ograniczone, pozostałe miejsca są bezpieczne.

4.2.2. Problem czytelników i pisarzy

Jako drugi przykład posłuży nam model wzajemnego wykluczania dla problemu czytelników i pisarzy. Zadanie polega na zsynchronizowaniu dwóch grup cyklicznych procesów konkurujących o dostęp do wspólnej *czytelni*. Proces *czytelnik* co jakiś czas odczytuje informację zgromadzoną w czytelni i może to robić razem z innymi czytelnikami. Proces *pisarz* co jakiś czas zapisuje nową informację i musi wówczas przebywać w czytelni sam. Zakładamy, że w czytelni może się znajdować maksymalnie czterech czytelników. Sieć \mathcal{N} stanowiącą model dla systemu z czterema czytelnikami i jednym pisarzem przedstawiono na rys. 4.3.



Rys. 4.3. Model wzajemnego wykluczania dla problemu czytelników i pisarzy

Znaczenie poszczególnych elementów sieci jest następujące:

- Miejsca p_3 i p_6 reprezentują czytelnię. Każdy znaczek w miejscu p_3 oznacza jednego czytelnika znajdującego się w czytelni, a znaczek w miejscu p_6 oznacza, że w czytelni znajduje się pisarz.
- Miejsce p_7 jest używane w celu zapewnienia wzajemnego wykluczania. Pisarz może wejść do czytelni, o ile jest pusta. Przy wejściu (przejście t_5) usuwa on wszystkie znaczki z miejsca p_7 , uniemożliwiając w ten sposób czytelnikom wejście do czytelni.
- Miejsca p_2 i p_5 reprezentują kolejkę do czytelni. Znaczki w tych miejscach oznaczają odpowiednio czytelników i pisarza, oczekujących na wejście

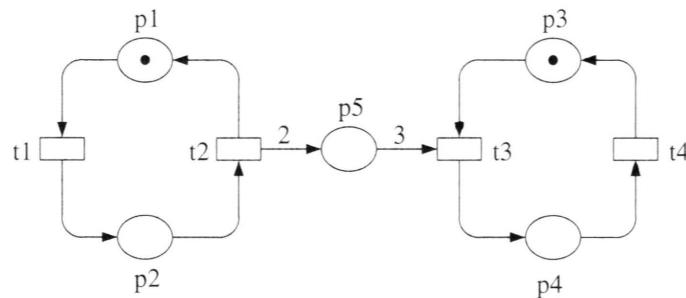
do czytelni. Znaczniki w miejscach p_1 i p_4 oznaczają odpowiednio czytelników i pisarza, znajdujących się poza czytelnią.

Sieć N nie jest siecią sprawiedliwą. Możliwe jest zagłodzenie procesu pisarza. Można bowiem wskazać taki ciąg wykonień przejść, przy którym pisarz nigdy nie wchodzi do czytelni, gdyż przebywa tam zawsze co najmniej jeden czytelnik.

Sieć N jest siecią żywą, ograniczoną i odwracalną. Miejsca p_4 , p_5 i p_6 są bezpieczne, pozostałe miejsca są 4-ograniczone. Zmieniając liczbę znaczników w miejscach p_1 i p_4 , można sterować liczbą czytelników i pisarzy w modelowanym systemie. Znakowanie początkowe miejsca p_7 odzwierciedla liczbę miejsc w czytelni.

4.2.3. Model systemu producent-konsument

Rozważmy ponownie system złożony z dwóch grup cyklicznych procesów. Z jednej strony występuje *producent*, który dostarcza w każdym cyklu swojej pracy pewną ilość *towaru* do *magazynu*, z drugiej strony *konsument*, który w każdym cyklu swojej pracy pobiera pewną ilość *towaru*. *Magazyn* w rozważanym systemie pełni funkcję bufora, przez który jest przekazywany *twarz* (np. informacje). Na rysunku 4.4 przedstawiono sieć uogólnioną modelującą ten system. W modelu uwzględniono jednego producenta, jednego konsumenta i nieograniczoną pojemność magazynu. W każdym cyklu pracy producenta są dostarczane do magazynu dwie jednostki towaru, a w każdym cyklu konsumenta są pobierane trzy jednostki.



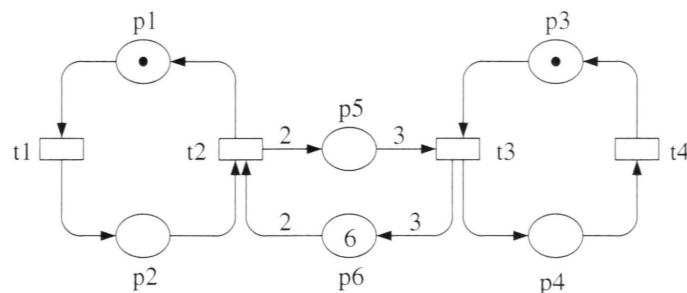
Rys. 4.4. Model systemu producent-konsument

Producenta reprezentuje znacznik krążący między miejscami p_1 i p_2 . Każde wykonanie przejścia t_2 powoduje dodanie do magazynu dwóch jednostek towaru. Analogicznie znacznik krążący między miejscami p_3 i p_4 reprezentuje konsumenta. Wykonanie przejścia t_3 powoduje usunięcie z magazynu trzech jednostek

towaru. Zwiększenie liczby znaczników w tych miejscach sieci powoduje dodanie do systemu kolejnych producentów lub konsumentów.

Sieć N przedstawiona na rys. 4.4 jest siecią żywą. Miejsce p_5 jest miejscem nieograniczonym, podczas gdy pozostałe miejsca sieci są bezpieczne. Znakowanie początkowe sieci jest odwracalne. Ze względu na nieograniczonosć miejsca p_5 , zbiór znakowań osiągalnych sieci jest zbiorem nieskończonym. Sieć nie jest sprawiedliwa, choć można podać sprawiedliwe ciągi przejść.

Wyeliminowanie nieograniczonej pojemności magazynu wymaga umieszczenia dodatkowego miejsca, które (podobnie jak miejsce p_7 na rys. 4.3) będzie ograniczało liczbę znaczników dodawanych do miejsca p_5 . Tak zmodyfikowaną sieć N' przedstawiono na rys. 4.5. Dla magazynu przyjęto pojemność równą 6. Sieć przedstawiona na rys. 4.5 jest siecią sprawiedliwą.



Rys. 4.5. Model systemu producent-konsument z ograniczonym magazynem

4.3. Podsumowanie

W tym rozdziale omówiliśmy podstawowe informacje dotyczące sieci uogólnionych, które powstały przez rozszerzenie sieci znakowanych o funkcję wag łuków, definiującą ich krotność. W wyniku takiego postępowania przy wykonywaniu dowolnego przejścia z jego miejsc wejściowych są usuwane nie pojedyncze znaczniki, ale zbiory znaczników o rozmiarze określonym przez wagę odpowiedniego łuku. Analogicznie są również dodawane zbiory znaczników do miejsc wyjściowych danego przejścia. Omawiane w rozdz. 3 sieci znakowane są szczególnym przypadkiem sieci uogólnionych, w których wszystkie wagi łuków są równe 1.

Analiza sieci uogólnionych

Podstawową zaletą stosowania sieci Petriego do modelowania systemów współbieżnych jest możliwość dokonania formalnej analizy stworzonego modelu. Sieci uogólnione stanowią tę klasę sieci Petriego, którą cechuje największa różnorodność opracowanych metod analizy. Jednocześnie metody te są podstawą do opracowywania metod analizy dla innych klas sieci, a bardzo często można ich używać bez żadnych modyfikacji.

W tym rozdziale omówimy najczęściej spotykane metody analizy sieci uogólnionych. Najpierw zajmiemy się metodami opartymi na analizie zbioru osiągalnych znakowań, reprezentowanym w postaci grafu skierowanego. Reprezentacja taka może mieć zarówno formę pełnego grafu osiągalności, w którym każde znakowanie jest reprezentowane przez jeden węzeł (zob. prace: [48], [56], [70], [74]), jak również formę zredukowaną, w której jeden węzeł reprezentuje niepusty zbiór osiągalnych znakowań (zob. prace: [20], [32], [48], [61], [62], [63], [72], [74], [84], [85], [86] i [87]).

Druga grupa metod analizy jest związana z algebraiczną reprezentacją sieci. Strukturę sieci uogólnionych można wzajemnie jednoznacznie przedstawić za pomocą dwóch macierzy o współczynnikach całkowitych. Reprezentacja taka pozwala wyznaczyć tzw. macierz incydencji, na której podstawie można obliczyć niezmienne sieci (zob. prace: [44], [45], [56], [66], [74]). Wyrażają one stałe własności modelu i można je stosować przy analizie zarówno ograniczości sieci (niezmienne miejsca), jak i jej żywotności (niezmienne przejęcie).

W kolejnych podrozdziałach omówimy również metody wykorzystujące istnienie specyficznych zbiorów miejsc określanych mianem zatrasków i pułapek (zob. prace: [2], [12], [14], [28], [30], [56] i [74]), oraz metody redukujące rozmiar sieci (zob. prace: [4], [18], [20], [48], [57] i [67]). Analiza istnienia pułapek i zatrasków w sieci pozwala zidentyfikować podzbiory miejsc, które albo pozostają znakowane, albo zawsze pozostają czyste. Metody redukcji pozwalają zmniejszyć liczbę elementów sieci z zachowaniem wybranych jej własności, takich jak: żywotność, bezpieczeństwo, brak znakowań martwych w zbiorze znakowań osiągalnych itp.

5.1. Grafy osiągalności

Zbiór znakowań osiągalnych sieci uogólnionej \mathcal{N} można przedstawić w postaci grafu skierowanego. Węzłami takiego grafu są znakowania sieci, a łuki prowadzące od jednego znakowania do drugiego są etykietowane nazwami przejść, których wykonanie powoduje zmianę znakowania z pierwszego na drugie. Graf taki nazywamy *grafem osiągalności* sieci \mathcal{N} . Dwa węzły takiego grafu mogą być połączone kilkoma łukami, jeżeli wykonanie kilku przejść powoduje identyczne zmiany znakowania sieci.

Niech dana będzie sieć uogólniona $\mathcal{N} = (P, T, A, W, M_0)$.

Definicja 5.1. *Grafem osiągalności* sieci \mathcal{N} nazywamy graf $\mathcal{G} = (V, A, \gamma)$ etykietowany nad zbiorem T^1 , spełniający warunki:

- (1) $V = \mathcal{R}(M_0)$ – jest zbiorem węzłów grafu.
- (2) $A = \left\{ (M, t, M') : M, M' \in \mathcal{R}(M_0) \wedge M \xrightarrow{t} M' \right\}$ – jest zbiorem łuków etykietowanych nazwami przejść, których wykonanie powoduje zmianę znakowania z M na M' .
- (3) $\forall a = (M, t, M') \in A : \gamma(a) = (M, M')$.

■

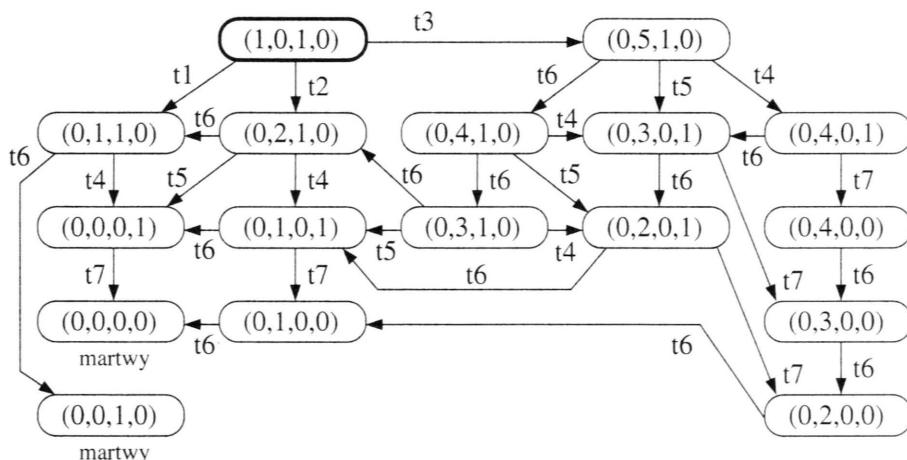
Zbiór znakowań osiągalnych można również przedstawić w postaci *drzewa osiągalności*. Korzeniem drzewa jest znakowanie początkowe sieci. Drzewo może zawierać powtarzające się węzły, które są wówczas opatrzone etykietą *duplikat*. W drzewie osiągalności można w jednoznaczny sposób dojść od korzenia do dowolnego innego węzła. Drzewo osiągalności można przekształcić w graf osiągalności, sklejając duplikaty. Zarówno w przypadku drzew, jak i grafów osiągalności, węzły reprezentujące znakowania martwe będą dodatkowo oznaczane etykietą *martwy*.

Rozważmy przedstawioną na rys. 4.2 sieć modelującą działanie dystrybutora napojów. Znakowanie początkowe $M_0 = (1, 0, 1, 0)$ przyjmujemy za pierwszy węzeł grafu osiągalności (*węzeł początkowy*). Przy znakowaniu początkowym są aktywne trzy przejścia: t_1 , t_2 i t_3 . W wyniku wykonania przejścia t_1 otrzymujemy znakowanie $M_1 = (0, 1, 1, 0)$, a zatem do grafu osiągalności zostanie dodany węzeł reprezentujący znakowanie M_1 oraz łuk z etykietą t_1 , łączący węzły M_0 i M_1 . Analogicznie rozważamy efekt wykonania przejść t_2 i t_3 . Węzeł M_0 będzie miał zatem trzy następniki. W kolejnym kroku analizujemy znakowania reprezentowane

¹ Symbol A jest używany do oznaczenia zarówno zbioru łuków sieci, jak i zbioru łuków grafu osiągalności. Właściwe znaczenie tego symbolu zawsze będzie wynikało bezpośrednio z kontekstu.

przez dodane węzły. Dla każdego aktywnego przejścia będzie dodany nowy węzeł i nowy łuk lub tylko łuk, jeśli odpowiedni węzeł już w grafie osiągalności istnieje. Jeżeli dany węzeł nie ma następców, to reprezentuje on znakowanie martwej (zostanie przy nim umieszczona etykieta *martywy*).

Na rysunku 5.1 przedstawiono graf osiągalności dla sieci modelującej działanie dystrybutora napojów (wyróżniono węzeł początkowy grafu). Łatwo sprawdzić, że wszystkie możliwe ciągi przejścia prowadzą do jednego z dwóch martwych znaków.



Rys. 5.1. Graf osiągalności dla sieci modelującej działanie dystrybutora napojów

Jeżeli sieć uogólniona jest ograniczona, to liczba znakowań osiągalnych ze znakowania początkowego jest skończona, a zatem również graf osiągalności jest skończony. Przedstawia on w sposób jawny wszystkie znakowania sieci osiągalne ze znakowania początkowego, oraz przejścia między poszczególnymi znakowaniami. Na podstawie analizy węzłów grafu osiągalności można sprawdzić ograniczonosć poszczególnych miejsc.

Twierdzenie 5.1. Niech $\mathcal{G} = (V, A, \gamma)$ będzie grafem osiągalności sieci N .

- (1) Miejsce p_i sieci \mathcal{N} jest znakowane wtedy i tylko wtedy, gdy w grafie \mathcal{G} istnieje węzeł, dla którego i -ta współrzędna jest różna od zera.
 - (2) Miejsce p_i sieci \mathcal{N} jest bezpieczne wtedy i tylko wtedy, gdy i -ta współrzędna wszystkich węzłów grafu \mathcal{G} jest równa 0 lub 1.
 - (3) Miejsce p_i sieci \mathcal{N} jest k -ograniczone ($k \in \mathbb{N}$) wtedy i tylko wtedy, gdy i -ta współrzędna wszystkich węzłów grafu \mathcal{G} jest mniejsza lub równa k . ■

Analizując drogi w grafie osiągalności, można sprawdzić własności związane z osiągalnością znakowań i żywotnością sieci:

Twierdzenie 5.2. Niech $\mathcal{G} = (V, A, \gamma)$ będzie grafem osiągalności sieci \mathcal{N} .

- (1) Niech $M_1, M_2 \in \mathcal{R}(M_0)$. Znakowanie $M_2 \in \mathcal{R}(M_1)$ wtedy i tylko wtedy, gdy w grafie \mathcal{G} istnieje droga od M_1 do M_2 .
- (2) Niech $M \in \mathcal{R}(M_0)$. Ciąg przejść $\alpha \in \mathcal{L}(M)$ wtedy i tylko wtedy, gdy w grafie \mathcal{G} istnieje droga prowadząca od węzła M , taka że kolejne etykiety łuków tej drogi odpowiadają kolejnym elementom ciągu α .
- (3) Znakowanie $M \in \mathcal{R}(M_0)$ jest znakowaniem własnym sieci \mathcal{N} wtedy i tylko wtedy, gdy dla dowolnego węzła $M' \in V$ istnieje droga od M' do M .
- (4) Znakowanie początkowe sieci \mathcal{N} jest odtwarzalne wtedy i tylko wtedy, gdy w grafie \mathcal{G} istnieje węzeł różny od węzła początkowego i droga prowadząca od tego węzła do węzła początkowego.
- (5) Znakowanie $M \in \mathcal{R}(M_0)$ jest znakowaniem martwym sieci \mathcal{N} wtedy i tylko wtedy, gdy węzeł M jest węzłem końcowym grafu \mathcal{G} .
- (6) Sieć \mathcal{N} jest odwracalna wtedy i tylko wtedy, gdy graf \mathcal{G} jest silnie spójny. ■

Twierdzenie 5.3. Niech $\mathcal{G} = (V, A, \gamma)$ będzie grafem osiągalności sieci \mathcal{N} .

- (1) Przejście t sieci \mathcal{N} jest martwe ($\mathcal{L}0$ -żywe) wtedy i tylko wtedy, gdy w grafie \mathcal{G} nie występuje łuk z etykietą t .
- (2) Przejście t sieci \mathcal{N} jest $\mathcal{L}1$ -żywe wtedy i tylko wtedy, gdy w grafie \mathcal{G} występuje łuk z etykietą t .
- (3) Przejście t sieci \mathcal{N} jest $\mathcal{L}2$ -żywe wtedy i tylko wtedy, gdy dla dowolnej liczby $k \in \mathbb{N}$ istnieje droga w grafie \mathcal{G} , w której co najmniej k razy występuje łuk z etykietą t .
- (4) Przejście t sieci \mathcal{N} jest $\mathcal{L}3$ -żywe wtedy i tylko wtedy, gdy istnieje droga w grafie \mathcal{G} , w której łuk z etykietą t występuje nieskończoność wiele razy.
- (5) Jeżeli w grafie \mathcal{G} istnieje cykl, w którym jeden z łuków ma etykietę t , to przejście t jest $\mathcal{L}3$ -żywe.
- (6) Jeżeli graf \mathcal{G} jest grafem silnie spójnym i występuje w nim łuk z etykietą t , to przejście t jest żywe ($\mathcal{L}4$ -żywe).
- (7) Sieć \mathcal{N} jest sprawiedliwa wtedy i tylko wtedy, gdy każda nieskończona droga w grafie \mathcal{G} zawiera nieskończoność wiele wystąpień łuku z etykietą t dla dowolnego $t \in T$. ■

Sformułowane powyżej twierdzenia wynikają bezpośrednio z definicji grafu osiągalności i definicji własności sieci, przedstawionych w rozdz. 3.

Definicja 5.2. Węzeł M grafu $\mathcal{G} = (V, A, \gamma)$ nazywamy pełnym, jeżeli dla każdego przejścia $t \in T$ istnieje droga prowadząca od węzła M , która zawiera łuk etykietowany przejściem t . ■

Twierdzenie 5.4. Sieć N jest żywa wtedy i tylko wtedy, gdy wszystkie węzły grafu \mathcal{G} są pełne. ■

Dowód. Niech $M \in \mathcal{R}(M_0)$ będzie znakowaniem osiągalnym sieci N . Jeżeli sieć jest żywa, to dla dowolnego przejścia $t \in T$ istnieje znakowanie $M' \in \mathcal{R}(M)$, przy którym przejście t jest aktywne. Istnieje zatem również droga prowadząca od węzła M , która zawiera łuk etykietowany przejściem t . Węzeł M (w grafie osiągalności) musi zatem być węzłem pełnym. Z dowolności wyboru znakowania M wynika, że wszystkie węzły grafu osiągalności są węzłami pełnymi. Z drugiej strony, jeżeli wszystkie węzły grafu osiągalności są pełne, to wszystkie przejścia sieci są potencjalnie wykonalne dla każdego znakowania $M \in \mathcal{R}(M_0)$, a zatem sieć jest żywa. □

W rozdziałach 3 i 4 omówiliśmy przykłady sieci wraz z opisem ich własności. Opierając się na przedstawionych w tym rozdziale twierdzeniach, można dowieść, że sieci te istotnie mają przypisane im własności.

Poniżej na kolejnych rysunkach możemy zobaczyć grafy osiągalności dla wybranych modeli. Zostały pominione sieci przedstawione na rys. 3.14 i 3.15, gdyż grafy osiągalności dla tych sieci są zaledwie trzyelementowe, oraz grafy dla sieci na rys. 3.16 i 3.17, gdyż grafy te, mimo że są skończone, są zbyt duże, by można je było zaprezentować.

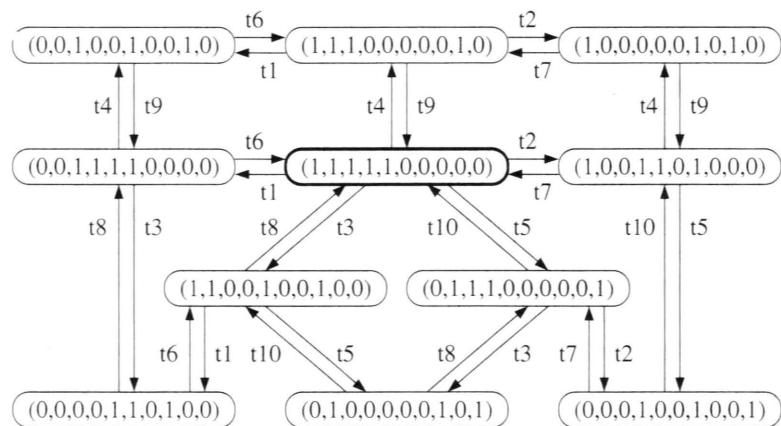
Rozważmy model problemu pięciu filozofów przedstawiony na rys. 3.18. Graf osiągalności dla tej sieci pokazano na rys. 5.2. Sieć ma następujące własności (wynikające z podanych twierdzeń): wszystkie miejsca są znakowane (twierdzenie 5.1 (1)), sieć jest bezpieczna (twierdzenie 5.1 (2)), odwracalna (twierdzenie 5.2 (6)), żywa (twierdzenie 5.3 (6)), nie jest sprawiedliwa (twierdzenie 5.3 (7)).

Graf osiągalności dla modelu protokołu komunikacyjnego (rys. 3.19), przedstawiono na rys. 5.3. Sieć jest bezpieczna, żywa, żywa ze względu na miejsca i odwracalna, lecz nie jest siecią sprawiedliwą.

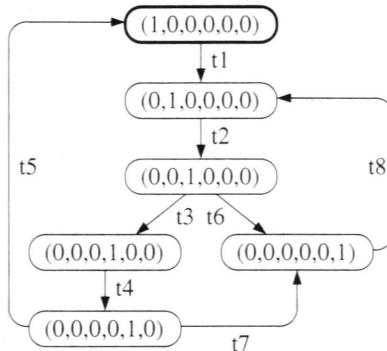
Rysunek 5.4 zawiera graf osiągalności dla modelu problemu czytelników i pisarzy (zob. rys. 4.3). Miejsca $p4$, $p5$ i $p6$ sieci są bezpieczne, a pozostałe miejsca są 4-ograniczone, sieć jest żywa, żywa ze względu na miejsca i odwracalna, lecz nie jest siecią sprawiedliwą.

Na rysunku 5.5 przedstawiono graf osiągalności dla modelu systemu producent-konsument z ograniczonym magazynem (zob. rys. 4.5). Miejsca $p5$ i $p6$ sieci są

6-ograniczone, a pozostałe miejsca są bezpieczne. Sieć jest żywa, żywa ze względu na miejsca, odwracalna i sprawiedliwa.

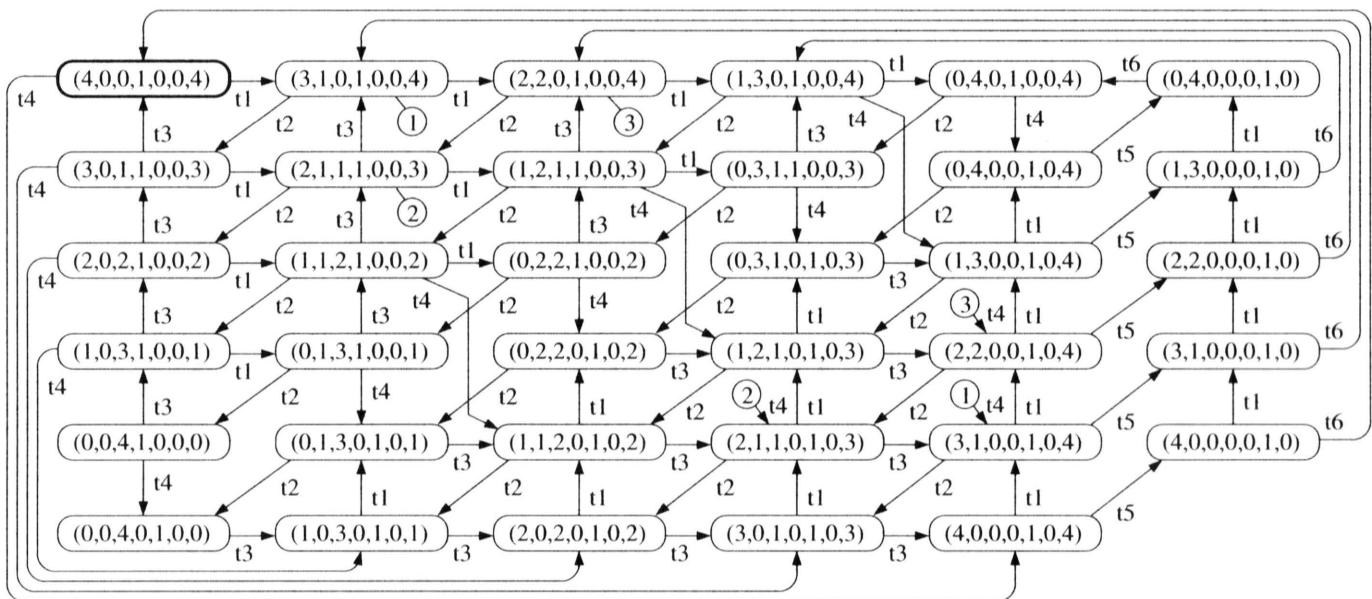


Rys. 5.2. Graf osiągalności dla sieci modelującej problem pięciu filozofów

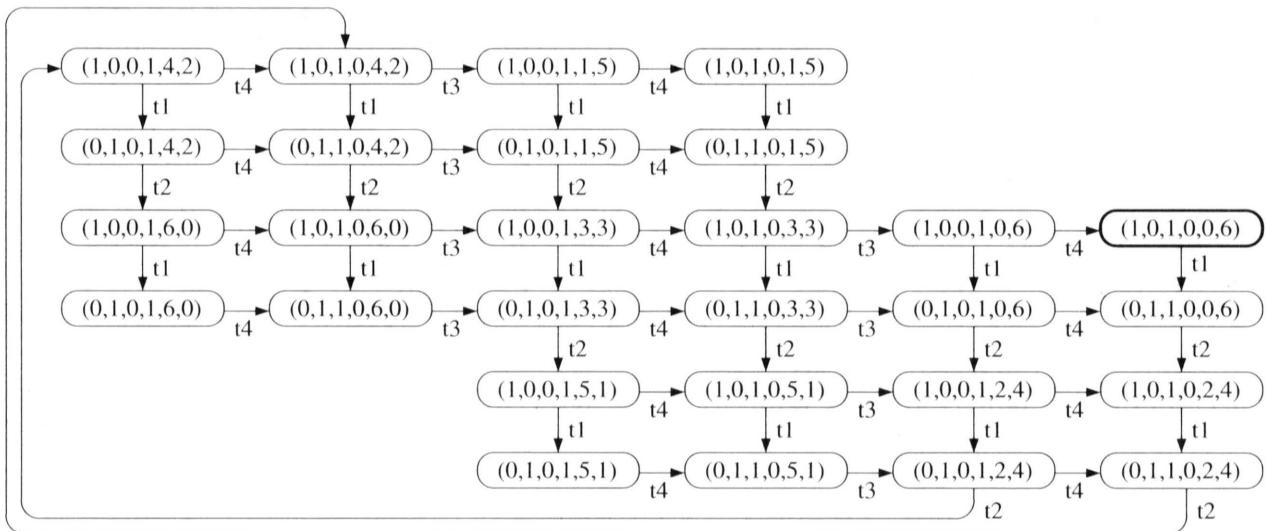


Rys. 5.3. Graf osiągalności dla sieci modelującej protokół komunikacyjny

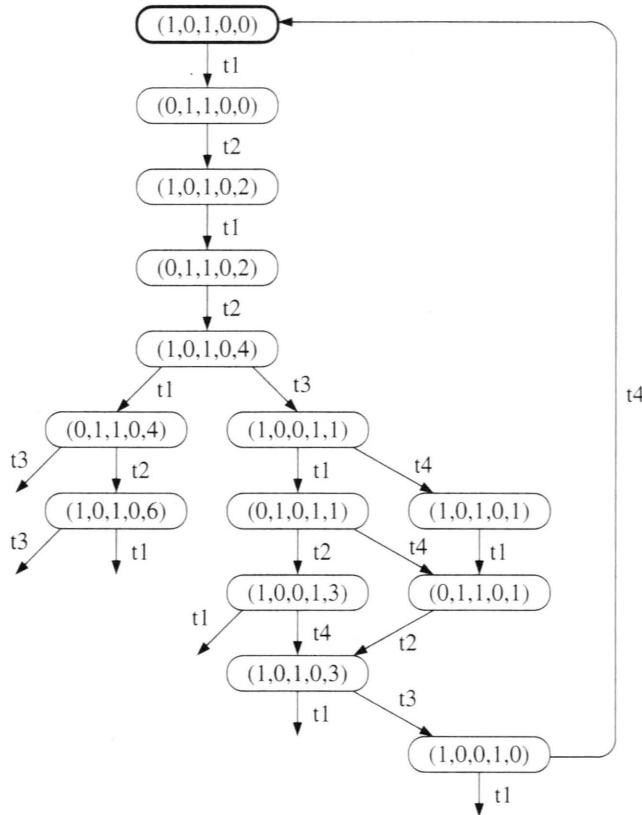
Rozważmy model systemu producent-konsument z nieograniczonym magazynem (zob. rys. 4.4). Zbiór znakowań osiągalnych dla tej sieci jest zbiorem nieskończonym (wynika to z nieograniczoneści miejsca p_5), a zatem graf osiągalności dla tej sieci jest również nieskończony. Fragment tego grafu osiągalności przedstawiono na rys. 5.6 (na rysunku pominięto łuki wejściowe prowadzące od niewidocznych węzłów). Fragment taki może być niewystarczający do analizy własności sieci. W następnych podrozdziałach omówimy metody reprezentacji nieskończonego zbioru znakowań za pomocą skończonego grafu skierowanego.



Rys. 5.4. Graf osiągalności dla sieci modelującej problem czytelników i pisarzy



Rys. 5.5. Graf osiągalności dla sieci modelującej system producent-konsument



Rys. 5.6. Fragment grafu osiągalności dla sieci modelującej system producent-konsument z nieograniczonym magazynem

5.2. Grafy pokrycia

Rozważmy ponownie sieć przedstawioną na rys. 4.4. W zbiorze znakowań osiągalnych tej sieci znajdują się wszystkie znakowania postaci $(1, 0, 1, 0, n)$, przy czym $n \in \mathbb{N}$. W grafie osiągalności każde z tych znakowań jest reprezentowane przez jeden węzeł, powodując, że zbiór węzłów grafu jest nieskończony. Jednocześnie tak duża liczba *podobnych* węzłów nie wzbogaca znacząco naszej wiedzy o właściwościach sieci. W grafie pokrycia rozważany zbiór węzłów jest reprezentowany przez jeden węzeł pokrywający o postaci $(1, 0, 1, 0, \infty)$.

Niech dana będzie sieć uogólniona $\mathcal{N} = (P, T, A, W, M_0)$.

Definicja 5.3. Znakowanie M sieci \mathcal{N} nazywamy *pokrywalnym*, jeżeli istnieje znakowanie $M' \in \mathcal{R}(M_0)$, takie że dla każdego miejsca $p \in P$ jest spełniony warunek $M'(p) \geq M(p)$. ■

Niech dane będą znakowania $M_1, M_2 \in \mathcal{R}(M_0)$. Przypuśćmy, że dla każdego miejsca p są spełnione nierówności: $M_1(p) \leq M_2(p)$ oraz $M_1 \neq M_2$ (piszemy wtedy $M_1 < M_2$). Każde przejście aktywne przy znakowaniu M_1 jest również aktywne przy znakowaniu M_2 . Tak więc, powtarzając ciąg przejść, które prowadzą od M_1 do M_2 , otrzymujemy nowe znakowanie M_3 , przy czym $M_2 < M_3$. Iterując tę procedurę, generujemy nieskończony ciąg różnych znakowań (M_i) dla $i = 1, 2, \dots$. Ciąg taki może być reprezentowany tzw. *węzłem pokrywającym* M , przy czym $M(p) = M_1(p)$, jeśli $M_1(p) = M_2(p)$, oraz $M(p) = \infty$, jeśli liczba znaczników w miejscu p wzrasta nieograniczenie.

Ze względu na powyższe rozważania musimy zmodyfikować definicję znakowania sieci. Przy tworzeniu grafu pokrycia będą dopuszczalne również znakowania, dla których liczba znaczników w danym miejscu sieci jest nieograniczona. Symbolem ∞ będziemy oznaczać liczbę nieskończoną, zaś zbiór $\mathbb{Z}_+ \cup \{\infty\}$ – symbolem \mathbb{Z}_+^∞ . Relację słabej mniejszości oraz operacje dodawania i odejmowania można rozciągnąć na zbiór \mathbb{Z}_+^∞ w sposób następujący:

$$\forall n \in \mathbb{Z}_+^\infty : n \leq \infty, \quad (5.1)$$

$$\forall n \in \mathbb{Z}_+ : \infty + n = \infty \wedge \infty - n = \infty. \quad (5.2)$$

Definicja 5.4. Znakowaniem uogólnionym sieci \mathcal{N} nazywamy dowolną funkcję:

$$M: P \rightarrow \mathbb{Z}_+^\infty. \quad (5.3)$$

■

Od tej pory terminu *znakowanie* będziemy używać również w odniesieniu do znakowań uogólnionych.

Zajmiemy się teraz algorytmem tworzenia drzewa pokrycia. Graf pokrycia otrzymujemy z drzewa pokrycia przez scalenie duplikatów. Oto ten algorytm:

1. Jako korzeń drzewa przyjmujemy węzeł odpowiadający znakowaniu M_0 .
2. Niech M_i oznacza dowolny węzeł w drzewie pokrycia sieci \mathcal{N} . Nowy węzeł tworzymy w sposób następujący:
Jeżeli M_i jest znakowaniem martwym, to jest końcem rozważanej gałęzi.
W przeciwnym przypadku bierzemy pod uwagę wszystkie znakowania sieci,

które można uzyskać w wyniku wykonania przejść aktywnych przy znakowaniu M_i . Niech M_j oznacza jedno z uzyskanych w ten sposób znakowań.

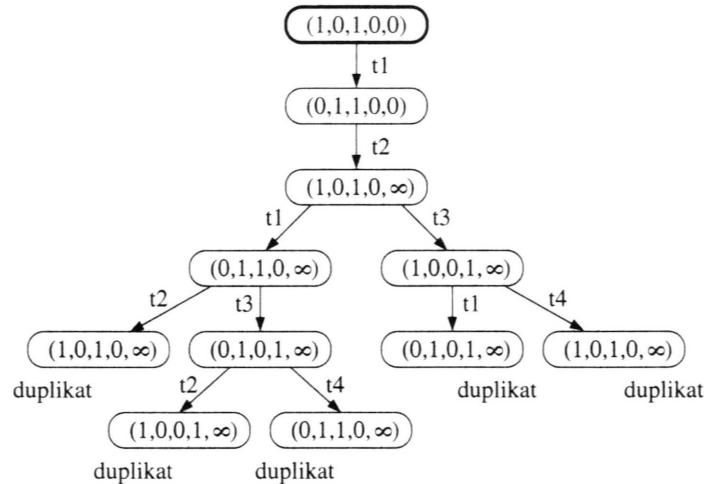
- Jeżeli w drzewie istnieje węzeł reprezentujący takie samo znakowanie, to do drzewałączamy nowy węzeł M_j i nazywamy go duplikatem (koniec budowy tej gałęzi).
- Jeżeli w drzewie na drodze od węzła początkowego do rozpatrywanego węzła M_j istnieje węzeł $M_k < M_j$, to do drzewałączamy nowy węzeł M'_j , taki że $M'_j(p) = M_j(p)$, gdy $M_k(p) = M_j(p)$, oraz $M'_j(p) = \infty$, gdy $M_k(p) < M_j(p)$. Rozważany węzeł M_j nie będziełączony do drzewa pokrycia.
- Jeżeli nie zachodzi żaden z powyższych przypadków, to do drzewałączamy nowy węzeł odpowiadający znakowaniu M_j .

Zanalizujmy przedstawiony na rys. 4.4 model systemu producent-konsument z nieograniczonym magazynem. Dla sieci zbudujemy drzewo pokrycia. Jako korzeń drzewa przyjmujemy znakowanie początkowe $M_0 = (1, 0, 1, 0, 0)$. Przy znakowaniu początkowym M_0 aktywne jest tylko przejście $t1$, a jego wykonanie prowadzi do znakowania $M_1 = (0, 1, 1, 0, 0)$. Do drzewa pokryciałączamy węzeł M_1 razem z łukiem $(M_0, t1, M_1)$. Przy znakowaniu M_1 aktywne jest przejście $t2$, a jego wykonanie prowadzi do znakowania $M_2 = (1, 0, 1, 0, 2)$. Ponieważ $M_2 > M_0$, do drzewałączamy węzeł $M'_2 = (1, 0, 1, 0, \infty)$ oraz łuk $(M_1, t2, M'_2)$. Przy znakowaniu M'_2 aktywne są przejścia $t1$ i $t3$. W wyniku ich wykonania otrzymujemy odpowiednio znakowania $M_3 = (0, 1, 1, 0, \infty)$ i $M_4 = (1, 0, 0, 1, \infty)$. Obydwa węzły sąłączane do drzewa razem z odpowiednimi łukami. Budując dalej drzewo pokrycia, otrzymujemy jeszcze węzeł $M_5 = (0, 1, 0, 1, \infty)$ i pięć duplikatów. Kompletne drzewo pokrycia dla rozważanej sieci przedstawiono na rys. 5.7, a graf pokrycia (otrzymany przez sklejenie duplikatów), na rys. 5.8.

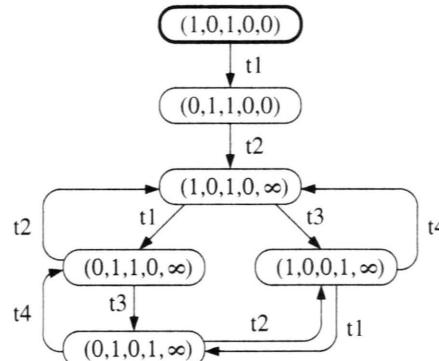
Twierdzenie 5.5. Graf pokrycia sieci uogólnionej \mathcal{N} jest grafem skończonym. ■

Dowód. Dowód tego twierdzenia opiera się na fakcie, że z każdego nieograniczonego zbioru znakowań można zawsze wybrać ciąg nieskończony i silnie monotoniczny. Przypuśćmy, że graf pokrycia jest nieskończony. Istnieje więc w nim droga rozpoczęjąca się od węzła początkowego, nieskończona i niepowtarzalna, która nie prowadzi przez żaden cykl. Z konstrukcji grafu pokrycia wynika, że znakowania węzłów tej drogi byłyby parami różne. Tworzyłyby więc nieskończony zbiór, z którego można wybrać ciąg nieskończony i silnie monotoniczny.

Dla każdego ciągu rosnącego znakowanie miejsc nie może wzrastać bez końca, ponieważ po skończonej liczbie kroków (co najwyżej $n = |P|$) otrzymamy znakowanie ∞ . Dla każdego ciągu malejącego znakowanie miejsc może maleć przez co najwyżej $n = \sum_{p \in P} M(p)$ kroków. Wynika stąd, że nie może istnieć taki nie-



Rys. 5.7. Drzewo pokrycia dla sieci modelującej system producent-konsument z nieograniczonym magazynem



Rys. 5.8. Graf pokrycia dla sieci modelującej system producent-konsument z nieograniczonym magazynem

skończony podciąg, jak również nieskończona droga wewnątrz grafu pokrycia. Przeczy to hipotezie, że graf pokrycia jest grafem nieskończonym. (Pełny dowód tego twierdzenia można znaleźć w pracach [37] i [70]). \square

Przedstawione twierdzenie jest bardzo ważnym argumentem przemawiającym za stosowaniem grafów pokrycia. Umożliwiają one badanie własności sieci o nieskończonym zbiorze osiągalnych znakowań na podstawie skończonego grafu pokrycia.

Każdy węzeł grafu reprezentuje znakowanie sieci, przy czym niektóre węzły reprezentują rzeczywiście osiągalne znakowania, a inne tylko je pokrywają. Dla sieci ograniczonych graf osiągalności i graf pokrycia są identyczne. W takim przypadku żaden z węzłów grafu nie zawiera ∞ -składowej.

Wniosek 5.1. Niech $\mathcal{G} = (V, A, \gamma)$ będzie grafem pokrycia sieci N . Zbiór znakowań osiągalnych $\mathcal{R}(M_0)$ jest zbiorem skończonym wtedy i tylko wtedy, gdy żaden węzeł grafu \mathcal{G} nie ma ∞ -składowej. ■

Dla grafów pokrycia można sformułować twierdzenia analogiczne do twierdzeń przytoczonych w poprzednim podrozdziale.

Twierdzenie 5.6. Niech $\mathcal{G} = (V, A, \gamma)$ będzie grafem pokrycia sieci N .

- (1) Miejsce p_i sieci N jest znakowane wtedy i tylko wtedy, gdy w grafie \mathcal{G} istnieje węzeł, dla którego i -ta współrzędna jest różna od zera.
- (2) Miejsce p_i sieci N jest bezpieczne wtedy i tylko wtedy, gdy i -ta współrzędna wszystkich węzłów grafu \mathcal{G} jest równa 0 lub 1.
- (3) Miejsce p_i sieci N jest k -ograniczone ($k \in \mathbb{N}$) wtedy i tylko wtedy, gdy i -ta współrzędna wszystkich węzłów grafu \mathcal{G} jest mniejsza lub równa k .
- (4) Miejsce p_i sieci N jest nieograniczone wtedy i tylko wtedy, gdy istnieje węzeł $M \in V$, taki że $M(p_i) = \infty$. ■

Sformułowane powyżej tezy twierdzenia wynikają bezpośrednio z algorytmu konstrukcji grafu pokrycia. Grafy pokrycia uniemożliwiają tak dokładną analizę osiągalności znakowań, jaką przedstawiono w twierdzeniu 5.2. Każdemu ciągowi przejść $\alpha = t_1, t_2, \dots, t_k \in \mathcal{L}(M_0)$ odpowiada droga w grafie osiągalności sieci, a ponadto droga ta jednoznacznie wyznacza osiągane pośrednie znakowania, tzn., jeżeli $M_0 \xrightarrow{\alpha} M_k$, to z grafu osiągalności można odczytać znakowania M_i (dla $i = 1, 2, \dots, k - 1$), takie że

$$M_0 \xrightarrow{t_1} M_1 \xrightarrow{t_2} M_2 \xrightarrow{t_3} \cdots \xrightarrow{t_k} M_k. \quad (5.4)$$

Ciągovi przejść α odpowiada również droga w grafie pokrycia sieci. Droga ta wyznacza węzły M'_0, M'_1, \dots, M'_k , takie że $M_0 = M'_0$ i $M_i \leq M'_i$, dla $i = 1, 2, \dots, k$. Jeżeli węzeł M'_i nie zawiera ∞ -składowej, to reprezentuje on rzeczywiście osiągalne znakowanie i wówczas zachodzi równość $M_i = M'_i$. W przeciwnym przypadku węzeł reprezentuje jedynie znakowanie pokrywające.

Rozważmy fragment grafu osiągalności przedstawiony na rys. 5.6. Ciągowi przejść $\alpha = t_1, t_2, t_1, t_2, t_3, t_4$ odpowiada droga w grafie osiągalności, która wyznacza ciąg znakowań: $M_0 = (1, 0, 1, 0, 0)$, $M_1 = (0, 1, 1, 0, 0)$, $M_2 = (1, 0, 1, 0, 2)$, $M_3 = (0, 1, 1, 0, 2)$, $M_4 = (1, 0, 1, 0, 4)$, $M_5 = (1, 0, 0, 1, 1)$, $M_6 = (1, 0, 1, 0, 1)$, a także droga w grafie pokrycia (zob. rys. 5.8), która wyznacza ciąg znakowań: $M'_0 = (1, 0, 1, 0, 0)$, $M'_1 = (0, 1, 1, 0, 0)$, $M'_2 = (1, 0, 1, 0, \infty)$, $M'_3 = (0, 1, 1, 0, \infty)$, $M'_4 = (1, 0, 1, 0, \infty)$, $M'_5 = (1, 0, 0, 1, \infty)$, $M'_6 = (1, 0, 1, 0, \infty)$.

Martwym znakowaniom sieci należącym do zbioru $\mathcal{R}(M_0)$ odpowiadają węzły końcowe grafu osiągalności, a także węzły końcowe grafu pokrycia, przy czym w grafie pokrycia węzeł końcowy może pokrywać zbiór martwych znakowań.

Twierdzenie 5.7. Niech \mathcal{N} będzie siecią uogólnioną, $\mathcal{G} = (V, A, \gamma)$ grafem pokrycia, a M znakowaniem sieci \mathcal{N} .

- (1) Jeżeli przejście $t \in T$ nie jest aktywne przy znakowaniu M , to nie jest również aktywne przy żadnym znakowaniu $M' \leq M$.
- (2) Jeżeli istnieje węzeł końcowy $M' \in V$, taki że $M \leq M'$, to M jest znakowaniem martwym. ■

Dowód. Pierwsza teza tego twierdzenia wynika bezpośrednio z definicji aktywności przejść (definicja 4.2). Jeżeli przejście t nie jest aktywne przy znakowaniu M , to nie może być aktywne przy żadnym znakowaniu M' , przy którym co najmniej jedno miejsce sieci zawiera mniej znaczników niż przy znakowaniu M . Z pierwszej tezy twierdzenia wynika, że jeżeli znakowanie M jest martwe, to martwe jest dowolne znakowanie $M'' \leq M$. Jeżeli zatem w grafie pokrycia istnieje węzeł końcowy M' , taki że $M \leq M'$, to znakowanie M musi być znakowaniem martwym. □

Dla grafów pokrycia zachowuje ważność twierdzenie 5.3 (jeżeli w założeniu twierdzenia przyjmiemy, że $\mathcal{G} = (V, A, \gamma)$ jest grafem pokrycia sieci \mathcal{N}).

5.3. Grafy silnie spójnych składowych

Graf pokrycia dla sieci nieograniczonych pozwala uzyskać skończoną reprezentację nieskończonego zbioru osiągalnych znakowań. W przypadku sieci, dla których niemożliwa byłaby analiza pełnego grafu osiągalności, uzyskuje się więc zredukowaną reprezentację zbioru $\mathcal{R}(M_0)$. Analiza pełnego grafu osiągalności może być niemożliwa również dla sieci ograniczonych, jeżeli graf ten jest zbyt duży ze względu na zasoby czasowe czy też sprzętowe, którymi dysponujemy. W najczęściej spotykanych metodach redukcji grafu osiągalności korzysta się z następującego algorytmu:

1. Określa się relację równoważności na zbiorze $\mathcal{R}(M_0)$.
2. Konstruuje się graf skierowany, którego węzły reprezentują klasy abstrakcji tej relacji.

Graf silnie spójnych składowych może być traktowany jako *skondensowana forma reprezentowania grafu osiągalności lub swego rodzaju nakładka*, która ułatwia analizę grafu osiągalności. Węzły takiego grafu reprezentują *silnie spójne składowe grafu osiągalności*.

Definicja 5.5. Graf $\mathcal{G}' = (V', A', \gamma')$ nazywamy *podgrafem* grafu skierowanego $\mathcal{G} = (V, A, \gamma)$, jeżeli są spełnione warunki:

- (1) $V' \subseteq V$;
- (2) $A' \subseteq \{a \in A : \gamma(a) \in V' \times V'\}$;
- (3) $\gamma' = \gamma|_{A'}$, tzn. γ' jest obcięciem funkcji zaczepienia γ do zbioru A' .

Jeżeli dodatkowo w warunku (2) występuje równość, to mówimy, że podgraf jest wyznaczony przez zbiór węzłów V' . ■

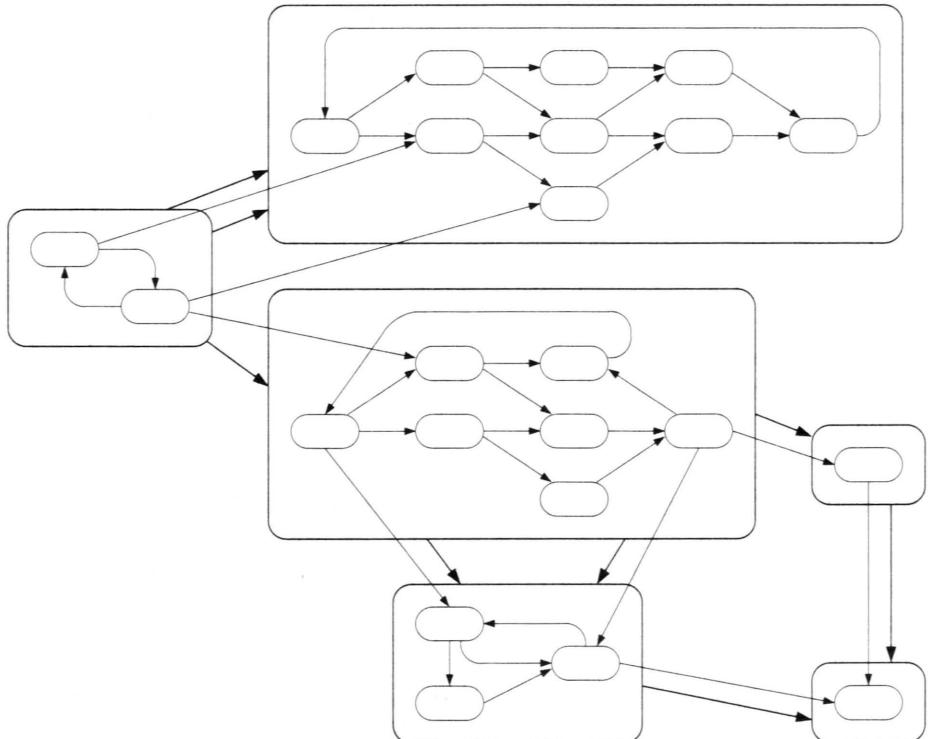
Definicja 5.6. *Silnie spójną składową* grafu skierowanego $\mathcal{G} = (V, A, \gamma)$ nazywamy podgraf \mathcal{G}' wyznaczony przez niepusty zbiór węzłów $V' \subseteq V$, taki że \mathcal{G}' jest grafem silnie spójnym i nie istnieje silnie spójny podgraf wyznaczony przez zbiór $V'' \subseteq V$, taki że $V' \subset V''$. ■

Silnie spójną składową, która zawiera tylko jeden węzeł, nazywamy *trywialną*. Zbiór wszystkich silnie spójnych składowych grafu \mathcal{G} oznaczamy symbolem \mathcal{G}_S . Zbiór \mathcal{G}_S wyznacza podział zbioru węzłów V grafu \mathcal{G} na rozłączne i niepuste podzbiory. Dla dowolnego węzła $x \in V$ symbol $\mathcal{G}_S(x)$ będzie oznaczał składową grafu \mathcal{G} , do której należy węzeł x .

Podział grafu skierowanego na silnie spójne składowe wyznacza relację równoważności określoną na zbiorze węzłów tego grafu. Dwa węzły $x, y \in V$ pozostają w relacji, jeżeli istnieje droga od x do y i jednocześnie istnieje droga od y do x .

Definicja 5.7. Niech $\mathcal{G} = (V, A, \gamma)$ będzie grafem. *Grafem silnie spójnych składowych* grafu \mathcal{G} nazywamy graf skierowany $\mathcal{G}' = (V', A', \gamma')$, taki że:

- (1) $V' = \mathcal{G}_S$;
 - (2) $A' \subseteq A$ jest zbiorem tych łuków, które łączą węzły należące do różnych składowych;
 - (3) $\forall a \in A': (\gamma(a) = (x, y)) \Rightarrow (\gamma'(a) = (\mathcal{G}_S(x), \mathcal{G}_S(y)))$.
-

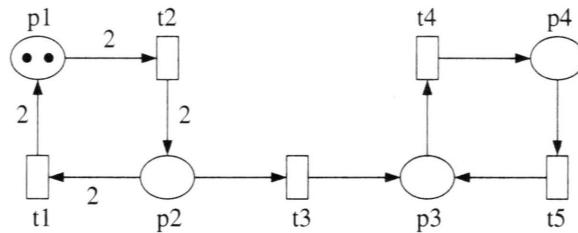


Rys. 5.9. Graf silnie spójnych składowych

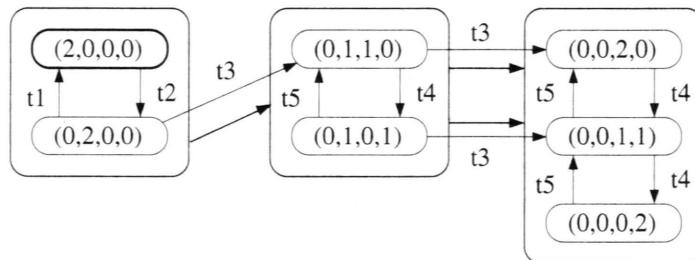
Ideę grafu silnie spójnych składowych przedstawiono na rys. 5.9². Węzeł grafu silnie spójnych składowych reprezentuje zbiór znakowań sieci i przejścia między tymi znakowaniami. Określenie, że znakowanie M należy do danego węzła grafu silnie spójnych składowych, oznacza, iż jest jednym ze znakowań, które reprezentuje ten węzeł. Podobnie określenie, że łuk grafu osiągalności należy do danego węzła grafu silnie spójnych składowych, oznacza, iż reprezentuje on przejście między dwoma znakowaniami należącymi do tego węzła.

² Rysunek na podstawie pracy [32].

Rozważmy przykład sieci uogólnionej przedstawionej na rys. 5.10. Graf osiągalności dla tej sieci wraz z nałożonym na nim grafem silnie spójnych składowych pokazano na rys. 5.11.



Rys. 5.10. Sieć uogólniona



Rys. 5.11. Graf osiągalności i graf silnie spójnych składowych

Dla grafu silnie spójnych składowych prawdziwe jest następujące twierdzenie:

Twierdzenie 5.8. Niech $\mathcal{G} = (V, A, \gamma)$ będzie grafem, a $\mathcal{G}' = (V', A', \gamma')$ – grafem silnie spójnych składowych grafu \mathcal{G} .

- (1) Graf \mathcal{G}' jest grafem acyklicznym.
- (2) Jeżeli zbiór V jest zbiorem skończonym, to zbiór V' również jest zbiorem skończonym.
- (3) Jeżeli zbiór V jest zbiorem skończonym, to dla dowolnego węzła $x \in V'$ istnieje węzeł końcowy $y \in V'$, taki że istnieje droga od x do y w grafie \mathcal{G}' .
- (4) Dla dowolnych węzłów $x, y \in V$ istnieje droga od x do y w grafie \mathcal{G} wtedy i tylko wtedy, gdy istnieje droga od $\mathcal{G}_S(x)$ do $\mathcal{G}_S(y)$ w grafie \mathcal{G}' (drogi te mogą mieć długość zero). ■

Dowód. Przypuśćmy, że graf \mathcal{G}' zawiera cykl. Cykl musi zawierać co najmniej jeden łuk, a ponieważ w grafie silnie spójnych składowych łuki łączą dwa różne węzły, więc najmniejszy możliwy cykl musiałby składać się z dwóch łuków łączących dwie różne silnie spójne składowe. Łatwo sprawdzić, że w takiej sytuacji węzły należące do tych dwóch składowych wyznaczają nową silnie spójną składową. Jest to sprzeczne z definicją grafu silnie spójnych składowych, w którym są rozważane maksymalne silnie spójne składowe, a zatem w grafie \mathcal{G}' nie może istnieć cykl.

Teza (2) wynika bezpośrednio z definicji grafu silnie spójnych składowych. Zbiór V' wyznacza podział zbioru V na niepuste i rozłączne podzbiory. Teza (3) jest konsekwencją pierwszych dwóch. Jeżeli graf \mathcal{G}' ma skońzoną liczbę węzłów i jest grafem acyklicznym, to każdą drogę w tym grafie da się przedłużyć, tak by prowadziła do węzła końcowego. Droga taka może mieć długość co najwyżej $|V'| - 1$.

Teza (4) wynika z definicji grafu silnie spójnych składowych. Przypuśćmy, że dla węzłów $x, y \in V$ istnieje droga a_1, a_2, \dots, a_k od węzła x do y . Jeżeli węzły te należą do jednej silnie spójnej składowej, to w grafie \mathcal{G}' węzły te są połączone drogą długości 0; jeżeli nie, to część z łuków a_1, a_2, \dots, a_k występuje w grafie \mathcal{G}' i wyznacza drogę od $\mathcal{G}_S(x)$ do $\mathcal{G}_S(y)$. Z drugiej strony przypuśćmy, że istnieje droga od węzła $\mathcal{G}_S(x)$ do $\mathcal{G}_S(y)$. Jeżeli $\mathcal{G}_S(x) = \mathcal{G}_S(y)$, to w grafie \mathcal{G} istnieje zarówno droga od węzła x do y , jak i od węzła y do x . Jeżeli $\mathcal{G}_S(x) \neq \mathcal{G}_S(y)$ oraz istnieje droga od węzła $\mathcal{G}_S(x)$ do $\mathcal{G}_S(y)$, to w grafie \mathcal{G} istnieje droga od węzła $x' \in \mathcal{G}_S(x)$ do węzła $y' \in \mathcal{G}_S(y)$. Ponieważ istnieje droga od węzła x do x' (należą do jednej silnie spójnej składowej) i droga od y' do y , to w grafie \mathcal{G} musi istnieć również droga od x do y . \square

Graf silnie spójnych składowych ogranicza możliwości analizy własności sieci w porównaniu z pełnym grafem osiągalności, niemniej jednak pozwala analizować pewne własności dynamiczne.

Twierdzenie 5.9. Niech $\mathcal{G} = (V, A, \gamma)$ będzie grafem silnie spójnych składowych grafu osiągalności sieci uogólnionej $\mathcal{N} = (P, T, A, W, M_0)$.

- (1) Niech $M_1, M_2 \in \mathcal{R}(M_0)$. Znakowanie $M_2 \in \mathcal{R}(M_1)$ wtedy i tylko wtedy, gdy w grafie \mathcal{G} istnieje droga od $\mathcal{G}_S(M_1)$ do $\mathcal{G}_S(M_2)$ (droga ta może mieć długość zero).
- (2) Istnieje znakowanie własne sieci \mathcal{N} wtedy i tylko wtedy, gdy graf \mathcal{G} zawiera dokładnie jeden węzeł końcowy. Znakowaniami własnymi sieci są znakowania ze zbioru reprezentowanego przez ten węzeł.
- (3) Sieć \mathcal{N} jest odwracalna wtedy i tylko wtedy, gdy graf \mathcal{G} zawiera dokładnie jeden węzeł.

- (4) Znakowanie $M \in \mathcal{R}(M_0)$ jest znakowaniem martwym sieci \mathcal{N} wtedy i tylko wtedy, gdy węzeł $\mathcal{G}_S(M)$ jest trywialnym węzłem końcowym grafu \mathcal{G} .
- (5) Przejście t sieci \mathcal{N} jest żywe wtedy i tylko wtedy, gdy każdy węzeł końcowy grafu \mathcal{G} zawiera łuk z etykietą t . ■

Twierdzenie powyższe wynika z twierdzenia 5.8 i definicji grafu silnie spójnych składowych.

Stosując graf silnie spójnych składowych do analizy sieci przedstawionej na rys. 5.10, można wysunąć następujące wnioski: sieć nie jest odwracalna, istnieją trzy znakowania własne $(0, 0, 2, 0)$, $(0, 0, 1, 1)$ i $(0, 0, 0, 2)$ oraz sieć nie jest żywa, chociaż ma dwa żywe przejścia $t4$ i $t5$.

Rozważmy przedstawione w tym rozdziale przykłady grafów osiągalności. Dla sieci modelującej działanie dystrybutora napojów (zob. rys. 4.2) graf osiągalności przedstawiono na rys. 5.1. Graf silnie spójnych składowych dla tego grafu zawiera wyłącznie trywialne składowe i jest identyczny (dokładniej mówiąc izomorficzny) z grafem osiągalności.

W przypadku sieci modelującej problem czytelników i pisarzy (zob. rys. 4.3 i rys. 5.4) oraz sieci modelującej system producent-konsument z ograniczonym magazynem (zob. rys. 4.5 i rys. 5.5) grafy silnie spójnych składowych zawierają dokładnie jeden węzeł, który reprezentuje wszystkie osiągalne znakowania tych sieci.

5.4. Symetrie

Zastosowanie pojęcia *symetrii* sieci Petriego jest jedną z kilku metod używanych do zredukowania rozmiaru grafu osiągalności (zob. prace: [61], [62], [63], [72]). Symetrie są funkcjami o ściśle określonych własnościach, których dziedziną i przeciwdziedziną jest zbiór węzłów sieci. Zdefiniowanie symetrii sieci wyznacza relację równoważności w zbiorze znakowań osiągalnych. Węzły grafu osiągalności uwzględniającego symetrie reprezentują nie pojedyncze znakowanie sieci, ale klasy abstrakcji takiej relacji.

Niech dana będzie sieć uogólniona $\mathcal{N} = (P, T, A, W, M_0)$.

Definicja 5.8. Symetrią sieci \mathcal{N} nazywamy bijekcję $f: V \rightarrow V$, ($V = P \cup T$) spełniającą warunki:

- (1) Funkcja f respektuje typ węzła sieci \mathcal{N} , tzn.:

$$f(P) = P \wedge f(T) = T. \quad (5.5)$$

(2) Funkcja f respektuje łuki sieci \mathcal{N} , tzn.:

$$\forall x, y \in V: (x, y) \in A \iff (f(x), f(y)) \in A. \quad (5.6)$$

(3) Funkcja f respektuje wagi łuków sieci \mathcal{N} , tzn.:

$$\forall (x, y) \in A: W(x, y) = W(f(x), f(y)). \quad (5.7)$$

■

Wszystkie symetrie sieci \mathcal{N} tworzą grupę z operacją składania funkcji. Zbiór wszystkich symetrii sieci \mathcal{N} oznaczamy symbolem \mathcal{S} . Stosując symetrie sieci uogólnionej do analizy ich własności, bierzemy pod uwagę wszystkie symetrie sieci albo podgrupę grupy \mathcal{S} . Elementem neutralnym operacji składania funkcji jest *funkcja tożsamościowa* (f_{id}) i tworzy ona trywialną podgrupę grupy symetrii sieci \mathcal{N} . Użycie wyłącznie funkcji tożsamościowej nie powoduje redukcji rozmiaru grafu osiągalności. Najprostsze podgrupy warte rozważenia składają się z: funkcji tożsamościowej f_{id} , pewnej symetrii $f \neq f_{id}$ i symetrii f^{-1} (funkcji odwrotnej do funkcji f), przy czym może się zdarzyć, że $f = f^{-1}$.

Niech $\mathcal{S}_{\mathcal{N}}$ będzie rozważaną podgrupą symetrii sieci \mathcal{N} . Podgrupa $\mathcal{S}_{\mathcal{N}}$ wyznacza relację równoważności $\sim_{\mathcal{S}}$ określona na zbiorze węzłów sieci:

$$\forall x, y \in V: x \sim_{\mathcal{S}} y \iff \exists f \in \mathcal{S}_{\mathcal{N}}: y = f(x). \quad (5.8)$$

Definicja 5.9. Węzły $x, y \in V$ nazywamy *równoważnymi*, jeżeli $x \sim_{\mathcal{S}} y$. Węzeł $x \in V$ nazywamy *węzłem stałym* ze względu na podgrupę symetrii $\mathcal{S}_{\mathcal{N}}$, jeżeli $[x]_{\mathcal{S}} = \{x\}$, przy czym $[x]_{\mathcal{S}}$ oznacza klasę abstrakcji elementu x w relacji $\sim_{\mathcal{S}}$. ■

Symetrie sieci można rozszerzyć na znakowania osiągalne ze znakowania początkowego. Niech M będzie znakowaniem sieci \mathcal{N} ; znakowanie $f(M)$ jest znakowaniem spełniającym warunek:

$$\forall p \in P: f(M)(f(p)) = M(p) \quad (5.9)$$

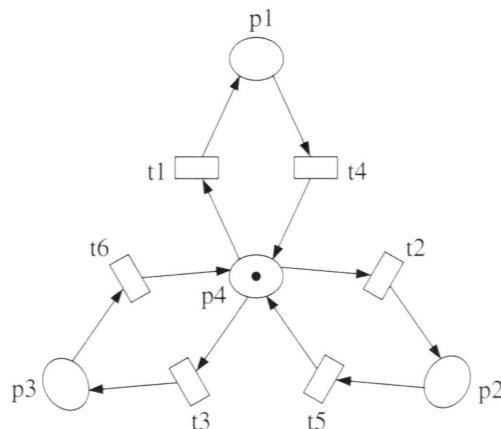
$$\text{tzn. } f(M)(p) = M(f^{-1}(p)).$$

Relację równoważności dla znakowań sieci definiujemy następująco (do oznaczenia tej relacji również użyto symbolu $\sim_{\mathcal{S}}$):

$$\forall M_1, M_2 \in \mathcal{R}(M_0): M_1 \sim_{\mathcal{S}} M_2 \iff \exists f \in \mathcal{S}_{\mathcal{N}}: M_2 = f(M_1). \quad (5.10)$$

Definicja 5.10. Znakowania $M_1, M_2 \in \mathcal{R}(M_0)$ nazywamy *równoważnymi*, jeżeli $M_1 \sim_S M_2$. Znakowanie $M \in \mathcal{R}(M_0)$ nazywamy *symetrycznym* ze względu na podgrupę symetrii \mathcal{S}_N , jeżeli $[M]_S = \{M\}$. ■

Zanalizujmy sieć N przedstawioną na rys. 5.12. Sieć składa się z trzech symetrycznych części, na których podstawie można wskazać symetrie sieci oraz równoważne jej węzły. Wszystkie symetrie przedstawiono w tabeli 5.1. Między symetriami sieci N zachodzą następujące zależności: $f_1 = f_{id}$, $f_2^{-1} = f_2$, $f_3^{-1} = f_3$, $f_4^{-1} = f_4$, $f_5^{-1} = f_6$ i $f_6^{-1} = f_5$.



Rys. 5.12. Sieć uogólniona

Tabela 5.1. Symetrie sieci N

	$p1$	$p2$	$p3$	$p4$	$t1$	$t2$	$t3$	$t4$	$t5$	$t6$
f_1	$p1$	$p2$	$p3$	$p4$	$t1$	$t2$	$t3$	$t4$	$t5$	$t6$
f_2	$p2$	$p1$	$p3$	$p4$	$t2$	$t1$	$t3$	$t5$	$t4$	$t6$
f_3	$p3$	$p2$	$p1$	$p4$	$t3$	$t2$	$t1$	$t6$	$t5$	$t4$
f_4	$p1$	$p3$	$p2$	$p4$	$t1$	$t3$	$t2$	$t4$	$t6$	$t5$
f_5	$p2$	$p3$	$p1$	$p4$	$t2$	$t3$	$t1$	$t5$	$t6$	$t4$
f_6	$p3$	$p1$	$p2$	$p4$	$t3$	$t1$	$t2$	$t6$	$t4$	$t5$

W tabeli 5.2 przedstawiono wyniki składania wszystkich symetrii z tabeli 5.1. Przy składaniu symetrii jako pierwsze będzie wykonywane przekształcenie podane w pierwszym wierszu tej tabeli, a potem przekształcenie podane w pierwszej jej

Tabela 5.2. Złożenia symetrii sieci \mathcal{N}

\circ	f_1	f_2	f_3	f_4	f_5	f_6
f_1	f_1	f_2	f_3	f_4	f_5	f_6
f_2	f_2	f_1	f_6	f_5	f_4	f_3
f_3	f_3	f_5	f_1	f_6	f_2	f_4
f_4	f_4	f_6	f_5	f_1	f_3	f_2
f_5	f_5	f_3	f_4	f_2	f_6	f_1
f_6	f_6	f_4	f_2	f_3	f_1	f_5

kolumnie. Na podstawie tabeli 5.2 można wyznaczyć wszystkie podgrupy symetrii dla rozważanej sieci: $\{f_1, f_2, \dots, f_6\}$, $\{f_1\}$, $\{f_1, f_2\}$, $\{f_1, f_3\}$, $\{f_1, f_4\}$ i $\{f_1, f_5, f_6\}$.

Rozważmy zbiór $\mathcal{S}_{\mathcal{N}}$ uwzględniający wszystkie symetrie sieci \mathcal{N} . Relacja $\sim_{\mathcal{S}}$ dzieli zbiór węzłów sieci na następujące klasy abstrakcji: $[p1] = \{p1, p2, p3\}$, $[p4] = \{p4\}$, $[t1] = \{t1, t2, t3\}$ i $[t4] = \{t4, t5, t6\}$.

Węzeł $p4$ jest węzłem stałym ze względu na grupę symetrii $\mathcal{S}_{\mathcal{N}}$. Z definicji 5.8 wynika, że dwa węzły mogą być równoważne tylko wówczas, gdy oba węzły mają tyle samo łuków wejściowych i wyjściowych (a ponadto o odpowiednich parami równych wagach). Klasa abstrakcji węzła $p4$ jest jednoelementowa, gdyż nie istnieje w sieci inne miejsce, które ma trzy przejścia wejściowe i trzy wyjściowe.

Stosując symetrie sieci, możemy wyznaczyć równoważne znakowania. Dla wszystkich symetrii spełniony jest warunek $f_i(0, 0, 0, 1) = (0, 0, 0, 1)$ (dla $i = 1, 2, \dots, 6$). Znakowanie początkowe jest zatem znakowaniem symetrycznym. Pozostałe znakowania sieci tworzą jedną klasę abstrakcji.

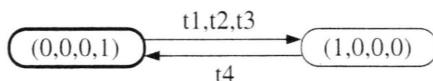
Oprócz znakowania początkowego, w zbiorze $\mathcal{R}(M_0)$ znajdują się trzy znakowania: $M_1 = (1, 0, 0, 0)$, $M_2 = (0, 1, 0, 0)$ i $M_3 = (0, 0, 1, 0)$. Rozważmy znakowania M_1 i M_2 .

$$f_2(M_1)(p1) = M_1(f_2^{-1}(p1)) = M_1(f_2(p1)) = M_1(p2) = 0. \quad (5.11)$$

Kontynuując te rozważania dla kolejnych miejsc sieci, otrzymujemy zależność $f_2(M_1) = M_2$, a zatem $M_1 \sim_{\mathcal{S}} M_2$. Podobnie można pokazać, że znakowanie M_3 jest równoważne znakowaniom M_1 i M_2 .

Jeżeli zbiór symetrii sieci \mathcal{N} zawiera więcej niż jedną symetrię, to możemy zbudować graf osiągalności, którego węzły reprezentują nie pojedyncze znakowania sieci, ale klasy abstrakcji relacji $\sim_{\mathcal{S}}$. Graf taki nazywamy *grafem osiągalności zredukowanym ze względu na symetrie* i konstruujemy go w sposób następujący:

1. Jako węzeł początkowy grafu przyjmujemy znakowanie początkowe.
2. Dla każdego węzła dodanego do grafu osiągalności wyznaczamy wszystkie aktywne przejścia i znakowania będące wynikiem wykonania tych przejść. Niech M_i będzie rozważanym węzłem, a M_j takim znakowaniem, że $M_i \xrightarrow{t} M_j$.
 - Jeżeli w grafie osiągalności istnieje węzeł M_k , taki że $M_j \sim_S M_k$, to do grafu dodajemy łuk łączący węzeł M_i z M_k i nadajemy mu etykietę t .
 - Jeżeli w grafie nie istnieje węzeł równoważny znakowaniu M_j , to do grafu dodajemy węzeł M_j i łuk łączący węzeł M_i z M_j (z etykietą t).



Rys. 5.13. Graf osiągalności zredukowany ze względu na symetrię

Dla sieci występującej na rys. 5.12 przedstawiono na rysunku 5.13 graf osiągalności uwzględniający symetrie. Upraszczając rysunek, zamiast wielokrotnych łuków użyto pojedynczych, ale z wielokrotnymi etykietami.

Zastosowanie symetrii sieci pozwala zredukować rozmiar grafu osiągalności. Pomiędzy zredukowaną postacią grafu, możliwa jest analiza jego niektórych własności dynamicznych. Dla zredukowanego grafu osiągalności jest prawdziwe następujące twierdzenie, które wynika bezpośrednio z algorytmu konstruowania tego grafu.

Twierdzenie 5.10. Niech $\mathcal{G} = (V, A, \gamma)$ będzie grafem osiągalności sieci \mathcal{N} zredukowanym ze względu na symetrię.

- (1) Dla dowolnego znakowania $M \in \mathcal{R}(M_0)$ istnieje znakowanie $M' \in V$, takie że $M \sim_S M'$.
- (2) Jeżeli znakowanie początkowe M_0 jest symetryczne, to $M \in \mathcal{R}(M_0)$ wtedy i tylko wtedy, gdy istnieje węzeł $M' \in V$, taki że $M \sim_S M'$.
- (3) W zbiorze $\mathcal{R}(M_0)$ nie ma znakowania martwego wtedy i tylko wtedy, gdy żaden węzeł grafu \mathcal{G} nie jest węzłem końcowym. ■

Graf zredukowany zawiera zazwyczaj mniej łuków niż pełny graf osiągalności, dzięki czemu analiza żywotności poszczególnych przejść sieci jest ograniczona. Symbolem $T_{\mathcal{G}}(M)$ oznaczamy zbiór takich przejść t sieci \mathcal{N} , dla których istnieje symetria $f \in S_{\mathcal{N}}$ i węzeł M' w grafie \mathcal{G} , taki że istnieje w grafie droga od M do M' i przejście $f(t)$ jest aktywne przy znakowaniu M' . Tak więc $T_{\mathcal{G}}(M)$ jest zbiorem tych przejść, o których na podstawie zredukowanego grafu można stwierdzić, że są potencjalnie wykonalne dla znakowania M .

Twierdzenie 5.11. Niech $\mathcal{G} = (V, A, \gamma)$ będzie grafem osiągalności sieci N zredukowanym ze względu na symetrie. Jeżeli znakowanie początkowe M_0 jest znakowaniem symetrycznym, to przejście $t \in T$ jest martwe wtedy i tylko wtedy, gdy dla dowolnego znakowania $M \in V$, $t \notin T_{\mathcal{G}}(M)$. ■

Twierdzenie 5.12. Niech $\mathcal{G} = (V, A, \gamma)$ będzie grafem osiągalności sieci N zredukowanym ze względu na symetrie. Jeżeli przejście $t \in T$ jest żywe, to dla dowolnego znakowania $M \in V$, $t \in T_{\mathcal{G}}(M)$. ■

Twierdzenie odwrotne do twierdzenia 5.12 nie jest prawdziwe. Wynika stąd, że możliwości analizy sieci w przypadku grafów zredukowanych ze względu na symetrie są ograniczone w porównaniu z pełnymi grafami osiągalności. Dokładna analiza żywotności pojedynczego przejścia t jest możliwa po zbudowaniu zredukowanego grafu uwzględniającego symetrie należące do podgrupy symetrii, dla których węzeł t jest węzłem stałym. W najgorszym przypadku jedyną taką podgrupą może się okazać podgrupa $\{f_{id}\}$ i wówczas otrzymujemy pełny graf osiągalności. Formalne dowody powyższych twierdzeń można znaleźć w pracy [72].

Przedstawiony na rys. 5.13 zredukowany graf osiągalności nie zawiera łuków o etykietach $t5$ i $t6$, mimo że sieć z rys. 5.12 jest żywa. Oznaczmy drugi z węzłów grafu przez M_1 , tzn. $M_1 = (1, 0, 0, 0)$. Zbiór $T_{\mathcal{G}}(M_1)$ zawiera wszystkie przejścia sieci. Przejścia $t1, t2, t3$ i $t4$ występują jawnie jako etykiety łuków należących do dróg prowadzących od znakowania M_1 . Przejście $t5$ należy do tego zbioru, gdyż przejście $t4$ jest aktywne przy znakowaniu M_1 ($M_1 \sim_S M_1$) oraz istnieje droga o długości 0 prowadząca od M_1 do M_1) i $f_2(t5) = t4$. Analogicznie można uzasadnić, że zbiór $T_{\mathcal{G}}(M_1)$ zawiera również przejście $t6$. Zbiór $T_{\mathcal{G}}(M_0)$ również zawiera wszystkie przejścia sieci.

Rozważmy sieć modelującą problem 5 filozofów, przedstawioną na rys. 3.16. Można dla tej sieci wskazać pięć symetrii. Oprócz funkcji tożsamościowej pozostałe symetrie mogą być potraktowane jako *obroty* odpowiednio o $\frac{1}{5}, \frac{2}{5}, \frac{3}{5}$ i $\frac{4}{5}$ kąta pełnego. Obcięcie tych symetrii do zbioru miejsc sieci przedstawiono w tabeli 5.3.

Rozważmy zbiór S_N uwzględniający wszystkie symetrie sieci N . Relacja \sim_S dzieli zbiór węzłów sieci na następujące klasy abstrakcji:

$$[p1] = \{p1, p2, p3, p4, p5\},$$

$$[p6] = \{p6, p7, p8, p9, p10\},$$

$$[p11] = \{p11, p12, p13, p14, p15\},$$

$$[p16] = \{p16, p17, p18, p19, p20\},$$

$$[t1] = \{t1, t2, t3, t4, t5\},$$

$$[t6] = \{t6, t7, t8, t9, t10\},$$

$$[t11] = \{t11, t12, t13, t14, t15\}.$$

Tabela 5.3. Symetrie sieci N modelującej problem pięciu filozofów (obcięcie do zbioru miejsc sieci)

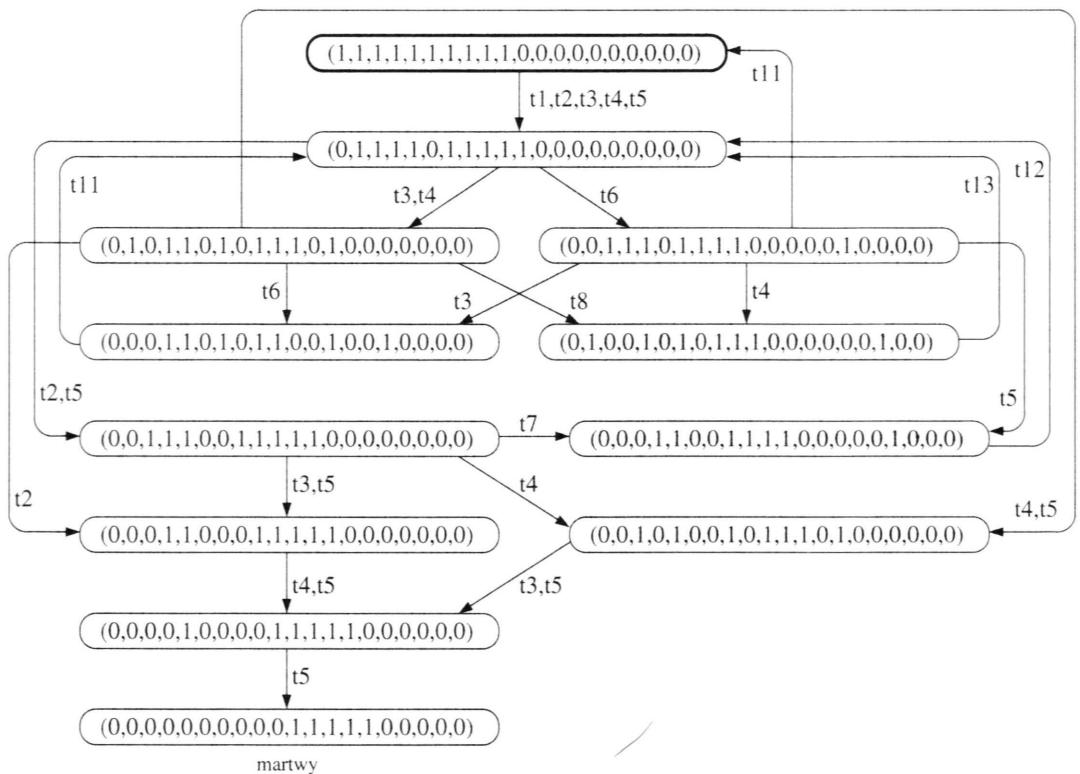
	$p1$	$p2$	$p3$	$p4$	$p5$	$p6$	$p7$	$p8$	$p9$	$p10$
f_1	$p1$	$p2$	$p3$	$p4$	$p5$	$p6$	$p7$	$p8$	$p9$	$p10$
f_2	$p2$	$p3$	$p4$	$p5$	$p1$	$p7$	$p8$	$p9$	$p10$	$p6$
f_3	$p3$	$p4$	$p5$	$p1$	$p2$	$p8$	$p9$	$p10$	$p6$	$p7$
f_4	$p4$	$p5$	$p1$	$p2$	$p3$	$p9$	$p10$	$p6$	$p7$	$p8$
f_5	$p5$	$p1$	$p2$	$p3$	$p4$	$p10$	$p6$	$p7$	$p8$	$p9$
	$p11$	$p12$	$p13$	$p14$	$p15$	$p16$	$p17$	$p18$	$p19$	$p20$
f_1	$p11$	$p12$	$p13$	$p14$	$p15$	$p16$	$p17$	$p18$	$p19$	$p20$
f_2	$p12$	$p13$	$p14$	$p15$	$p11$	$p17$	$p18$	$p19$	$p20$	$p16$
f_3	$p13$	$p14$	$p15$	$p11$	$p12$	$p18$	$p19$	$p20$	$p16$	$p17$
f_4	$p14$	$p15$	$p11$	$p12$	$p13$	$p19$	$p20$	$p16$	$p17$	$p18$
f_5	$p15$	$p11$	$p12$	$p13$	$p14$	$p20$	$p16$	$p17$	$p18$	$p19$

Zbiór znakowań $\mathcal{R}(M_0)$ jest podzielony na 18 klas abstrakcji. Znakowanie początkowe jest znakowaniem symetrycznym. Również klasa abstrakcji znakowania martwego $M = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0)$ zawiera dokładnie jedno znakowanie. Pozostałe klasy abstrakcji są pięcioelementowe. Przykład jednej z klas abstrakcji przedstawiono w tabeli 5.4.

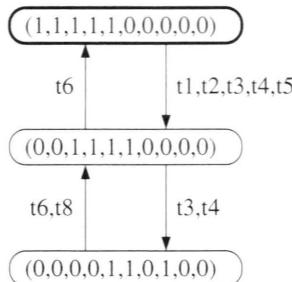
Tabela 5.4. Przykład klasy abstrakcji relacji \sim_S

M	(0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
$f_1(M)$	(0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
$f_2(M)$	(1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0)
$f_3(M)$	(1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0)
$f_4(M)$	(1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0)
$f_5(M)$	(1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0)

Fragment grafu osiągalności sieci \mathcal{N} zredukowanego ze względu na symetrie przedstawiono na rys. 5.14. Zawiera on między innymi węzeł końcowy reprezentujący martwe znakowanie sieci. (Na rysunku pominięto łuki wejściowe prowadzące od niewidocznych węzłów).



abstrakcji zawiera tylko znakowanie początkowe, które jest znakowaniem symetrycznym. Pozostałe klasy zawierają po pięć znakowań osiągalnych ze znakowania początkowego.



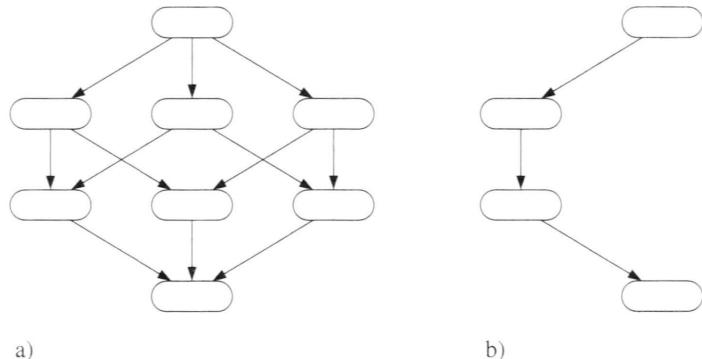
Rys. 5.15. Graf osiągalności zredukowany ze względu na symetrie dla sieci modelującej problem pięciu filozofów (wersja 4)

5.5. Trwałe zbiory przejść

Wyznaczanie *trwałych zbiorów przejść* jest jedną z metod stosowanych do redukcji rozmiaru grafu osiągalności (zob. prace: [20], [84], [85], [86], [87]). Zbiory te są podzbiorami przejść sieci zależnymi od znakowania i przy generowaniu zredukowanego grafu osiągalności konieczne jest wyznaczanie trwałych zbiorów przejść dla każdego osiągalnego znakowania. Pełny graf osiągalności zawiera drogi przedstawiające wszystkie możliwe sposoby przeplatania się wykonywania przejść z takiego zbioru. Graf osiągalności zredukowany ze względu na trwałe zbiory przejść zawiera tylko jedną z możliwych dróg. W efekcie następuje redukcja zarówno liczby węzłów grafu (pewne pośrednie znakowania nie są w zredukowanym grafie przedstawiane) jak i liczby łuków grafu. Podstawową ideę metody redukcji grafu osiągalności ze względu na trwałe zbiory przejść przedstawiono na rys. 5.16.

Powiemy nieformalnie, że dany zbiór przejść sieci jest trwały, jeżeli spełnia następujące warunki:

- zawiera co najmniej jedno aktywne przejście;
- żadne aktywne przejście z tego zbioru nie może utracić aktywności w wyniku wykonania przejść nienależących do tego zbioru;
- żadne nieaktywne przejście z tego zbioru nie może uzyskać aktywności w wyniku wykonania wyłącznie przejść nienależących do tego zbioru.



Rys. 5.16. Fragment grafu osiągalności przed (a) i po (b) zastosowaniu redukcji ze względu na trwałe zbiory przejść

Niech $\mathcal{N} = (P, T, A, W, M_0)$ będzie siecią uogólnioną i niech będą dane: znakowanie $M \in \mathcal{R}(M_0)$, przejście $t \in T$ i miejsce $p \in P$. Dla sieci \mathcal{N} zdefiniujemy kilka funkcji przydatnych przy wprowadzaniu pojęcia trwałości zbioru przejść.

$$f_1: \mathcal{R}(M_0) \times P \rightarrow 2^T \wedge \\ f_1(M, p) = \{t' \in In(p): M(p) \geq W(p, t') \wedge W(t', p) > W(p, t')\}, \quad (5.12)$$

$$f_2: \mathcal{R}(M_0) \times T \times P \rightarrow 2^T \wedge \\ f_2(M, t, p) = f_4(p) \cup \{t' \in Out(p): W(p, t) > W(t, p) \wedge \\ W(p, t') > M(p) - W(p, t) + W(t, p)\}, \quad (5.13)$$

$$f_3: \mathcal{R}(M_0) \times T \times P \rightarrow 2^T \wedge \\ f_3(M, t, p) = f_1(M, p) \cup \{t' \in In(p): M(p) \geq W(p, t') \wedge \\ W(t', p) > W(t, p)\}, \quad (5.14)$$

$$f_4: P \rightarrow 2^T \wedge f_4(p) = \{t' \in Out(p): W(p, t') > W(t', p)\}. \quad (5.15)$$

- Funkcja f_1 przypisuje parze (M, p) zbiór przejść, których wykonanie zwiększa liczbę znaczników w tym miejscu i których ewentualna nieaktywność przy znakowaniu M nie jest spowodowana brakiem odpowiedniej liczby znaczników w miejscu p .
- Wartością funkcji f_2 jest zbiór przejść, których wykonanie zmniejsza liczbę znaczników w miejscu p lub które stają się nieaktywne w wyniku wykonania przejścia t przy znakowaniu M .

- Funkcja f_3 przypisuje trójce (M, t, p) zbiór przejść, których wykonanie zwiększa liczbę znaczników w miejscu p lub które dodają do tego miejsca więcej znaczników niż przejście t i dla których miejsce p zawiera odpowiednią liczbę znaczników do ich wykonania.
- Funkcja f_4 przypisuje miejscu p zbiór przejść, których wykonanie zmniejsza liczbę znaczników w tym miejscu.

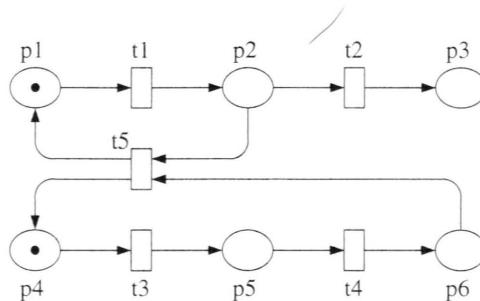
W każdym przypadku przy wyznaczaniu wartości funkcji f_i ($i = 1, 2, 3, 4$), jeżeli łuk $(x, y) \notin A$, to przyjmujemy $W(x, y) = 0$.

Definicja 5.11. Przejście $t \in T$ nazywamy *przejściem kluczowym* zbioru $T' \subseteq T$ przy znakowaniu M , jeżeli: $t \in T'$, t jest aktywne przy znakowaniu M oraz

$$\forall p \in In(t): f_4(p) \subseteq T'. \quad (5.16)$$

Zbiór $T' \subseteq T$ nazywamy *trwałym* przy znakowaniu M , jeżeli istnieje przejście będące przejściem kluczowym zbioru T' przy znakowaniu M oraz każde przejście $t \in T'$ spełnia warunek:

$$\begin{aligned} & (\exists p \in In(t): M(p) < W(p, t) \wedge f_1(M, p) \subseteq T') \vee \\ & (t \text{ jest } M\text{-aktywne} \wedge (\forall p \in In(t): W(p, t) \leq W(t, p) \vee \\ & f_2(M, t, p) \subseteq T' \vee f_3(M, t, p) \subseteq T')). \end{aligned} \quad \blacksquare \quad (5.17)$$



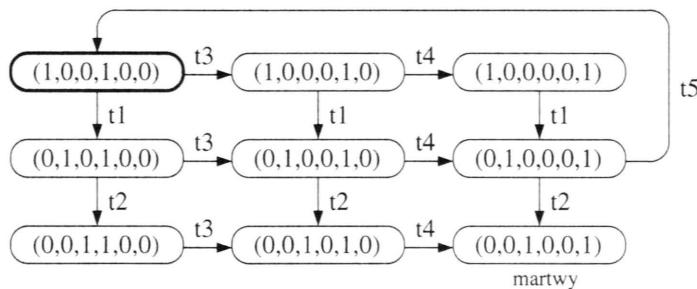
Rys. 5.17. Przykład sieci uogólnionej

Rozważmy przykład sieci przedstawionej na rys. 5.17. Przy znakowaniu początkowym istnieją dwa trwałe zbiory przejść $\{t1\}$ i $\{t3\}$. Dla zbioru $\{t1\}$ mamy: $In(t1) = \{p1\}$, $f_4(p1) = \{t1\} \subseteq \{t1\}$, ponadto przejście $t1$ jest aktywne przy znakowaniu M_0 , jest zatem przejściem kluczowym zbioru $\{t1\}$. Ponieważ $f_2(M_0, t1, p1) = \{t1\}$, a zatem spełniony jest warunek (5.17), więc zbiór $\{t1\}$ jest trwałym. Podobnie można pokazać, że trwały jest zbiór $\{t3\}$.

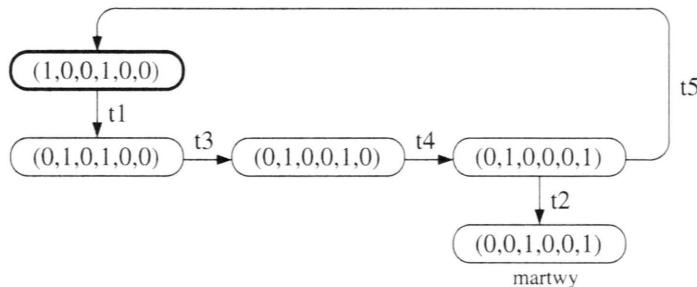
Określenie *trwały zbiór przejść* spowodowane jest faktem, że zbiór ten pozostaje trwały, jeżeli wykonamy przejście nienależące do niego. Stosując definicję trwałości zbioru przejść, można wykazać prawdziwość następującego wniosku.

Wniosek 5.2. Niech będą dane: trwały zbiór przejść $T' \subseteq T$, znakowanie M sieci N , przejście $t \in T'$ i ciąg przejść $\alpha = t_1, t_2, \dots, t_k$, przy czym $t_1, t_2, \dots, t_k \notin T'$, takie że $M \xrightarrow{\alpha} M_1 \xrightarrow{t} M_2$. Istnieje znakowanie $M'_1 \in \mathcal{R}(M)$, takie że $M \xrightarrow{t} M'_1 \xrightarrow{\alpha} M_2$. ■

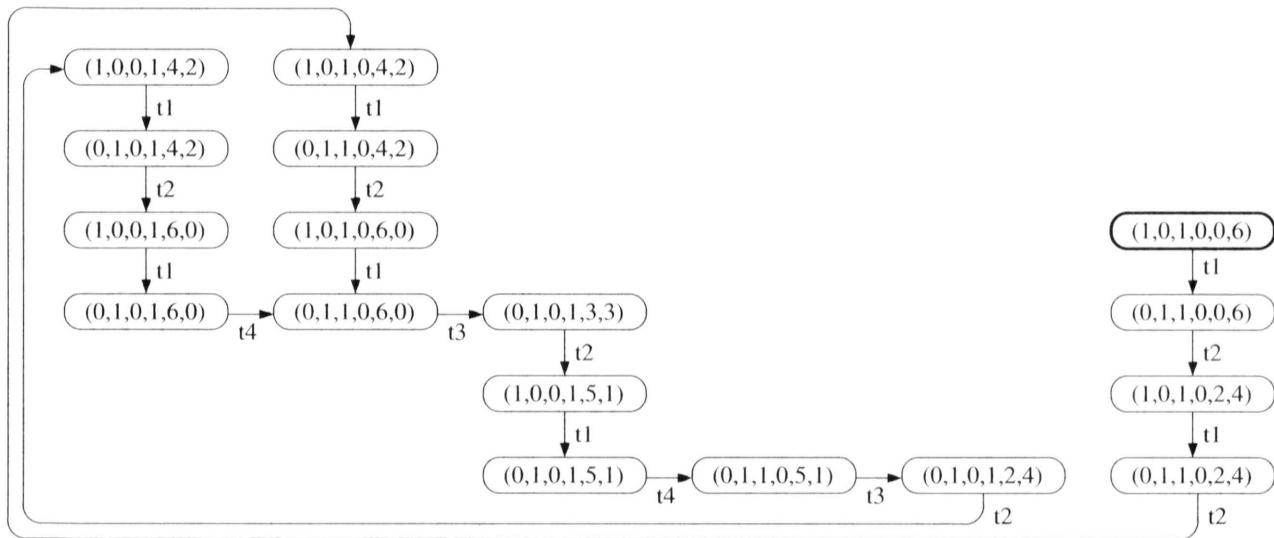
Redukcja grafu osiągalności ze względu na trwałe zbiory przejść polega na podziale zbioru przejść na dwa podzbiory, dokonywanym przy każdym osiągalnym znakowaniu sieci. Jeden z tych podzbiorów jest trwałym zbiorem przejść, a drugi jego dopełnieniem. Przy wyznaczaniu kolejnych węzłów grafu rozważa się wyłącznie aktywne przejścia należące do trwałego zbioru przejść. Jeżeli przy pewnym znakowaniu M istnieje więcej niż jeden trwały zbiór przejść, to wybieramy niedeterministycznie jeden z tych zbiorów i rozważamy tylko aktywne przejścia należące do tego zbioru.



Rys. 5.18. Graf osiągalności dla sieci z rys. 5.17



Rys. 5.19. Graf osiągalności dla sieci z rys. 5.17, zredukowany ze względu na trwałe zbiory przejść



Rys. 5.20. Graf osiągalności dla sieci modelującej system producent-konsument, zredukowany ze względu na trwałe zbiory przejść

Efektywność stosowanej metody często zależy od sposobu rozstrzygania niedeterministycznego wyboru. Jedną z proponowanych strategii jest wybór zbioru o jak najmniejszej liczbie aktywnych przejść. W tym celu wyznacza się tzw. *minimalne trwałe zbiory przejść* i wybiera się tylko jeden z nich.

Na rysunku 5.18 przedstawiono pełny graf osiągalności dla sieci ukazanej na rys. 5.17. Graf osiągalności zredukowany ze względu na trwałe zbiory przejść tworzymy następująco. Jako pierwszy węzeł grafu przyjmujemy znakowanie początkowe, a jako trwały zbiór przejść wybierzemy zbiór $\{t1\}$. Wynikiem wykonania przejęcia $t1$ jest znakowanie $M_1 = (0, 1, 0, 1, 0, 0)$. Przy znakowaniu M_1 istnieje dokładnie jeden minimalny trwały zbiór przejść $\{t3\}$. Wynikiem wykonania przejęcia $t3$ jest znakowanie $M_2 = (0, 1, 0, 0, 1, 0)$, a następnie wynikiem wykonania przejęcia $t4$ – znakowanie $M_3 = (0, 1, 0, 0, 0, 1)$ (zbiór $\{t4\}$) jest minimalnym trwałym zbiorem przejść przy znakowaniu M_2). Minimalnym trwałym zbiorem przejść przy znakowaniu M_3 jest zbiór $\{t2, t5\}$. Oba przejścia są aktywne, a zatem należy rozważyć wyniki wykonania obu tych przejść. Na rysunku 5.19 przedstawiono zredukowany graf osiągalności.

Możliwości analizy dynamicznych właściwości sieci są znacznie ograniczone po zastosowaniu redukcji grafu osiągalności ze względu na trwałe zbiory przejść. Jedną z ważnych cech tej metody redukcji jest fakt, że zredukowany graf osiągalności zawiera wszystkie martwe znakowania sieci. Ponadto, jeżeli w pełnym grafie osiągalności istnieje droga nieskończona, to w grafie zredukowanym również istnieje taka droga. Redukcję grafu osiągalności można wykonać, stosując jednocześnie symetrie sieci i redukcję ze względu na trwałe zbiory przejść.

Na rysunku 5.20 przedstawiono graf osiągalności sieci modelującej system producent-konsument (wersja z ograniczonym magazynem, zob. rys. 4.5) zredukowany ze względu na trwałe zbiory przejść. Sposób przedstawienia grafu umożliwia jego łatwe porównanie z pełnym grafem osiągalności (zob. rys. 5.5).

5.6. Algebraiczna reprezentacja sieci

Strukturę dowolnej sieci uogólnionej można przedstawić za pomocą dwóch macierzy o współczynnikach całkowitych. *Macierz wejść* opisuje łuki łączące przejścia z miejscami sieci, a współczynniki tej macierzy reprezentują liczby znaczników dodawanych do miejsc sieci przez jej przejścia. Z drugiej strony, *macierz wyjść* opisuje łuki łączące miejsca z przejściami, a współczynniki tej macierzy reprezentują liczby znaczników usuwane z miejsc sieci.

Niech $\mathcal{N} = (P, T, A, W, M_0)$ będzie siecią uogólnioną i niech zbiory P i T będą zbiorami uporządkowanymi, $P = \{p_1, p_2, \dots, p_n\}$ i $T = \{t_1, t_2, \dots, t_m\}$.

Definicja 5.12. Macierzą wejścia sieci \mathcal{N} nazywamy macierz $N^+ = (\alpha_{ij})_{n \times m}$, której współczynniki α_{ij} są zdefiniowane następująco:

$$\alpha_{ij} = \begin{cases} W(t_j, p_i), & \text{jeżeli } t_j \in In(p_i) \\ 0 & \text{w pozostałych przypadkach.} \end{cases} \quad (5.18)$$

Macierzą wyjścia sieci \mathcal{N} nazywamy macierz $N^- = (\alpha_{ij})_{n \times m}$, której współczynniki α_{ij} są zdefiniowane następująco:

$$\alpha_{ij} = \begin{cases} W(p_i, t_j), & \text{jeżeli } t_j \in Out(p_i) \\ 0 & \text{w pozostałych przypadkach.} \end{cases} \quad (5.19)$$

Przedstawienie struktury sieci za pomocą macierzy wejścia i wyjścia jest wzajemnie jednoznaczne, tzn. dla danej sieci macierze te wyznaczone są jednoznacznie i na odwrót, mając dane macierze wejścia i wyjścia, możemy jednoznacznie odtworzyć strukturę sieci. Macierze wejścia i wyjścia w połączeniu ze znakowaniem początkowym stanowią pełną reprezentację sieci uogólnionych. Reprezentacja ta jest bardzo wygodną formą opisu sieci w wypadku ich analizy za pomocą oprogramowania komputerowego.

Na podstawie macierzy wyjścia możemy łatwo sprawdzić, które przejścia są aktywne przy danym znakowaniu sieci. Niech t_i^- oznacza wektor stanowiący i -tą kolumnę macierzy N^- . (Analogicznie t_i^+ oznacza wektor stanowiący i -tą kolumnę macierzy N^+).

Wniosek 5.3. Niech M będzie znakowaniem sieci \mathcal{N} . Przejście $t_i \in T$ jest aktywne przy znakowaniu M wtedy i tylko wtedy, gdy

$$t_i^- \leq M. \quad (5.20)$$

Przejście t_i jest więc aktywne przy znakowaniu M , jeżeli w wyniku odejmowania $M - t_i^-$ otrzymujemy wektor o współczynnikach nieujemnych. Jeżeli przejście t_i jest aktywne przy znakowaniu M , to nowe znakowanie M' uzyskane w wyniku wykonania przejścia t_i można wyznaczyć następująco:

$$M' = M - t_i^- + t_i^+. \quad (5.21)$$

Definicja 5.13. Macierzą incydencji sieci \mathcal{N} nazywamy macierz $N = (\alpha_{ij})_{n \times m}$, taką że:

$$N = N^+ - N^- \quad (5.22)$$

Współczynnik α_{ij} macierzy incydencji opisuje zmianę znakowania miejsca p_i w wyniku wykonania przejścia t_j . Przedstawienie sieci za pomocą jej macierzy incydencji nosi nazwę *liniowo algebraicznej reprezentacji* sieci. Przyporządkowanie sieci jej macierzy incydencji jest wzajemnie jednoznaczne tylko dla sieci czystych (tzn. sieci bez ciasnych pętli). Jeżeli sieć zawiera ciasne pętle, to na podstawie macierzy incydencji nie można w jednoznaczny sposób odtworzyć struktury sieci.

5.7. Niezmienności miejsc

Niezmienniki miejsc wyrażają pewne stałe właściwości znakowań osiąganych w danej sieci. Charakteryzują one zbiory miejsc sieci, w których łączna (ewentualnie ważona) liczba znaczników nie ulega zmianie. Niezmienności miejsc stosowane są między innymi do badania ograniczości miejsc oraz zachowawczości sieci.

Niech $\mathcal{N} = (P, T, A, W, M_0)$ będzie siecią uogólnioną i niech zbiory P i T będą zbiorami uporządkowanymi, $P = \{p_1, p_2, \dots, p_n\}$ i $T = \{t_1, t_2, \dots, t_m\}$.

Niech $P' \subseteq P$ będzie zbiorem miejsc sieci \mathcal{N} . Traktując zbiór P jako uniwersum, zbiór P' można opisać, stosując wektor charakterystyczny tego zbioru, który będziemy oznaczać symbolem $C_{P'}$ ³.

Przypuśćmy, że łączna liczba znaczników w miejscach ze zbioru P' pozostaje stała podczas pracy sieci. Dla zbioru P' i przejścia $t_i \in T$ spełniony jest więc następujący warunek:

$$(t_i^+ - t_i^-) \circ C_{P'} = 0, \quad (5.23)$$

przy czym symbol \circ oznacza iloczyn skalarny wektorów.

Warunek ten zachodzi dla dowolnego przejścia $t_i \in T$, musi zatem być spełniony układ równań:

$$\left\{ \begin{array}{l} (t_1^+ - t_1^-) \circ C_{P'} = 0 \\ (t_2^+ - t_2^-) \circ C_{P'} = 0 \\ \dots \\ (t_m^+ - t_m^-) \circ C_{P'} = 0 \end{array} \right. \quad (5.24)$$

³ Wektor charakterystyczny jest wektorem o $|P|$ współrzędnych, w którym i -ta współrzędna ma wartość 1, jeżeli $p_i \in P'$, oraz wartość 0 w przeciwnym przypadku.

Różnice $(t_i^+ - t_i^-)$ są w rzeczywistości kolumnami macierzy incydencji, zatem powyższy układ równań można zapisać macierzowo:

$$N^T C_{P'} = 0, \quad (5.25)$$

przy czym macierz N^T jest macierzą transponowaną do macierzy incydencji N .

Podsumowując, jeśli łączna liczba znaczników w miejscach ze zbioru P' nie zmienia się podczas pracy sieci, to jest spełniony układ równań (5.25). Z drugiej strony każde rozwiązanie tego układu będące wektorem charakterystycznym wyznacza pewien zbiór miejsc o powyżej wymienionej własności.

Definicja 5.14. Niech I będzie wektorem o współrzędnych całkowitych, których liczba jest równa liczbie miejsc sieci \mathcal{N} . Wektor I nazywamy *niezmiennikiem miejsc* (P -niezmiennikiem) sieci \mathcal{N} , jeżeli:

$$N^T \cdot I = 0. \quad (5.26)$$

■

Twierdzenie 5.13. Dla dowolnego niezmiennika miejsc I sieci \mathcal{N} i dowolnego znakowania $M \in \mathcal{R}(M_0)$ spełniony jest warunek:

$$M \circ I = M_0 \circ I. \quad (5.27)$$

■

Dowód. Niech $M \in \mathcal{R}(M_0)$ i niech $M_0 \xrightarrow{\alpha} M$, dla ciągu przejść $\alpha = t_{i_1}, t_{i_2}, \dots, t_{i_k}$. Stosując wzór (5.21), zależność między znakowaniami M_0 i M można przedstawić następująco:

$$M = M_0 + \sum_{j=1}^k (t_{i_j}^+ - t_{i_j}^-). \quad (5.28)$$

Ponieważ I jest niezmiennikiem miejsc, zatem dla dowolnego przejścia t_{i_j} , przy czym $j = 1, 2, \dots, k$, jest spełniony warunek:

$$(t_{i_j}^+ - t_{i_j}^-) \circ I = 0. \quad (5.29)$$

Stąd wynika, że

$$\begin{aligned} M \circ I &= (M_0 + \sum_{j=1}^k (t_{i_j}^+ - t_{i_j}^-)) \circ I = \\ M_0 \circ I + \sum_{j=1}^k &(t_{i_j}^+ - t_{i_j}^-) \circ I = M_0 \circ I. \end{aligned} \quad (5.30)$$

□

Twierdzenie odwrotne do powyższego jest prawdziwe tylko wtedy, gdy każde przejście można przynajmniej raz wykonać. W szczególności jest ono prawdziwe dla sieci żywych.

Twierdzenie 5.14. Niech \mathcal{N} będzie żywą siecią uogólnioną i niech I będzie wektorem o $|P|$ współrzędnych całkowitych, takim że dla dowolnego znakowania $M \in \mathcal{R}(M_0)$ jest spełniony warunek (5.27). Wówczas I jest niezmiennikiem miejsc sieci \mathcal{N} . ■

Dowód. Wystarczy pokazać, że dla każdego przejścia t_i jest spełniony warunek (5.29). Niech $t_i \in T$. Sieć jest żywa, więc istnieje znakowanie $M \in \mathcal{R}(M_0)$, takie że przejście t_i jest aktywne przy znakowaniu M . Przyjmijmy, że $M' = M + t_i^+ - t_i^-$. Zgodnie z założeniem są spełnione równości:

$$M \circ I = M' \circ I = (M + (t_i^+ - t_i^-)) \circ I \quad (5.31)$$

Wynika stąd, że $(t_i^+ - t_i^-) \circ I = 0$, a zatem dla dowolnego przejścia jest spełniony warunek (5.29), czyli I jest niezmiennikiem miejsc sieci \mathcal{N} . □

Z powyższych twierdzeń wynika bezpośrednio następujący wniosek:

Wniosek 5.4. Niech M_1 i M_2 będą znakowaniami sieci \mathcal{N} . Jeżeli istnieje niezmiennik miejsc I , taki że $M_1 \circ I \neq M_2 \circ I$, to $M_2 \notin \mathcal{R}(M_1)$. ■

Z definicji 5.14 wynika, że niezmiennik miejsc nie musi być wektorem charakterystycznym. Jeśli wszystkie współrzędne niezmiennika I są liczbami nieujemnymi, to niezmiennik nazywamy *nieujemnym*. Podobnie jak w przypadku znakowań sieci, można porównywać niezmienniki miejsc, stosując relację mniejszości. W sensie tej relacji można mówić o *minimalnych nieujemnych niezmiennikach miejsc*.

Wyznaczenie niezmienników miejsc sieci wymaga rozwiązyania układu równań liniowych. Do tego celu można zastosować następujący algorytm, bazujący na przekształcaniu macierzy uzupełnionej układu równań.

Niech będzie dany układ m równań liniowych o n niewiadomych:

$$Ax = b, \quad (5.32)$$

przy czym

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}. \quad (5.33)$$

Punktem wyjścia do rozwiązywania układu (5.32) jest stworzenie *macierzy uzupełnionej* układu równań liniowych:

$$U = \left(\begin{array}{cc} A & b \end{array} \right) = \left(\begin{array}{ccccc} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} & b_m \end{array} \right). \quad (5.34)$$

Na macierzach tych można wykonywać następujące operacje elementarne:

- (1) Do dowolnego wiersza macierzy U można dodać kombinację liniową innych wierszy tej macierzy.
- (2) Dowolny wiersz macierzy U można pomnożyć przez liczbę rzeczywistą różną od zera.
- (3) Można zmienić kolejność dowolnych dwóch wierszy macierzy U .
- (4) Można zmienić kolejność dowolnych dwóch kolumn macierzy A .

Należy zwrócić uwagę, że przy zmianie kolejności dwóch kolumn macierzy A , zmienia się kolejność niewiadomych w wektorze x (i -ta kolumna macierzy A odpowiada i -tej współrzędnej wektora x).

Stosując powyższe przekształcenia, można sprowadzić macierz U do jednej z *postaci bazowych*:

$$\left(\begin{array}{cc} E & c_1 \end{array} \right), \quad (5.35)$$

$$\left(\begin{array}{ccc} E & M & c_1 \end{array} \right), \quad (5.36)$$

$$\left(\begin{array}{cc} E & c_1 \\ 0 & c_2 \end{array} \right), \quad (5.37)$$

$$\left(\begin{array}{ccc} E & M & c_1 \\ 0 & c_2 \end{array} \right), \quad (5.38)$$

przy czym:

- E jest macierzą jednostkową o wymiarze $k \times k$, (k jest rzędem macierzy A);
- M jest dowolną macierzą o wymiarach $k \times (n - k)$;
- c_1 jest wektorem o k współrzędnych;
- c_2 jest wektorem o $m - k$ współrzędnych;
- 0 jest macierzą zerową o wymiarach $(m - k) \times n$.

Jeżeli macierz U uda się sprowadzić do postaci (5.35) lub (5.36), to układ równań liniowych (5.32) jest zgodny. W przypadku sprowadzenia macierzy U do postaci

(5.37) lub (5.38) układ jest zgodny wtedy i tylko wtedy, gdy $c_2 = 0$ (tzn., gdy c_2 jest wektorem zerowym).

W przypadku (5.35) i (5.37), jeżeli układ (5.32) jest zgodny, to jego rozwiązanie ogólne ma postać:

$$x = c_1. \quad (5.39)$$

W przypadku (5.36) i (5.38), jeżeli układ (5.32) jest zgodny, to jego rozwiązanie ogólne ma postać:

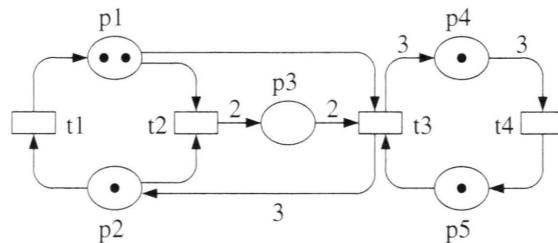
$$x = \begin{pmatrix} c_1 - Ma \\ a \end{pmatrix}, \quad (5.40)$$

przy czym a jest wektorem dowolnych parametrów.

Wśród rozwiązań układu można wyróżnić tzw. *rozwiązania bazowe*, które wyznaczamy w następujący sposób:

- (1) Tworzymy k -elementowe kombinacje z n -elementowego zbioru niewiadomych $\{x_1, x_2, \dots, x_n\}$ (k jest rzędem macierzy A). Niewiadome znajdujące się w kombinacji nazywamy *zmiennymi bazowymi*, a pozostałe – *zmiennymi niebazowymi*.
- (2) Jeżeli danej kombinacji możemy przypisać jednoznacznie takie rozwiązanie szczegółowe, że wszystkie zmienne niebazowe są równe zero, to takie rozwiązanie szczegółowe nazywamy *rozwiązaniem bazowym*.

Wyznaczanie niezmienników sieci sprowadza się do rozwiązywania układów jednorodnych (wektor b jest zawsze wektorem zerowym), a zatem rozwiązanie układu zawsze istnieje. Jeżeli jedynym rozwiązaniem jest wektor zerowy, to przyjmujemy, że nie istnieją niezmienniki danej sieci.



Rys. 5.21. Przykład sieci uogólnionej pokrytej niezmiennikami miejsc

Rozważmy przykład sieci uogólnionej N przedstawionej na rys. 5.21. Macierz incydencji tej sieci jest następująca:

$$N = \begin{pmatrix} 1 & -1 & -1 & 0 \\ -1 & -1 & 3 & 0 \\ 0 & 2 & -2 & 0 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & -1 & 1 \end{pmatrix}. \quad (5.41)$$

Aby wyznaczyć niezmienniki sieci N , należy rozwiązać układ równań (5.32). Macierz uzupełniona układu równań jest postaci:

$$U = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 & 0 \\ -1 & 3 & -2 & 3 & -1 & 0 \\ 0 & 0 & 0 & -3 & 1 & 0 \end{pmatrix}. \quad (5.42)$$

Przekształcanie macierzy U rozpoczniemy od dodania wiersza pierwszego do wierszy: drugiego i trzeciego. Po tych przekształceniach otrzymujemy macierz postaci:

$$U = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & -2 & 2 & 0 & 0 & 0 \\ 0 & 2 & -2 & 3 & -1 & 0 \\ 0 & 0 & 0 & -3 & 1 & 0 \end{pmatrix}. \quad (5.43)$$

Na otrzymanej macierzy wykonamy następujące przekształcenia: wiersz drugi dodamy do wiersza trzeciego, sumę wierszy drugiego i trzeciego dodamy do wiersza czwartego oraz wiersz drugi pomnożymy przez $-\frac{1}{2}$. W wyniku tych przekształceń otrzymujemy macierz:

$$U = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (5.44)$$

W ostatnim już kroku, wiersz drugi dodamy do wiersza pierwszego, wiersz trzeci pomnożymy przez -1 i zamienimy miejscami kolumny trzecią z piątą (to przestawienie wpływa na zmienne x_3 i x_5). Oto ostateczna postać macierzy U :

$$U = \begin{pmatrix} 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & -3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (5.45)$$

Macierz U sprowadziliśmy zatem do postaci bazowej (5.38). Poszczególne podmacierze są następujące:

$$E = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad M = \begin{pmatrix} 0 & -1 \\ 0 & -1 \\ -3 & 0 \end{pmatrix}, \quad (5.46)$$

$$c_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad c_2 = \begin{pmatrix} 0 \end{pmatrix}.$$

Wektor c_2 jest wektorem zerowym, zatem układ równań jest oznaczony. Rozwiązańe ogólne otrzymujemy na podstawie wzoru (5.40). Macierz M ma dwie kolumny, a więc wektor parametrów będzie miał dwie współrzędne:

$$a = \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix}. \quad (5.47)$$

Ostatecznie rozwiązanie ogólne układu (po zmianie kolejności trzeciej i piątej współrzędnej ze względu na wcześniejszą zmianę kolejności kolumn) jest postaci:

$$x = \begin{pmatrix} \alpha_2 \\ \alpha_2 \\ \alpha_2 \\ \alpha_1 \\ 3\alpha_1 \end{pmatrix}. \quad (5.48)$$

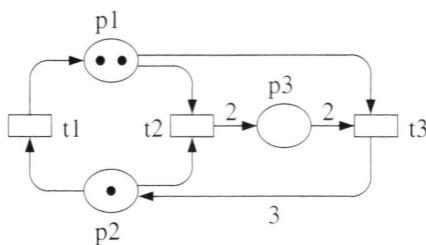
Przyjmując wartości parametrów $\alpha_1 = 1$ i $\alpha_2 = 0$ oraz $\alpha_1 = 0$ i $\alpha_2 = 1$, otrzymujemy dwa rozwiązania bazowe⁴:

$$I_1 = (1, 1, 1, 0, 0), \quad I_2 = (0, 0, 0, 1, 3). \quad (5.49)$$

Twierdzenie 5.15. Niech I_1 i I_2 będą niezmiennikami miejsc sieci N i niech n będzie liczbą całkowitą. Wówczas $I_1 + I_2$ oraz $n \cdot I_1$ również są niezmiennikami miejsc sieci N . ■

Zbiór tych miejsc sieci N , którym odpowiadają niezerowe współrzędne niezmiennika I nazywamy *nośnikiem niezmiennika I* , a podsieć wyznaczoną przez ten zbiór, *sieciową reprezentacją niezmiennika I* . Dla miejsc podsieci zachowujemy takie znakowanie początkowe, jakie miały one w sieci N . Sieciową reprezentację niezmiennika $I_1 = (1, 1, 1, 0, 0)$ przedstawiono na rys. 5.22.

⁴ Rozwiązanie ogólne układu równań zapisano jako wektor kolumnowy, ale niezmienniki, które są rozwiązaniami szczegółowymi, będziemy zapisywać jako wektory wierszowe.



Rys. 5.22. Sieciowa reprezentacja niezmiennika miejsc $I_1 = (1, 1, 1, 0, 0)$

Niezmienniki miejsc rozważanej sieci mają następującą interpretację. Niezmiennik $I_1 = (1, 1, 1, 0, 0)$ jest wektorem charakterystycznym, a jego nośnikiem jest zbiór miejsc $\{p1, p2, p3\}$. Sieciowa reprezentacja niezmiennika I_1 jest siecią zachowawczą. Nośnikiem drugiego niezmiennika jest zbiór $\{p4, p5\}$. Niezmiennik I_2 nie jest wektorem charakterystycznym, ale jego niezerowe współrzędne wyznaczają wagę dla miejsc należących do jego nośnika. Sieciowa reprezentacja niezmiennika I_2 jest zachowawcza względem wektora wag $(1, 3)$.

Wniosek 5.5. Niech I będzie nieujemnym niezmiennikiem miejsc sieci \mathcal{N} oraz niech P' będzie nośnikiem niezmiennika I . Jeżeli $M_0 \circ I = n$, to każde miejsce ze zbioru P' jest n -ograniczone. ■

Innym przykładem niezmiennika miejsc dla sieci przedstawionej na rys. 5.21 jest wektor $I_3 = I_1 + I_2 = (1, 1, 1, 1, 3)$. Niezmiennik ten jest jednocześnie wektorem wag, względem którego sieć jest zachowawcza. Dla rozważanej sieci ważona suma znaczników wynosi 7 i nie zmienia się podczas pracy sieci.

Definicja 5.15. Sieć \mathcal{N} nazywamy siecią pokrytą niezmiennikami miejsc, jeżeli dla każdego miejsca $p_i \in P$ istnieje nieujemny niezmiennik miejsc I , którego i -ta współrzędna jest większa niż zero. ■

Wniosek 5.6. Jeżeli sieć \mathcal{N} jest pokryta niezmiennikami miejsc, to istnieje niezmiennik miejsc I tej sieci, którego wszystkie współrzędne są większe niż zero. ■

Poszukiwanym niezmiennikiem miejsc może być suma niezmienników pokrywających sieć \mathcal{N} . W przypadku sieci zamieszczonej na rys. 5.21 przykładem niezmiennika pokrywającego sieć jest niezmiennik I_3 .

Twierdzenie 5.16. Jeżeli sieć \mathcal{N} jest pokryta niezmiennikami miejsc, to jest ograniczona. ■

Dowód. Jeżeli sieć jest pokryta niezmiennikami miejsc, to na podstawie twierdzenia 5.15 można wskazać niezmiennik pokrywający sieć. Niezmiennik ten jest wektorem wag, względem którego sieć jest zachowawcza. Ważona suma znaczników jest liczbą skończoną. Liczba ta jest górnym ograniczeniem wszystkich miejsc sieci. \square

Rozważmy przykłady sieci przedstawione w podrozdziale 4.2. Dla sieci modelującej działanie dystrybutora napojów (zob. rys. 4.2) macierz uzupełnioną układu równań (5.26) można przekształcić do postaci bazowej (5.37):

$$U = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (5.50)$$

Rozwiązaniem ogólnym ($x = c_1$) jest wektor zerowy, a zatem rozważana sieć nie ma niezmienników miejsc.

Macierz uogólnioną układu równań (5.26) dla sieci modelującej wzajemne wykluczanie dla problemu czytelników i pisarzy (zob. rys. 4.3) można przekształcić do postaci bazowej (5.38):

$$U = \begin{pmatrix} 1 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & -4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad (5.51)$$

przy czym, ze względu na zmiany kolejności kolumn, kolejność zmiennych jest następująca: $x_1, x_2, x_4, x_6, x_3, x_5, x_7$. Rozwiązanie ogólne (po uporządkowaniu zmiennych) jest postaci:

$$x = \begin{pmatrix} \alpha_1 - \alpha_3 \\ \alpha_1 - \alpha_3 \\ \alpha_1 \\ \alpha_2 \\ \alpha_2 \\ \alpha_2 + 4\alpha_3 \\ \alpha_3 \end{pmatrix}. \quad (5.52)$$

Przykładami niezmienników miejsc są wektory $I_1 = (1, 1, 1, 0, 0, 0, 0)$, $I_2 = (0, 0, 1, 0, 0, 4, 1)$ oraz $I_3 = (0, 0, 0, 1, 1, 1, 0)$. Sieć jest pokryta niezmiennikami miejsc, a zatem jest również ograniczona. Przykładem wektora pokrywającego sieć N jest wektor $I_4 = I_1 + I_2 + I_3 = (1, 1, 2, 1, 1, 5, 1)$. Sieć jest zatem zachowawcza względem wektora wag I_4 . Na podstawie niezmiennika I_1 możemy wnioskować, że miejsca p_1 , p_2 i p_3 są 4-ograniczone. Podobnie analizując niezmienniki I_2 i I_3 , wnioskujemy, że miejsce p_7 jest 4-ograniczone, a miejsca p_4 , p_5 i p_6 są bezpieczne.

W odniesieniu do modelowanego systemu znaczenie niezmienników I_1 , I_2 i I_3 jest następujące. Niezmiennik I_1 wyznacza podsieć, w obrębie której są *przemieszczane* znaczniki reprezentujące czytelników, a ich liczba się nie zmienia. Analogiczne znaczenie ma niezmiennik I_3 , ale w odniesieniu do pisarzy. Niezmiennik I_2 dotyczy dostępnych miejsc w czytelni; ważona suma wolnych miejsc i procesów znajdujących się w czytelni jest stała (pisarz zajmuje 4 miejsca).

Rozważmy teraz dwa modele systemu producent-konsument (zob. rys. 4.4 i rys. 4.5). Dla sieci przedstawionej na rys. 4.4 macierz uogólnioną układu równań (5.26) można przekształcić do postaci bazowej (5.38):

$$U = \begin{pmatrix} 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad (5.53)$$

przy czym kolumny drugą i piątą zamieniono miejscami. Rozwiążanie ogólne (po uporządkowaniu zmiennych) jest postaci:

$$x = \begin{pmatrix} \alpha_2 \\ \alpha_2 \\ \alpha_1 \\ \alpha_1 \\ 0 \end{pmatrix}. \quad (5.54)$$

Przykładami niezmienników są wektory $I_1 = (1, 1, 0, 0, 0)$ i $I_2 = (0, 0, 1, 1, 0)$. Sieć nie jest pokryta niezmiennikami miejsc, gdyż nie istnieje nieujemny niezmiennik, którego piąta współrzędna byłaby dodatnia (miejsce to oznacza magazyn i jest nieograniczone). Sieć jest częściowo zachowawcza dla zbioru miejsc będącego nośnikiem niezmiennika $I_3 = I_1 + I_2 = (1, 1, 1, 1, 0)$.

Dla modelu przedstawionego na rys. 4.5 macierz uogólnioną układu równań (5.26) można przekształcić do postaci bazowej (5.38):

$$U = \begin{pmatrix} 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad (5.55)$$

przy czym kolumny drugą i piątą zamieniono miejscami. Rozwiążanie ogólne (po uporządkowaniu zmiennych) jest postaci:

$$x = \begin{pmatrix} \alpha_2 \\ \alpha_2 \\ \alpha_1 \\ \alpha_1 \\ \alpha_3 \\ \alpha_3 \end{pmatrix}. \quad (5.56)$$

Przykładami niezmienników miejsc są wektory $I_1 = (1, 1, 0, 0, 0, 0)$, $I_2 = (0, 0, 1, 1, 0, 0)$ oraz $I_3 = (0, 0, 0, 0, 1, 1)$. Sieć jest pokryta niezmiennikami miejsc. Niezmiennikiem pokrywającym sieć jest wektor $I_4 = I_1 + I_2 + I_3 = (1, 1, 1, 1, 1, 1)$, a zatem sieć jest zachowawcza. Na podstawie niezmienników I_1 , I_2 i I_3 wnioskujemy, że miejsca $p1$, $p2$, $p3$ i $p4$ są bezpieczne, a miejsca $p5$ i $p6$ są 6-ograniczone.

5.8. Niezmienniki przejść

Rozważmy sieć uogólnioną $\mathcal{N} = (P, T, A, W, M_0)$. Niech t_{i_1} będzie dowolnym przejściem sieci \mathcal{N} . Jeżeli przejście t_{i_1} jest aktywne przy znakowaniu M_0 i jego wykonanie prowadzi do znakowania M_1 , to prawdziwa jest zależność:

$$M_1 = M_0 + t_{i_1}^+ - t_{i_1}^-. \quad (5.57)$$

Niech C_{i_1} będzie wektorem charakterystycznym jednoelementowego zbioru przejść $\{t_{i_1}\}$. Stosując macierz incydencji sieci, zależność (5.57) możemy zapisać następująco:

$$M_1 = M_0 + N \cdot C_{i_1}. \quad (5.58)$$

Podobne rozumowanie możemy przeprowadzić dla dwóch przejść t_{i_1} i t_{i_2} . Jeżeli $M_0 \xrightarrow{t_{i_1}} M_1$ i $M_1 \xrightarrow{t_{i_2}} M_2$, to prawdziwa jest zależność:

$$M_2 = M_0 + N \cdot C_{i_1} + N \cdot C_{i_2} = M_0 + N \cdot (C_{i_1} + C_{i_2}). \quad (5.59)$$

Uogólniając powyższą zależność na dowolną liczbę przejść, otrzymujemy:

$$M_k = M_0 + N \cdot \sum_{j=1}^k C_{i_j}. \quad (5.60)$$

Powyższe rozumowanie można zapisać w postaci następującego twierdzenia.

Twierdzenie 5.17. Niech α będzie ciągiem przejść prowadzącym od znakowania M do znakowania M_k , przy czym $M, M_k \in \mathcal{R}(M_0)$. Niech I będzie wektorem o $|T|$ współrzędnych, takim że i -ta współrzędna jest równa liczbie wystąpień przejścia t_i w ciągu α . Dla znakowań M_k i M jest spełniona zależność:

$$M_k = M + N \cdot I. \quad (5.61)$$

■

Twierdzenie odwrotne w przypadku ogólnym nie jest prawdziwe.

Interesujące są zwłaszcza te ciągi przejść, które prowadzą od znakowania M ponownie do niego samego. Taki ciąg przejść spełnia warunek:

$$M = M + N \cdot I. \quad (5.62)$$

Wynika stąd, że:

$$N \cdot I = 0. \quad (5.63)$$

Definicja 5.16. Niech I będzie wektorem o współrzędnych całkowitych, których liczba jest równa liczbie przejść sieci N . Wektor I nazywamy *niezmiennikiem przejść* (*T-niezmiennikiem*) sieci N , jeżeli wektor I spełnia zależność (5.63). ■

Zbiór przejść, które odpowiadają niezerowym współrzędnym niezmiennika przejść, nazywamy *nośnikiem niezmiennika przejść*. Podobnie jak w wypadku niezmienników miejsc, mówimy o *nieujemnych niezmiennikach przejść*, *minimalnych niezmiennikach przejść* oraz o *sieciowej reprezentacji niezmiennika przejść*.

Dla sieci przedstawionej na rys. 5.21 macierz uzupełnioną układu (5.63) można sprowadzić do postaci bazowej (5.38):

$$U = \begin{pmatrix} 1 & 0 & 0 & -2 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (5.64)$$

Rozwiązanie ogólne jest postaci:

$$x = \begin{pmatrix} 2\alpha \\ \alpha \\ \alpha \\ \alpha \end{pmatrix}. \quad (5.65)$$

Przykładem niezmiennika przejść jest wektor $I = (2, 1, 1, 1)$, a jego reprezentacją sieciową jest cała sieć.

Definicja 5.17. Niech I będzie niezmiennikiem przejść sieci \mathcal{N} . Niezmiennik I nazywamy *realizowalnym*, jeżeli istnieje znakowanie $M \in \mathcal{R}(M_0)$ i ciąg przejść $\alpha \in \mathcal{L}(M)$, taki że każde przejście sieci występuje w nim dokładnie tyle razy, ile wynosi wartość odpowiadającej mu współrzędnej wektora I . ■

Warto podkreślić, że nie każdy nieujemny niezmiennik przejść jest realizowalny. Niezmiennik przejść $I = (2, 1, 1, 1)$ sieci na rys. 5.21 jest realizowalny, a przykładem ciągu przejść, który realizuje ten niezmiennik, jest ciąg $\alpha = t2, t3, t4, t1, t1$. Nie jest to jedyny taki ciąg – trzy ostatnie przejścia można wykonać w dowolnej kolejności.

Twierdzenie 5.18. Niech I_1 i I_2 będą niezmiennikami przejść sieci uogólnionej \mathcal{N} i niech n będzie liczbą całkowitą. Wówczas $I_1 + I_2$ oraz $n \cdot I_1$ są również niezmiennikami przejść sieci \mathcal{N} . ■

Definicja 5.18. Sieć \mathcal{N} nazywamy *siecią pokrytą niezmiennikami przejść*, jeżeli dla każdego przejścia $t_i \in T$ istnieje nieujemny niezmiennik przejść I , którego i -ta współrzędna jest większa niż zero. ■

Wniosek 5.7. Jeżeli sieć \mathcal{N} jest pokryta niezmiennikami przejść, to istnieje niezmiennik przejść I tej sieci, którego wszystkie współrzędne są większe niż zero. ■

Sieć na rysunku 5.21 jest pokryta niezmiennikami przejść, a przykładem niezmiennika pokrywającego sieć jest wektor $I = (2, 1, 1, 1)$.

Twierdzenie 5.19. Jeżeli sieć \mathcal{N} jest żywa i ograniczona, to jest ona pokryta niezmiennikami przejść. ■

Dowód. Przyjmijmy, że $T = \{t_1, t_2, \dots, t_m\}$. Sieć \mathcal{N} jest ograniczona, zatem liczba znakowań osiągalnych ze znakowania początkowego jest skończona. Niech $k = |\mathcal{R}(M_0)|$, przy czym $k \in \mathbb{N}$. Sieć jest żywa, a więc dla dowolnego znakowania $M \in \mathcal{R}(M_0)$ istnieje znakowanie $M' \in \mathcal{R}(M_0)$, takie że $M \xrightarrow{\alpha} M'$ i każde przejście sieci \mathcal{N} występuje co najmniej jeden raz w ciągu α .

Utwórzmy pewien ciąg osiągalnych znakowań w następujący sposób:

- $M_1 = M_0$;
- dla znakowania M_1 wybieramy znakowanie $M'_1 \in \mathcal{R}(M_0)$, takie że $M_1 \xrightarrow{\alpha_1} M'_1$ i każde przejście sieci \mathcal{N} występuje co najmniej jeden raz w ciągu α_1 . Znakowanie M'_1 oznaczamy przez M_2 .

Jeżeli będziemy dalej tworzyć ten ciąg znakowań w podobny sposób, to po co najwyżej k krokach pewien element tego ciągu się powtórzy. Dla ustalenia uwagi, niech $M_i = M_j$. Zgodnie ze sposobem tworzenia tego ciągu, aby ze znakowania M_i otrzymać znakowanie M_j , każde przejście sieci musi zostać co najmniej jeden raz wykonane. Niech α oznacza ciąg przejść prowadzący od znakowania M_i do M_j . Weźmy pod uwagę wektor I , którego i -ta współrzędna jest równa liczbie wystąpień przejścia t_i w ciągu α . Tak zdefiniowany wektor I ma wszystkie współrzędne dodatnie i spełnia warunek: $M_j = M_i + N \cdot I$. Ponieważ $M_i = M_j$, więc wektor I spełnia równanie (5.63), a zatem jest niezmienikiem przejść pokrywającym sieć \mathcal{N} . □

Rozważmy przykłady sieci przedstawione w podrozdziale 4.2. Dla sieci modelującej działanie dystrybutora napojów (zob. rys. 4.2) macierz uzupełnioną układu równań (5.63) można przekształcić do postaci bazowej (5.36):

$$U = \begin{pmatrix} 1 & 0 & 0 & 0 & -3 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 4 & -1 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (5.66)$$

Rozwiązaniem ogólnym ($x = c_1$) jest wektor zerowy, a zatem rozważana sieć nie ma niezmienników przejść.

Macierz uogólnioną układu równań (5.63) dla sieci modelującej wzajemne wykluczanie dla problemu czytelników i pisarzy (zob. rys. 4.3) można przekształcić do postaci bazowej (5.38):

$$U = \begin{pmatrix} 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad (5.67)$$

przy czym ze względu na zmiany kolejności kolumn, kolejność zmiennych jest następująca: $x_1, x_2, x_4, x_5, x_3, x_6$. Rozwiązanie ogólne (po uporządkowaniu zmiennych) jest postaci:

$$x = \begin{pmatrix} \alpha_1 \\ \alpha_1 \\ \alpha_1 \\ \alpha_2 \\ \alpha_2 \\ \alpha_2 \end{pmatrix}. \quad (5.68)$$

Przykładami niezmienników przejść są wektory $I_1 = (1, 1, 1, 0, 0, 0)$ oraz $I_2 = (0, 0, 0, 1, 1, 1)$. Oba podane niezmienniki są realizowalne. Właściwe ciągi przejść prowadzące od znakowania M_0 ponownie do tego znakowania, to odpowiednio: $t1, t2, t3$ i $t4, t5, t6$. Sieć jest pokryta niezmiennikami przejść. W odniesieniu do modelowanego systemu niezmienniki przejść oznaczają, że wykonanie cyklu pracy czytelnika lub pisarza powoduje powrót do stanu początkowego.

Rozważmy teraz dwa modele systemu producent-konsument (zob. rys. 4.4 i rys. 4.5). Dla sieci na rys. 4.4 macierz uzupełnioną układu (5.63) można przekształcić do postaci bazowej (5.38):

$$U = \begin{pmatrix} 1 & 0 & 0 & -\frac{3}{2} & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & -\frac{3}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad (5.69)$$

przy czym kolumny drugą i trzecią zamieniono miejscami. Rozwiązanie ogólne (po uporządkowaniu zmiennych) jest postaci:

$$x = \begin{pmatrix} \frac{3}{2}\alpha \\ \frac{3}{2}\alpha \\ \alpha \\ \alpha \end{pmatrix}. \quad (5.70)$$

Przykładem niezmiennika przejść jest wektor $I = (3, 3, 2, 2)$, który jednocześnie pokrywa sieć. Łatwo sprawdzić, że jest to niezmiennik realizowalny. Odpowiedni ciąg przejść prowadzący od znakowania M_0 może mieć postać: $t_1, t_2, t_1, t_2, t_1, t_2, t_3, t_4, t_3, t_4$. Z postaci niezmiennika wynika, że w celu powrotu do stanu początkowego jest konieczne wykonanie trzech cykli pracy producenta i dwóch konsumenta. Wektor $I = (3, 3, 2, 2)$ jest również niezmiennikiem przejść dla sieci przedstawionej na rys. 4.5.

5.9. Pułapki i zatrzaski

Analiza występowania w sieci pułapek i zatrzasków pozwala rozstrzygać niektóre własności związane z żywotnością sieci. Pułapka jest zbiorem miejsc, który, jeśli jest znakowany dla pewnego znakowania M , to pozostaje już zbiorem znakowanym dla dowolnego znakowania osiągalnego z M . Zatrzask jest zbiorem miejsc, który, jeśli jest czysty dla pewnego znakowania M , to pozostaje już zbiorem czystym dla dowolnego znakowania osiągalnego z M . Omówimy teraz sposób wyznaczania zatrzasków i pułapek dla sieci uogólnionych oraz wpływ ich występowania na własności sieci.

Niech dana będzie sieć uogólniona $\mathcal{N} = (P, T, A, W, M_0)$.

Definicja 5.19. Niepusty zbiór miejsc $P' \subseteq P$ nazywamy *zatrzaskiem*, jeżeli każde przejście wejściowe zbioru P' jest również jego przejściem wyjściowym, tzn. gdy

$$In(P') \subseteq Out(P'). \quad (5.71)$$

Niepusty zbiór miejsc $P' \subseteq P$ nazywamy *pułapką*, jeżeli każde przejście wyjściowe zbioru P' jest również jego przejściem wejściowym, tzn. gdy

$$Out(P') \subseteq In(P'). \quad (5.72)$$

Bezpośrednio z definicji 5.19 wynika wniosek 5.8.

Wniosek 5.8. Niech \mathcal{N} będzie siecią uogólnioną.

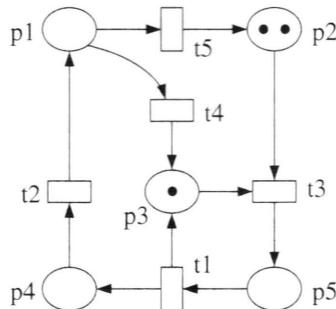
- (1) Jeżeli zbiór P' jest znakowaną pułapką przy znakowaniu M , to jest on również znakowaną pułapką przy dowolnym znakowaniu $M' \in \mathcal{R}(M)$.
- (2) Jeżeli zbiór P' jest czystym zatrzaskiem przy znakowaniu M , to jest on również czystym zatrzaskiem przy dowolnym znakowaniu $M' \in \mathcal{R}(M)$.
- (3) Jeżeli zachodzi warunek $In(P) \subseteq Out(P) = T$, to cała sieć stanowi zatrzask.
- (4) Suma mnogościowa dwóch zatrzasków jest również zatrzaskiem.
- (5) Suma mnogościowa dwóch pułapek jest również pułapką. ■

W prezentowanych w dalszej treści tego podrozdziału twierdzeniach wystarczy ograniczyć się do minimalnych (ze względu na relację zawierania) zatrzasków i maksymalnych zawartych w nich pułapek. *Minimalne zatrzaski* sieci \mathcal{N} możemy wyznaczyć w sposób następujący:

- (1) Tworzymy zbiór kandydatów postaci $\{p\}$, przy czym $p \in P$. Zbiór minimalnych zatrzasków jest wówczas jeszcze pusty.
- (2) Sprawdzamy kandydatów, czy są zatrzaskami. Jeśli rozważany kandydat jest zatrzaskiem i nie zawiera żadnego zatrzasku występującego w zbiorze minimalnych zatrzasków, to dołączamy go do tego zbioru.
Jeżeli kandydat nie jest zatrzaskiem, to nowego kandydata (kandydatów) tworzymy, dodając do rozważanego zbioru miejsc jedno miejsce wejściowe jednego z jego przejść wejściowych, które nie jest jego przejściem wyjściowym. Nowych kandydatów umieszczamy w zbiorze kandydatów (o ile nie są już w nim i nie byli jeszcze rozpatrywani).
- (3) Po rozważeniu wszystkich kandydatów otrzymujemy zbiór minimalnych zatrzasków.

Sprawdzanie niektórych własności sieci wymaga stwierdzenia, czy zatrzask zawiera niepustą pułapkę. Maksymalne pułapki zawarte w minimalnych zatrzaskach możemy wyznaczyć następująco. Rozważamy każdy z minimalnych zatrzasków. Jeżeli jest on również pułapką, to jest to maksymalna pułapka odpowiadająca danemu zatrzaskowi. W przeciwnym przypadku usuwamy z rozważanego zbioru miejsc te z nich, które mają przejścia wyjściowe niebędące jednocześnie przejściami wejściowymi, i sprawdzamy, czy ten nowy zbiór jest pułapką. Postępowanie kontynuujemy, aż rozważany zbiór będzie pułapką (być może, pustą).

Rozważmy sieć \mathcal{N} przedstawioną na rys. 5.23. Wyznaczenie zbioru minimalnych zatrzasków rozpoczynamy od przygotowania początkowej listy kandydatów:



Rys. 5.23. Sieć znakowana

- $\{p1\}, \{p2\}, \{p3\}, \{p4\}, \{p5\}$.

Dla pierwszego ze zbiorów mamy: $In(\{p1\}) = \{t2\}$ i $Out(\{p1\}) = \{t4, t5\}$. Ponieważ $In(\{p1\}) \not\subseteq Out(\{p1\})$, więc zbiór $\{p1\}$ nie jest zatrzaskiem. Usuwamy go z listy, a jego miejsce zajmuje zbiór $\{p1, p4\}$ (miejscie $p4$ jest miejscem wejściowym przejścia $t2$). Kontynuując przegląd wszystkich zbiorów jednoelementowych, otrzymujemy nową listę kandydatów (w przypadku zbiorów $\{p3\}$ i $\{p5\}$ otrzymujemy dwóch nowych kandydatów, ale w jednym przypadku kandydat się powtarza):

- $\{p1, p4\}, \{p1, p2\}, \{p1, p3\}, \{p3, p5\}, \{p4, p5\}, \{p2, p5\}$.

Powtarzając ten algorytm, otrzymujemy listę:

- $\{p1, p4, p5\}, \{p1, p2, p4\}, \{p1, p3, p4\}, \{p1, p3, p5\}, \{p2, p4, p5\}, \{p3, p4, p5\}, \{p1, p2, p5\}$.

W kolejnym kroku otrzymujemy już tylko dwóch kandydatów: $\{p1, p2, p4, p5\}$ i $\{p1, p3, p4, p5\}$. Oba te zbiorы są minimalnymi zatrzaskami.

Przy wyznaczaniu maksymalnych pułapek zawartych w otrzymanych zatrzaskach lista początkowa jest postaci:

- $\{p1, p2, p4, p5\}, \{p1, p3, p4, p5\}$.

Dla pierwszego ze zbiorów mamy: $In(\{p1, p2, p4, p5\}) = \{t1, t2, t3, t5\}$ i $Out(\{p1, p2, p4, p5\}) = \{t1, t2, t3, t4, t5\}$. Ponieważ $Out(\{p1, p2, p4, p5\}) \not\subseteq In(\{p1, p2, p4, p5\})$, zatem zbiór $\{p1, p2, p4, p5\}$ nie jest pułapką. Usuwamy go z listy, a jego miejsce zajmuje zbiór $\{p2, p4, p5\}$ (miejscie $p1$ ma przejście wyj-

ściowe $t4$ niebędące przejściem wejściowym rozważanego zbioru). W przypadku drugiego zbioru nowym kandydatem jest zbiór $\{p3, p4, p5\}$. Lista kandydatów przyjmuje więc postać:

- $\{p2, p4, p5\}, \{p3, p4, p5\}$.

Po kolejnym kroku otrzymujemy listę:

- $\{p2, p5\}, \{p3, p5\}$.

Zbiór $\{p3, p5\}$ jest pułapką odpowiadającą zatrzaskowi $\{p1, p3, p4, p5\}$. W kolejnym kroku rozważamy już tylko jednego kandydata. Nowym kandydatem zamiast zbioru $\{p2, p5\}$ jest zbiór $\{p2\}$. Nie jest on pułapką i zamiast niego dodajemy do listy zbiór pusty, który jest pułapką odpowiadającą zatrzaskowi $\{p1, p2, p4, p5\}$.

Niektóre zatrzaski i pułapki można wyznaczyć metodami algebry liniowej przy wyznaczaniu niezmienników miejsc. Nośnik niezmiennika miejsc sieci uogólnionej jest jednocześnie zatrzaskiem i pułapką.

Twierdzenie 5.20. Jeżeli istnieje nieujemny niezmiennik miejsc sieci \mathcal{N} , to zbiór miejsc wyznaczony przez ten niezmiennik jest zarówno zatrzaskiem, jak i pułapką. ■

Dowód. Niech I będzie niezmiennikiem miejsc sieci \mathcal{N} , a zbiór $P' \subseteq P$ nośnikiem tego niezmiennika. Ważona liczba znaczników (wagi są wyznaczone przez dodatnie współrzędne niezmiennika I) w miejscach ze zbioru P' nie zmienia się przy wykonywaniu dowolnego przejścia $t \in T$. Jeżeli przejście t usuwa co najmniej jeden znacznik z miejsca $p \in P'$, to musi również dodawać co najmniej jeden znacznik do któregoś z miejsc ze zbioru P' , aby zachowana została ważona suma znaczników. Wynika stąd, że dowolne przejście wyjściowe zbioru P' jest także jego przejściem wejściowym, czyli zbiór P' jest pułapką. Analogicznie można pokazać, że zbiór P' jest również zatrzaskiem sieci \mathcal{N} . □

W celu dalszych rozważań wprowadźmy następujące oznaczenia:

$$In(p)_{\min} = \min_{t \in In(p)} \{W(t, p)\}, \quad (5.73)$$

$$In(p)_{\max} = \max_{t \in In(p)} \{W(t, p)\}, \quad (5.74)$$

$$Out(p)_{\min} = \min_{t \in Out(p)} \{W(p, t)\}, \quad (5.75)$$

$$Out(p)_{\max} = \max_{t \in Out(p)} \{W(p, t)\}. \quad (5.76)$$

Definicja 5.20. Niech $\mathcal{N}' = (P', T', A', W', M'_0)$ będzie podsiecią⁵ sieci uogólnionej $\mathcal{N} = (P, T, A, W, M_0)$.

(1) Podsieć \mathcal{N}' nazywamy *jednolitą*, jeżeli jest spełniony warunek:

$$\forall p \in P' \quad \forall t_1, t_2 \in Out(p): W'(p, t_1) = W'(p, t_2). \quad (5.77)$$

(2) Podsieć \mathcal{N}' nazywamy *nieblokującą*, jeżeli jest spełniony warunek:

$$\forall p \in P': Out(p) \neq \emptyset \Rightarrow In(p)_{\min} \geq Out(p)_{\min}. \quad (5.78)$$

(3) Podsieć \mathcal{N}' nazywamy *silnie nieblokującą*, jeżeli jest spełniony warunek:

$$\forall p \in P': Out(p) \neq \emptyset \Rightarrow In(p)_{\min} \geq Out(p)_{\max}. \quad (5.79)$$

Ponadto przyjmujemy, że $In(p)_{\min} = 0$, jeżeli $In(p) = \emptyset$. ■

Powyższa definicja będzie się odnosić także do podsieci wyznaczonych przez zatraski. Niech P' będzie przykładem zatrasku. Każde przejście dodające znacznik (lub znaczniki) do miejsc ze zbioru P' usuwa również co najmniej jeden znacznik z miejsc tego zbioru. Mogą natomiast istnieć przejścia, które wyłącznie usuwają znaczniki. Jeżeli przy pewnym znakowaniu zbiór P' stanie się zbiorem czystym, to już takim pozostanie, a zatem żadne z przejść należących do zbioru $Out(P')$ nie będzie już aktywne (przejścia takie nie są więc żywe). W dalszej treści tego podrozdziału omówimy twierdzenia, które łączą własności zatrasków z właściwością żywotności sieci.

Definicja 5.21. Niech M będzie znakowaniem sieci \mathcal{N} . Zbiór miejsc $P' \subseteq P$ nazywamy *minimalnie znakowanym* przy znakowaniu M , jeżeli jest spełniony warunek:

$$\exists p \in P': M(p) \geq Out(p)_{\min}. \quad (5.80)$$

Zbiór miejsc $P' \subseteq P$ nazywamy *maksymalnie znakowanym* przy znakowaniu M , jeżeli jest spełniony warunek:

$$\exists p \in P': M(p) \geq Out(p)_{\max}. \quad (5.81)$$

⁵ Podsieć \mathcal{N}' spełnia warunki definicji 2.12, a ponadto funkcje W' i M'_0 są obcięciem funkcji W i M_0 odpowiednio do zbiorów A' i P' .

Definicja 5.22. Zatrzask P' sieci \mathcal{N} nazywamy *minimalnie kontrolowanym*, jeżeli jest minimalnie znakowany dla dowolnego znakowania $M \in \mathcal{R}(M_0)$. Sieć \mathcal{N} ma własność *minimalnie kontrolowanych zatrzasków*, jeżeli wszystkie zatrzaski sieci \mathcal{N} są minimalnie kontrolowane.

Zatrzask P' sieci \mathcal{N} nazywamy *maksymalnie kontrolowanym*, jeżeli jest maksymalnie znakowany dla dowolnego znakowania $M \in \mathcal{R}(M_0)$. Sieć \mathcal{N} ma własność *maksymalnie kontrolowanych zatrzasków*, jeżeli wszystkie zatrzaski sieci \mathcal{N} są maksymalnie kontrolowane. ■

Każdy zatrzask sieci, który jest maksymalnie kontrolowany, jest również minimalnie kontrolowany. Jeżeli podsieć wyznaczona przez zatrzask jest jednolita, to zatrzask minimalnie kontrolowany jest również maksymalnie kontrolowany.

Twierdzenie 5.21. Jeżeli sieć \mathcal{N} jest żywa, to ma własność minimalnie kontrolowanych zatrzasków. ■

Dowód. Wystarczy pokazać, że każdy minimalny zatrzask sieci jest minimalnie kontrolowany. Niech \mathcal{N} będzie żywą siecią uogólnioną i niech P' będzie zatrzaskiem sieci. Przypuśćmy, że istnieje znakowanie $M \in \mathcal{R}(M_0)$, takie że dla dowolnego miejsca $p \in P': M(p) < Out(p)_{\min}$. Przy takim znakowaniu M nie jest aktywne żadne przejście należące do zbioru $Out(P')$. Ponieważ P' jest zatrzaskiem ($In(P') \subseteq Out(P')$), więc żadne z przejść ze zbioru $Out(P')$ nie jest aktywne dla żadnego znakowania $M' \in \mathcal{R}(M)$; jest to sprzeczne z założeniem, że sieć jest żywa. □

Twierdzenie 5.22. Jeżeli sieć \mathcal{N} ma własność maksymalnie kontrolowanych zatrzasków, to w zbiorze $\mathcal{R}(M_0)$ nie ma znakowania martwego. ■

Dowód. Przypuśćmy, że istnieje martwe znakowanie $M \in \mathcal{R}(M_0)$. Spełniony musi być więc następujący warunek:

$$\forall t \in T \quad \exists p_t \in In(t): M(p_t) < W(p_t, t) \leqslant Out(p_t)_{\max}. \quad (5.82)$$

Niech $P' = \sum_{t \in T} \{p_t\}$ będzie zbiorem takich miejsc. Ponieważ $Out(P') = T$, więc musi być spełniony warunek $In(P') \subseteq Out(P')$, a zatem zbiór P' jest zatrzaskiem. Zatrzask P' nie jest maksymalnie znakowany przy znakowaniu M , co jest sprzeczne z założeniem. □

Można podać dwa warunki powiązane ze strukturą sieci, które pozwalają zapewnić, że zatrzask jest minimalnie lub maksymalnie kontrolowany. Pierwszy z nich dotyczy istnienia pułapek zawartych w zatrzaskach, a drugi wiąże się z istnieniem niezmienników miejsc.

Twierdzenie 5.23. Niech P' będzie zatrzaskiem sieci \mathcal{N} . Jeżeli jest spełniony jeden z poniższych warunków, to zatrzask P' jest minimalnie kontrolowany.

- (1) Istnieje pułapka $P'' \subseteq P'$, która jest minimalnie znakowana przy znakowaniu M_0 oraz podsieć wyznaczona przez zbiory P'' i $Out(P'')$ jest nieblokująca.
- (2) Istnieje niezmiennik miejsc I , taki że:

$$\begin{aligned} & \forall p \in (\{q \in P : I(q) < 0\} \cap P') : Out(p)_{\min} = 1 \\ & \wedge \{q \in P : I(q) > 0\} \subseteq P' \\ & \wedge I \circ M_0 > \sum_{p \in P'} I(p)(Out(p)_{\min} - 1). \end{aligned} \quad \blacksquare \quad (5.83)$$

Dowód. Przypuśćmy, że jest spełniony pierwszy warunek i zatrzask P' nie jest minimalnie kontrolowany. Istnieje więc takie znakowanie $M \in \mathcal{R}(M_0)$, że:

$$\forall p \in P' : M(p) < Out(p)_{\min}. \quad (5.84)$$

Również pułapka $P'' \subseteq P'$ nie jest minimalnie znakowana przy znakowaniu M . Podsieć wyznaczona przez zbiory P'' i $Out(P'')$ jest nieblokującą i minimalnie znakowana przy znakowaniu początkowym. Każde przejście, które usuwa znacznik (lub znaczniki) ze zbioru P'' , dodaje również znacznik (lub znaczniki) do co najmniej jednego miejsca zbioru P'' . Niech p będzie takim miejscem. Liczba znaczników dodawanych do tego miejsca jest nie mniejsza niż $Out(p)_{\min}$, a zatem zbiór P'' jest minimalnie znakowany przy dowolnym znakowaniu osiągalnym z M_0 . Jest to sprzeczne z hipotezą.

Przypuśćmy, że spełniony jest drugi warunek i zatrzask P' nie jest minimalnie kontrolowany. Istnieje więc znakowanie $M \in \mathcal{R}(M_0)$, takie że:

$$\forall p \in P' : M(p) \leq Out(p)_{\min} - 1. \quad (5.85)$$

Ponieważ $\forall p \in (\{q \in P : I(q) < 0\} \cap P') : Out(p)_{\min} = 1$, zatem:

$$\sum_{p \in P'} I(p)M(p) \leq \sum_{p \in P'} I(p)(Out(p)_{\min} - 1). \quad (5.86)$$

Ponadto, dla $p \in P - P'$, $I(p) \leq 0$, więc:

$$I \circ M = \sum_{p \in P} I(p)M(p) \leq \sum_{p \in P'} I(p)M(p), \quad (5.87)$$

a zatem:

$$I \circ M \leq \sum_{p \in P'} I(p)(Out(p)_{\min} - 1). \quad (5.88)$$

Ponieważ I jest niezmiennikiem miejsc, więc spełnia twierdzenie 5.13; ponadto z założenia $I \circ M_0 > \sum_{p \in P'} I(p)(Out(p)_{\min} - 1)$, a zatem nierówność (5.88) nie może być spełniona. Tak więc hipoteza upada. \square

Spełnienie przez sieć uogólnioną warunku 1 twierdzenia 5.23 dla dowolnego jej zatrzasku jest określane jako *własność zatrzasku-pułapki*.

Definicja 5.23. Sieć uogólniona \mathcal{N} ma właściwość zatrzasku-pułapki, jeżeli każdy zatrzask zawiera pułapkę minimalnie znakowaną przy znakowaniu M_0 . ■

Twierdzenie 5.24. Niech P' będzie zatrzaskiem sieci \mathcal{N} . Jeżeli spełniony jest jeden z poniższych warunków, to zatrzask P' jest maksymalnie kontrolowany.

- (1) Istnieje pułapka $P'' \subseteq P'$, która jest maksymalnie znakowana przy znakowaniu początkowym oraz podsieć wyznaczona przez zbiory P'' i $Out(P'')$ jest silnie nieblokująca.
- (2) Istnieje niezmiennik miejsc I , taki że:

$$\begin{aligned} \forall p \in (\{q \in P : I(q) < 0\} \cap P') : Out(p)_{\max} = 1 \\ \wedge \{q \in P : I(q) > 0\} \subseteq P' \\ \wedge I \circ M_0 > \sum_{p \in P'} I(p)(Out(p)_{\max} - 1). \end{aligned} \quad (5.89) \quad \blacksquare$$

Dowód. Przypuśćmy, że jest spełniony pierwszy warunek i zatrzask P' nie jest maksymalnie kontrolowany. Istnieje więc znakowanie $M \in \mathcal{R}(M_0)$, takie że:

$$\forall p \in P' : M(p) < Out(p)_{\max}. \quad (5.90)$$

Również pułapka $P'' \subseteq P'$ nie jest maksymalnie znakowana przy znakowaniu M . Podsieć wyznaczona przez zbiory P'' i $Out(P'')$ jest silnie nieblokująca i maksymalnie znakowana przy znakowaniu początkowym. Każde przejście, które usuwa znacznik (lub znaczniki) ze zbioru P'' , dodaje również znacznik (lub znaczniki) do co najmniej jednego miejsca zbioru P'' . Niech p będzie takim miejscem. Liczba znaczników dodawanych do tego miejsca jest nie mniejsza niż $Out(p)_{\max}$, więc

zbiór P'' jest maksymalnie znakowany przy dowolnym znakowaniu osiągalnym z M_0 . Jest to sprzeczne z hipotezą.

Przypuśćmy, że spełniony jest drugi warunek i zatrzask P' nie jest maksymalnie kontrolowany. Istnieje więc znakowanie $M \in \mathcal{R}(M_0)$, takie że:

$$\forall p \in P': M(p) \leqslant Out(p)_{\max} - 1. \quad (5.91)$$

Ponieważ $\forall p \in (\{q \in P : I(q) < 0\} \cap P') : Out(p)_{\max} = 1$, zatem:

$$\sum_{p \in P'} I(p)M(p) \leqslant \sum_{p \in P'} I(p)(Out(p)_{\max} - 1). \quad (5.92)$$

Ponadto, dla $p \in P - P'$, $I(p) \leqslant 0$, więc:

$$I \circ M = \sum_{p \in P} I(p)M(p) \leqslant \sum_{p \in P'} I(p)M(p), \quad (5.93)$$

a zatem:

$$I \circ M \leqslant \sum_{p \in P'} I(p)(Out(p)_{\max} - 1). \quad (5.94)$$

Ponieważ I jest niezmiennikiem miejsc, więc spełnia twierdzenie 5.13; ponadto z założenia $I \circ M_0 > \sum_{p \in P'} I(p)(Out(p)_{\max} - 1)$, a zatem nierówność (5.94) nie może być spełniona. Tak więc hipoteza upada. \square

Podsumowując: istnienie własności minimalnie kontrolowanych zatrzasków jest warunkiem koniecznym żywotności sieci, a własności maksymalnie kontrolowanych zatrzasków – warunkiem wystarczającym braku martwych znakowań w zbiorze $\mathcal{R}(M_0)$. Jeżeli dla sieci uogólnionej przyjmiemy dodatkowe ograniczenia dotyczące jej struktury, to można udowodnić, że własność maksymalnie kontrolowanych zatrzasków jest warunkiem wystarczającym żywotności sieci.

Twierdzenie 5.25. Niech \mathcal{N} będzie siecią uogólnioną spełniającą warunek:

$$\forall t \in T : |\{p \in In(t) : |Out(p)| > 1\}| \leqslant 1, \quad (5.95)$$

tzn. każde przejście sieci ma co najwyżej jedno miejsce wejściowe, które ma więcej niż jedno przejście wyjściowe.

- (1) Jeżeli sieć \mathcal{N} ma własność maksymalnie kontrolowanych zatrzasków, to jest żywa.

- (2) Jeżeli sieć \mathcal{N} jest jednolita, to jest żywa wtedy i tylko wtedy, gdy ma własność maksymalnie kontrolowanych zatrzasków. ■

Twierdzenie 5.26. Jeżeli sieć \mathcal{N} jest jednolita i jest rozszerzoną siecią prostą, to sieć jest żywa wtedy i tylko wtedy, gdy ma własność maksymalnie kontrolowanych zatrzasków. ■

Dowody powyższych twierdzeń można znaleźć w pracy [2].

Dla sieci znakowanych przedstawione w tym podrozdziale twierdzenia przyjmują prostszą formę. Wagi wszystkich łuków są równe 1; a zatem sieci są jednolite. Sieć, która ma własność minimalnie kontrolowanych zatrzasków, ma również własność maksymalnie kontrolowanych zatrzasków. W obu przypadkach własność taka oznacza, że każdy zatrzask sieci jest znakowany dla dowolnego znakowania $M \in \mathcal{R}(M_0)$. Definicja własności zatrzasku-pułapki, dla sieci znakowanych przyjmuje następującą postać.

Definicja 5.24. Sieć znakowana $\mathcal{N} = (P, T, A, M_0)$ ma własność zatrzasku-pułapki, jeżeli każdy zatrzask zawiera znakowaną przy M_0 pułapkę. ■

Wniosek 5.9. Jeżeli sieć $\mathcal{N} = (P, T, A, M_0)$ jest żywa, to każdy zatrzask sieci jest znakowany dla dowolnego znakowania $M \in \mathcal{R}(M_0)$. ■

Wniosek 5.10. Jeżeli sieć $\mathcal{N} = (P, T, A, M_0)$ ma własność pułapki-zatrzasku, to w zbiorze $\mathcal{R}(M_0)$ nie ma znakowania martwego. ■

Twierdzenie 5.27. Jeżeli znakowanie $M \in \mathcal{R}(M_0)$ jest martwym znakowaniem sieci $\mathcal{N} = (P, T, A, M_0)$, to zbiór miejsc $P' = \{p \in P : M(p) = 0\}$ jest nieznakowanym zatrzaskiem sieci \mathcal{N} . ■

Dowód. Zbiór P' nie może być zbiorem pustym, gdyż w przeciwnym wypadku wszystkie przejścia byłyby aktywne przy znakowaniu M . Rozważmy dowolne przejście $t \in In(P')$. Przejście t nie jest aktywne przy znakowaniu M , zatem $In(t) \cap P' \neq \emptyset$, więc $t \in Out(P')$. Ponieważ przejście t było dowolne, zatem zbiór P' jest zatrzaskiem sieci \mathcal{N} . Ze sposobu określenia zbioru P' wynika, że jest on nieznakowany. □

Wniosek 5.11. Niech $P' \subseteq P$ będzie czystym zatrzaskiem sieci \mathcal{N} przy znakowaniu M_0 . Dla dowolnego przejścia t , jeżeli $In(t) \cap P' \neq \emptyset$, to przejście t jest martwe. ■

Metody wyznaczania zatrzasków i pułapek możemy również stosować do sprawdzania, czy dana sieć jest rozkładalna na maszyny stanowe. Z listy minimalnych

zatrzasków wygenerowanych dla sieci \mathcal{N} wybieramy te, które jednocześnie są pułapkami. Uzyskamy w ten sposób listę SSMS-składowych sieci \mathcal{N} . Opierając się na tej liście, sprawdzamy, czy sieć jest rozkładalna na maszyny stanowe.

Rozważmy przykład przedstawionej na rys. 4.3 sieci modelującej wzajemne wykluczanie dla problemu czytelników i pisarzy. Sieć ma trzy zatrzaski: $P_1 = \{p1, p2, p3\}$, $P_2 = \{p4, p5, p6\}$ i $P_3 = \{p3, p6, p7\}$. Zbiory te są jednocześnie pułapkami zawartymi w tych zatrzaskach. Są one wyznaczone przez niezmienniki miejsc, odpowiednio: $I_1 = (1, 1, 1, 0, 0, 0, 0)$, $I_2 = (0, 0, 0, 1, 1, 1, 0)$ i $I_3 = (0, 0, 1, 0, 0, 4, 1)$ (zob. wzór (5.52)).

Przy zadanym znakowaniu początkowym wszystkie zatrzaski są maksymalnie znakowane i nieblokujące, przy czym zatrzaski P_1 i P_2 są silnie nieblokujące. Na podstawie twierdzenia 5.23 (1) możemy wnioskować, że wszystkie zatrzaski rozważanej sieci są minimalnie kontrolowane, natomiast z twierdzenia 5.24 (1) wynika, że maksymalnie kontrolowane są zatrzaski P_1 i P_2 .

Pomiśmy znakowanie początkowe rozważanej sieci. Na podstawie twierdzeń 5.23 (2) i 5.24 (2) możemy określić warunki, które powinno spełniać znakowanie początkowe, aby rozważane zatrzaski były minimalnie lub maksymalnie kontrolowane. Rozważmy warunek (5.83) oraz niezmienniki I_1 , I_2 i I_3 . Dla wszystkich niezmienników spełniony jest pierwszy i drugi człon koniunkcji, a zatem wystarczy, aby była spełniona nierówność $I \circ M_0 > \sum_{p \in P'} I(p)(Out(p)_{\min} - 1)$. Dla wszystkich miejsc $Out(p)_{\min} = 1$, zatem lewa strona nierówności jest równa 0 dla każdego z niezmienników. Stąd wynika, że zatrzaski rozważanej sieci są minimalnie kontrolowane, o ile są znakowane. Jeżeli warunek ten nie jest spełniony, to sieć nie jest żywa.

Analogicznie w przypadku warunku (5.89) wystarczy, aby spełniona była nierówność $I \circ M_0 > \sum_{p \in P'} I(p)(Out(p)_{\max} - 1)$. Dla zatrzasków I_1 i I_2 nierówność ta jest spełniona, jeżeli są one znakowane. Dla zatrzasku P_3 prawa strona nierówności jest równa 3, a zatem zatrzask P_3 jest maksymalnie kontrolowany, jeżeli $M_0(p3) + 4M_0(p6) + M_0(p7) > 3$. Jeżeli spełnione są te warunki, to w zbiorze $\mathcal{R}(M_0)$ nie ma znakowania martwego.

Weźmy pod uwagę przedstawioną na rys. 3.14 sieć znakowaną, modelującą system sygnalizacji świetlnej. Rozważana sieć zawiera aż 10 minimalnych zatrzasków: $\{p1, p2\}$, $\{p3, p4\}$, $\{p3, p7\}$, $\{p4, p8\}$, $\{p5, p6\}$, $\{p7, p8\}$, $\{p1, p3, p6\}$, $\{p1, p6, p8\}$, $\{p2, p4, p5\}$, $\{p2, p5, p7\}$. Wszystkie te zbiory są jednocześnie pułapkami i są znakowane, a więc sieć ma własność pułapki-zatrzasku. Podane zbiory miejsc pozwalają jednocześnie utworzyć dziesięcioelementową listę SSMS-składowych sieci \mathcal{N} . Łatwo sprawdzić, że sieć jest rozkładalna na maszyny stanowe i rozkład ten nie jest jednoznaczny.

Tabela 5.5. Minimalne zatrzaski i odpowiadające im maksymalne pułapki

Minimalne zatrzaski	Maksymalne pułapki
$\{p_6, p_{11}, p_{16}\}$	$\{p_6, p_{11}, p_{16}\}$
$\{p_7, p_{12}, p_{17}\}$	$\{p_7, p_{12}, p_{17}\}$
$\{p_8, p_{13}, p_{18}\}$	$\{p_8, p_{13}, p_{18}\}$
$\{p_9, p_{14}, p_{19}\}$	$\{p_9, p_{14}, p_{19}\}$
$\{p_{10}, p_{15}, p_{20}\}$	$\{p_{10}, p_{15}, p_{20}\}$
$\{p_1, p_{11}, p_{16}, p_{20}\}$	$\{p_1, p_{11}, p_{16}, p_{20}\}$
$\{p_2, p_{12}, p_{16}, p_{17}\}$	$\{p_2, p_{12}, p_{16}, p_{17}\}$
$\{p_3, p_{13}, p_{17}, p_{18}\}$	$\{p_3, p_{13}, p_{17}, p_{18}\}$
$\{p_4, p_{14}, p_{18}, p_{19}\}$	$\{p_4, p_{14}, p_{18}, p_{19}\}$
$\{p_5, p_{15}, p_{19}, p_{20}\}$	$\{p_5, p_{15}, p_{19}, p_{20}\}$
$\{p_1, p_2, p_3, p_4, p_5, p_{16}, p_{17}, p_{18}, p_{19}, p_{20}\}$	\emptyset

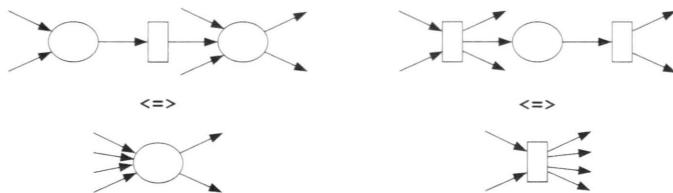
Sieć modelująca problem pięciu filozofów (zob. rys. 3.16) nie ma własności pułapki-zatrzasku. Listę minimalnych zatrzasków i odpowiadających im maksymalnych pułapek przedstawiono w tabeli 5.5. W przypadku tej sieci można było stwierdzić bez wyznaczania listy zatrzasków i pułapek, że nie może ona mieć własności pułapki-zatrzasku. Wynika to z faktu, że już wcześniej wskazano martwe znakowanie tej sieci (zob. wniosek 5.10). Sieć jest natomiast rozkładalna na maszyny stanowe; 10 pierwszych zatrzasków wyznacza 10 SSMS-składowych tej sieci.

5.10. Metody redukcji sieci

Przy analizie sieci uogólnionych o dużej liczbie węzłów można stosować tzw. *metody redukcji sieci*, które pozwalają uprościć strukturę sieci z zachowaniem jej własności, takich jak: żywotność, ograniczoność i bezpieczeństwo. Jednocześnie można używać tych metod przy rozbudowie struktury sieci z zachowaniem jej pierwotnych własności. W tym podrozdziale omówimy sześć wybranych metod redukcji, które można stosować przy analizie sieci znakowanych. Metod tych można też używać podczas analizowania sieci uogólnionych, ale przy zachowaniu pewnych ograniczeń dotyczących wag łuków sieci.

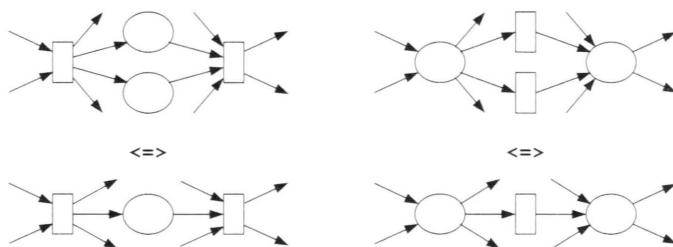
Twierdzenie 5.28. Niech $\mathcal{N} = (P, T, A, M_0)$ będzie siecią znakowaną i niech $\mathcal{N}' = (P', T', A', M'_0)$ będzie siecią, którą otrzymujemy z sieci \mathcal{N} w wyniku zastosowania jednej z metod

redukcji. Sieć N' jest żywa (ograniczona, bezpieczna) wtedy i tylko wtedy, gdy sieć N jest żywa (ograniczona, bezpieczna). ■



Rys. 5.24. Scalanie kolejnych miejsc i scalanie kolejnych przejść

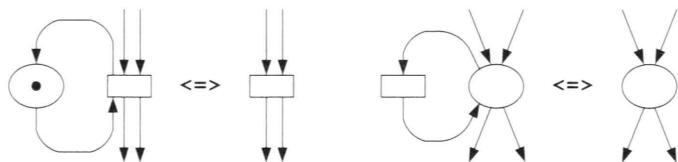
Pierwszą parę metod redukcji sieci tworzą metody scalania kolejnych miejsc i scalania kolejnych przejść. Graficzną reprezentację tych metod przedstawiono na rys. 5.24. Przyjmijmy, że redukcję wykonujemy przy znakowaniu M sieci $N = (P, T, A, M_0)$. Przy scalaniu kolejnych miejsc poszukujemy przejścia t i miejsc p i p' , takich że: $Out(p) = \{t\}$, $In(t) = \{p\}$, $t \in In(p')$ i $M(p) = 0$. W wyniku redukcji węzły t i p są usuwane z sieci, natomiast wszystkie przejścia wejściowe miejsca p stają się przejściami wejściowymi miejsca p' , tzn. łuki (t', p') dla $t' \in In(p)$ będą zastąpione łukami (t', p') (inaczej mówiąc, miejsca p i p' będą scalone w jedno miejsce). Metoda scalania kolejnych przejść działa analogicznie. Ma ona zastosowanie w odniesieniu do pary przejść t , t' i miejsca p spełniających warunki: $In(p) = \{t\}$, $Out(p) = \{t'\}$, $In(t') = \{p\}$ i $M(p) = 0$.



Rys. 5.25. Scalanie równoległych miejsc i scalanie równoległych przejść

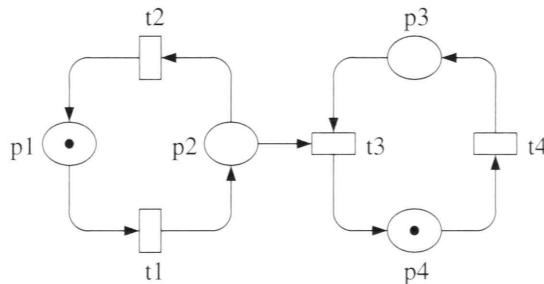
Na rysunku 5.25 przedstawiono drugą parę metod redukcji, to jest metodę scalania równoległych miejsc i metodę scalania równoległych przejść. Przy scalaniu równoległych miejsc poszukuje się miejsc p i p' , takich że $In(p) = In(p') = \{t\}$, $Out(p) = Out(p') = \{t'\}$ i $M(p) = M(p')$. W wyniku redukcji jedno z tych miejsc będzie usunięte z sieci. Analogicznie w metodzie scalania równoległych przejść

poszukuje się przejść t i t' , takich że $In(t) = In(t') = \{p\}$, $Out(t) = Out(t') = \{p'\}$. W wyniku zastosowania tej metody jedno z tych przejść będzie usunięte.



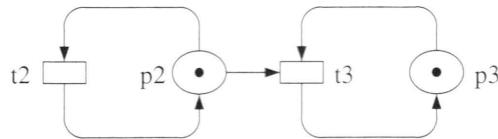
Rys. 5.26. Usuwanie ciasnych pętli miejsc i usuwanie ciasnych pętli przejść

Ostatnia para metod redukcji polega na usuwaniu ciasnych pętli z sieci. Metody usuwania ciasnych pętli miejsc i ciasnych pętli przejść przedstawiono na rys. 5.26. W wyniku zastosowania pierwszej metody usuwa się z sieci miejsce p spełniające warunki $In(p) = Out(p) = \{t\}$, $M(p) = 1$ oraz $In(t) \neq \{p\} \vee Out(t) \neq \{p\}$. Ostatni warunek oznacza, że w wyniku zastosowania tej metody nie możemy otrzymać izolowanego przejścia. Metodę tę można zastosować również wtedy, gdy $M(p) > 1$, ale wówczas otrzymana sieć N' może okazać się siecią bezpieczną, pomimo że sieć N nie była bezpieczna. Druga metoda pozwala na usunięcie przejścia t spełniającego warunki $In(t) = Out(t) = \{p\}$ i $Out(p) \neq \{t\}$.



Rys. 5.27. Sieć znakowana przed zastosowaniem redukcji

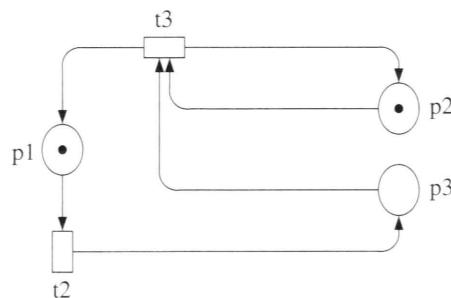
Rozważmy sieć N przedstawioną na rys. 5.27. Jeżeli wykonamy przejście t_1 , to znacznik z miejsca p_1 zostanie przeniesiony do miejsca p_2 . Możemy wówczas zgodnie z metodą scalania kolejnych przejść usunąć nieznakowane miejsce p_1 i połączyć przejścia t_2 i t_1 w jedno przejście t_2 . Podobnie, stosując tę samą metodę po wykonaniu przejścia t_4 , można usunąć nieznakowane miejsce p_4 i połączyć przejścia t_3 i t_4 w jedno przejście t_3 . Efektem wykonania tych przekształceń jest sieć N_1 przedstawiona na rys. 5.28.

Rys. 5.28. Sieć N_1 uzyskana po scaleniu kolejnych przejść

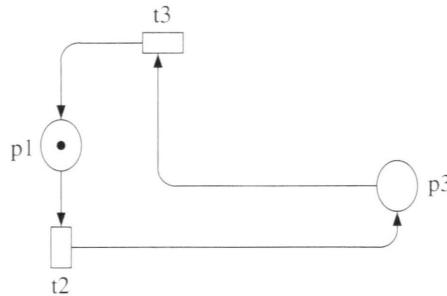
Stosując następnie metodę usuwania ciasnych pętli miejsc w odniesieniu do miejsca p_3 i metodę usuwania ciasnych pętli przejść w odniesieniu do przejścia t_2 , otrzymujemy ostatecznie sieć N_2 przedstawioną na rys. 5.29. Sieć N_2 jest bezpieczna, ale nie jest żywa, zatem sieć N również jest bezpieczna i nie jest żywa.

Rys. 5.29. Sieć N_2 uzyskana po usunięciu ciasnych pętli z sieci N_1

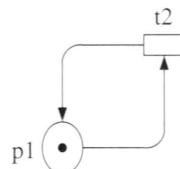
Rozważmy zastosowanie metod redukcji do wybranych sieci znakowanych, przedstawionych w podrozdz. 3.5. Sieć będącą modelem sygnalizacji świetlnej (wersja 2, rys. 3.15) można zredukować do dwóch węzłów. Na kolejnych rysunkach możemy obejrzeć sieci, które otrzymamy: po scaleniu kolejnych przejść t_3 i t_1 (rys. 5.30), po wyeliminowaniu ciasnej pętli miejsca – usunięte miejsce p_2 (rys. 5.31) i po scaleniu kolejnych przejść t_2 i t_3 (rys. 5.32). Obie sieci, zarówno sieć wyjściowa (rys. 3.15), jak i sieć uzyskana w wyniku redukcji (rys. 5.32) – są żywe i bezpieczne.



Rys. 5.30. Sieć otrzymana po scaleniu kolejnych przejść w sieci na rys. 3.15



Rys. 5.31. Sieć otrzymana po usunięciu ciasnej pętli z sieci na rys. 5.30



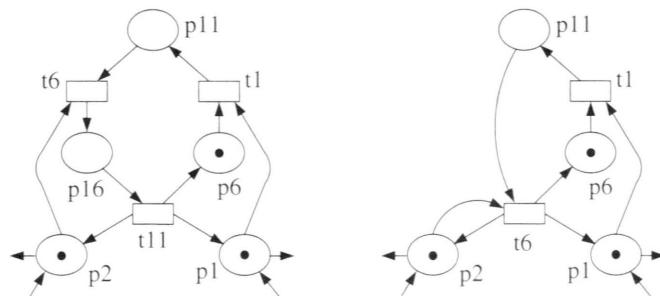
Rys. 5.32. Sieć otrzymana po scaleniu kolejnych przejść w sieci na rys. 5.31

W przypadku w pierwszej wersji modelu sygnalizacji świetlnej (zob. rys. 3.14) można jedynie dwukrotnie zastosować metodę scalania równoległych miejsc w odniesieniu do miejsc p_3 i p_8 oraz p_4 i p_7 .

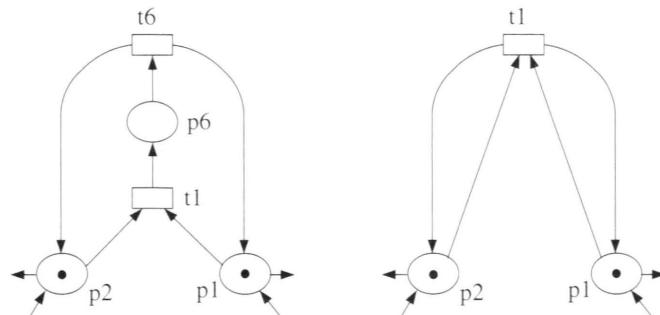
W odniesieniu do pierwszej wersji modelu problemu pięciu filozofów (zob. rys. 3.16) można jedynie pięciokrotnie zastosować regułę scalania kolejnych przejść. Fragment modelu przed i po zastosowaniu tej reguły w odniesieniu do przejść t_6 i t_{11} przedstawiono na rys. 5.33.

W przypadku trzeciej wersji modelu problemu pięciu filozofów (zob. rys. 3.17) nie jest możliwa żadna redukcja struktury sieci ani przy znakowaniu początkowym, ani po ewentualnym wykonaniu wybranych przejść. Można pięciokrotnie zastosować regułę scalania kolejnych przejść do czwartej wersji rozważanego systemu (zob. sieć na rys. 3.18). Fragment modelu przed i po zastosowaniu tej reguły w odniesieniu do przejść t_1 i t_6 przedstawiono na rys. 5.34.

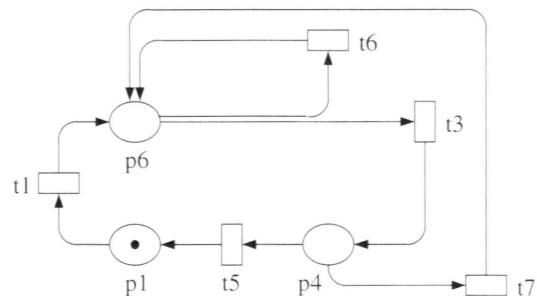
Jak sieć będącą modelem sygnalizacji świetlnej, tak i sieć modelującą protokół komunikacyjny (zob. rys. 3.19) można zredukować do dwóch węzłów. W pierwszym etapie wykonujemy scalenie kolejnych miejsc p_2 i p_3 (pozostaje miejsce p_2) oraz p_4 i p_5 (pozostaje miejsce p_4). Następnie scalamy kolejne miejsca p_6 i p_2 . Sieć będącą wynikiem tych redukcji przedstawiono na rys. 5.35.



Rys. 5.33. Fragment sieci pokazanej na rys. 3.16 po scaleniu kolejnych przejść

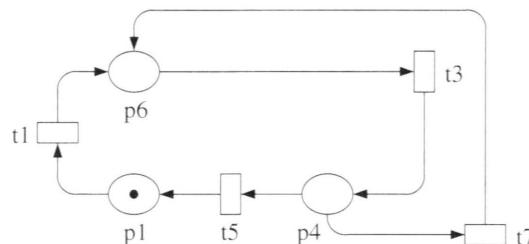


Rys. 5.34. Fragment sieci pokazanej na rys. 3.18 po scaleniu kolejnych przejść

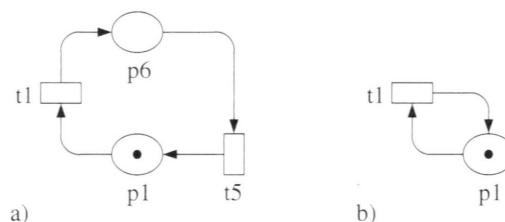


Rys. 5.35. Sieć otrzymana po scaleniu kolejnych miejsc w sieci na rys. 3.19

W kolejnym kroku możemy usunąć przejście t_6 tworzące ciasną pętlę (sieć po redukcji przedstawiono na rys. 5.36), a następnie scalić kolejne miejsca p_6 i p_4 oraz usunąć przejście t_7 tworzące ciasną pętlę. Sieć będącą wynikiem tych redukcji przedstawiono na rys. 5.37 a). W ostatnim kroku scalamy kolejne przejścia t_1 i t_5 . Sieć będącą wynikiem tej ostatniej redukcji przedstawiono na rys. 5.37 b).



Rys. 5.36. Sieć otrzymana po usunięciu ciasnej pętli przejścia z sieci na rys. 5.35



Rys. 5.37. Sieć otrzymana z sieci na rys. 5.36 po scaleniu kolejnych miejsc i usunięciu ciasnej pętli przejścia a) oraz scaleniu kolejnych przejść b)

5.11. Podsumowanie

W tym rozdziale dokonaliśmy przeglądu najczęściej spotykanych metod analizy sieci uogólnionych. Przede wszystkim omówiliśmy grafy osiągalności, których analizowanie jest podstawową metodą weryfikacji dla dowolnej klasy sieci Petriego. Przedstawienie zbioru osiągalnych znakowań w postaci uporządkowanej, mającej postać grafu skierowanego, pozwala zweryfikować wszystkie omawiane w rozdz. 3 własności dynamiczne sieci. Możliwość efektywnego stosowania tej metody jest uzależniona od rozmiaru zbioru osiągalnych znakowań, który może być zbiorem nieskończonym. W celu wyeliminowania tego problemu opracowano kilka metod redukcji grafu osiągalności, m.in. grafy silnie spójnych składowych oraz grafy zredukowane ze względu na symetrie lub trwałe zbiory przejść.

Równie ważne są metody, w których korzysta się z algebraicznej reprezentacji sieci w celu wyznaczania niezmienników miejsc i przejść. Podstawową zaletą tych metod jest możliwość analizowania wielu własności dynamicznych bez konieczności wyznaczania zbioru osiągalnych znakowań. Ponadto omówiliśmy metodę opartą na analizie występowania zatrzasków i pułapek oraz metody redukcji, które pozwalają zmniejszyć liczbę elementów sieci, zachowując wybrane jej własności.



Sieci miejsc i przejść

Sieci *miejsc i przejść* (lub krótko PT-sieci, zob. prace: [48], [52], [56], [70] i [74]) są jedną z najczęściej spotykanych w literaturze klas sieci Petriego. Stanowią one punkt wyjścia do definiowania większości klas sieci niskiego poziomu. W odróżnieniu od sieci uogólnionych sieci miejsc i przejść zawierają funkcję ograniczającą pojemności miejsc. Sieci uogólnione są szczególnym przypadkiem sieci miejsc i przejść, w których wszystkie miejsca mają nieograniczoną pojemność.

Dla dowolnej sieci miejsc i przejść, której miejsca mają ograniczoną pojemność, można łatwo zbudować sieć uogólnioną, mającą takie same własności. Dla takiej pary sieci można ponadto wykazać, że mogą wykonywać identyczne ciągi przejść, a także wskazać funkcję, która wzajemnie jednoznacznie przekształca zbiór znakowań osiągalnych jednej z tych sieci na zbiór znakowań osiągalnych drugiej. W związku z tym termin *sieć miejsc i przejść* często odnosi się w literaturze również do sieci uogólnionych.

W tym rozdziale omówimy definicję sieci miejsc i przejść oraz przykład modelowania z ich zastosowaniem. Wszystkie omawiane własności sieci uogólnionych dotyczą również sieci miejsc i przejść. Zmianie ulega jedynie definicja aktywności przejść, gdyż po wykonaniu przejścia liczba znaczników w miejscach wyjściowych nie może przekroczyć zadeklarowanych pojemności tych miejsc.

6.1. Dynamika sieci miejsc i przejść

W porównaniu z definicją 4.1 sieci uogólnionej sieci miejsc i przejść zawierają dodatkowo tylko funkcję definiującą pojemności miejsc sieci.

Definicja 6.1. Uporządkowaną szóstkę postaci $\mathcal{N} = (P, T, A, K, W, M_0)$ nazywamy *siecią miejsc i przejść* (PT-siecią), jeżeli są spełnione warunki:

- (1) $\mathcal{N} = (P, T, A)$ jest siecią.

- (2) $W: A \rightarrow \mathbb{N}$ jest funkcją wag łuków.
- (3) $K: P \rightarrow \mathbb{N} \cup \{\infty\}$ jest funkcją zwaną *pojemnością miejsc*, określającą dla każdego miejsca maksymalną liczbę znaczników, które może ono zawierać (symbol ∞ oznacza, że dane miejsce ma nieograniczoną pojemność).
- (4) $M_0: P \rightarrow \mathbb{Z}_+$ jest *znakowaniem początkowym* sieci N spełniającym dodatkowo warunek: $\forall p \in P: M_0(p) \leq K(p)$. ■

Wprowadzenie ograniczonej pojemności miejsc ma wpływ zarówno na definicję znakowania sieci, jak i definicję aktywności przejść.

Definicja 6.2. *Znakowaniem* sieci miejsc i przejść N nazywamy dowolną funkcję:

$$M: P \rightarrow \mathbb{Z}_+, \text{ taką że } \forall p \in P: M(p) \leq K(p). \quad (6.1) \quad ■$$

Dla sieci miejsc i przejść można zdefiniować trzy różne reguły wykonywania przejść. *Słaba reguła wykonywania przejść* jest równoważna definicji 4.2. Wymaga się tylko, by miejsca wejściowe przejść zawierały odpowiednią liczbę znaczników. Przy tym podejściu pomija się funkcję pojemności miejsc. Reguły *średnia* i *mocna* uwzględniają pojemność miejsc, a wykonanie przejścia jest możliwe, o ile dodatkowo wszystkie miejsca wyjściowe mają wystarczającą pojemność, by przyjąć dodawane znaczniki.

Definicja 6.3. Niech N będzie siecią miejsc i przejść.

- (1) *Słaba reguła wykonywania przejść*:

Przejście $t \in T$ jest *aktywne* przy znakowaniu M , jeżeli każde z jego miejsc wejściowych zawiera co najmniej tyle znaczników, ile wynosi waga łuku prowadzącego od tego miejsca do przejścia t , tzn.:

$$\forall p \in In(t): M(p) \geq W(p, t). \quad (6.2)$$

- (2) *Średnia reguła wykonywania przejść*:

Przejście $t \in T$ jest *aktywne* przy znakowaniu M , jeżeli jest spełniony warunek (6.2) oraz każde z jego miejsc wyjściowych ma wystarczającą pojemność, by przyjąć tyle znaczników, ile wynosi różnica wag łuków (t, p) i (p, t) , tzn.:

$$\forall p \in Out(t): M(p) - W(p, t) + W(t, p) \leq K(p), \quad (6.3)$$

przy czym, jeżeli miejsce $p \notin In(t)$, to przyjmujemy $W(p, t) = 0$.

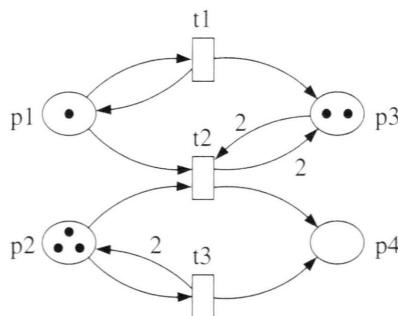
(3) *Mocna reguła wykonywania przejść:*

Przejście $t \in T$ jest aktywne przy znakowaniu M , jeżeli jest spełniony warunek (6.2) oraz każde z jego miejsc wyjściowych ma wystarczającą pojemność, by przyjąć tyle znaczników, ile wynosi waga łuku (t, p) , tzn.:

$$\forall p \in Out(t): M(p) + W(t, p) \leq K(p). \quad (6.4)$$

■

Analizując dynamiczne właściwości danej sieci miejsc i przejść, należy z góry przyjąć, która z reguł wykonywania przejść będzie obowiązywać dla wszystkich przejść w tej sieci. Jeżeli przejście t jest aktywne przy znakowaniu M , to w wyniku jego wykonania powstaje nowe znakowanie M' , zdefiniowane tak jak dla sieci uogólnionych (zob. wzór (4.1)).



Rys. 6.1. Przykład sieci miejsc i przejść

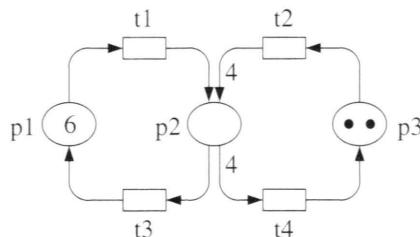
Rozważmy sieć N przedstawioną na rys. 6.1. Jeżeli przyjmiemy funkcję pojemności miejsc $K = (3, 3, 3, 3)^1$, to: przy słabej regule wykonywania przejść są aktywne wszystkie przejścia, przy średniej regule są aktywne przejścia t_1 i t_2 , a przy mocnej regule tylko przejście t_1 jest aktywne.

6.2. Modelowanie z zastosowaniem sieci miejsc i przejść

W rozdziale 4 omówiliśmy model wzajemnego wykluczania dla problemu czytelników i pisarzy w postaci sieci uogólnionej (zob. rys. 4.3). Rysunek 6.2 zawiera model podobnego systemu, ale w postaci sieci miejsc i przejść. Dla sieci przyjęto

¹ Jak w przypadku znakowań sieci, tak i tu zakładamy, że zbiór miejsc jest uporządkowany, zatem funkcję K można zapisać w postaci wektora.

funkcję pojemności miejsc $K = (6, 4, 2)$. Znaczniki w miejscach $p1$ i $p3$ reprezentują odpowiednio czytelników i pisarzy znajdujących się poza czytelnią, a miejsce $p2$ – czytelnię. W miejscu $p2$ mogą się znaleźć maksymalnie 4 znaczniki reprezentujące albo jednego pisarza, albo 4 czytelników. Interpretacja znaczników w miejscu $p2$ zależy od znakowania pozostałych miejsc sieci, które pozwalają sprawdzać, ilu czytelników i pisarzy znajduje się poza czytelnią. Znaczniki dodane do miejsca $p2$ przez przejście $t1$ nie mogą zostać zabrane przez przejście $t4$, ponieważ nie pozwala na to ograniczenie pojemności miejsca $p3$. Analogiczna sytuacja dotyczy znaczników dodawanych do miejsca $p2$ przez przejście $t2$. Nie mogą one zostać zabrane przez przejście $t3$, gdyż nie pozwala na to ograniczenie pojemności miejsca $p1$.



Rys. 6.2. Model wzajemnego wykluczania dla problemu czytelników i pisarzy

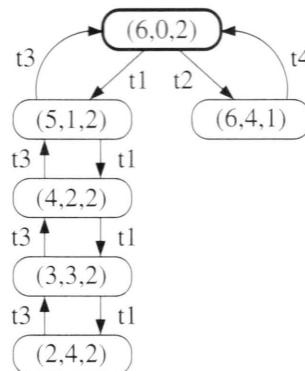
Do analizy własności sieci miejsc i przejść stosuje się takie same metody jak w sieciach uogólnionych (zob. rozdz. 5). Należy jednak zwrócić uwagę na fakt, że funkcja pojemności miejsc może mieć wpływ na pewne własności sieci, na przykład na żywotność przejść, żywotność miejsc czy też występowanie martwych znakowań. Na rysunku 6.3 przedstawiono graf osiągalności dla rozważanej sieci. Przyjęto założenie, że sieć pracuje zgodnie ze średnią regułą wykonywania przejść².

Rozważana sieć \mathcal{N} jest żywa, żywa ze względu na miejsca, ograniczona (zgodnie z pojemnościami miejsc) i odwracalna. Sieć \mathcal{N} nie jest siecią sprawiedliwą. Możliwe jest zagłodzenie zarówno pisarzy, jak i czytelników; jest to wyrażone przez dwa cykle występujące w grafie osiągalności.

Macierz incydencji rozważanej sieci jest postaci:

$$N = \begin{pmatrix} -1 & 0 & 1 & 0 \\ 1 & 4 & -1 & -4 \\ 0 & -1 & 0 & 1 \end{pmatrix}. \quad (6.5)$$

² Średnia reguła wykonywania przejść jest najczęściej stosowaną regułą.



Rys. 6.3. Graf osiągalności dla sieci z rys. 6.2

Rozwiązań układu równań (5.26) jest następujące:

$$x = \begin{pmatrix} \alpha \\ \alpha \\ 4\alpha \end{pmatrix}. \quad (6.6)$$

Przykładem niezmiennika miejsc jest wektor $I = (1, 1, 4)$, pokrywający sieć, i jednocześnie będący wektorem wag, względem którego sieć jest zachowawcza.

Rozwiązań układu równań (5.63) jest następujące:

$$x = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_1 \\ \alpha_2 \end{pmatrix}. \quad (6.7)$$

Przykładami niezmienników są wektory $I_1 = (1, 0, 1, 0)$ i $I_2 = (0, 1, 0, 1)$. Oba niezmienniki są realizowalne. Rozważana sieć jest pokryta niezmiennikami przejść.

6.3. Symulowanie PT-sieci za pomocą sieci uogólnionych

Jak już wspomniałem na początku tego rozdziału, dla dowolnej sieci miejsc i przejść, której miejsca mają ograniczoną pojemność, można zbudować równoważną jej sieć uogólnioną (zob. prace: [69], [74]).

Niech $\mathcal{N} = (P, T, A, K, W, M_0)$ będzie siecią miejsc i przejść. Sieć uogólnioną $\mathcal{N}' = (P', T, A', W', M'_0)$ konstruujemy w sposób następujący. Sieć \mathcal{N}' zawiera wszystkie

miejsca sieci \mathcal{N} , zachowuje dla nich znakowanie początkowe, ma dokładnie te same przejścia co sieć \mathcal{N} oraz zawiera wszystkie jej łuki wraz z ich wagami. Ponadto dla każdego miejsca $p \in P$, dla którego $K(p) < \infty$, dodajemy miejsce uzupełniające p' , dla którego przyjmujemy znakowanie początkowe $M'_0(p') = K(p) - M_0(p)$. Miejsce p' łączymy łukami z przejściami ze zbioru $In(p) \cup Out(p)$, tak aby był spełniony warunek:

$$\forall M \in \mathcal{R}(M'_0): M(p) + M(p') = K(p). \quad (6.8)$$

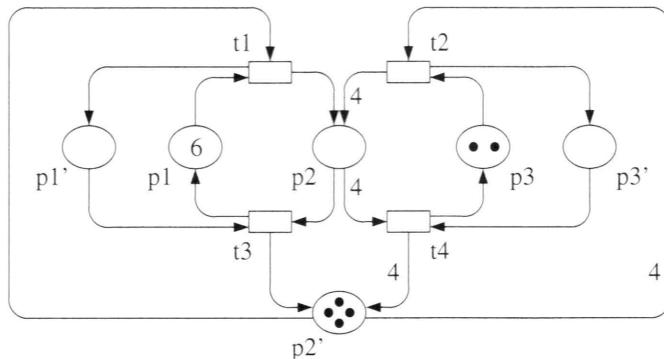
Spełnienie warunku (6.8) wymaga dodania do sieci \mathcal{N}' następujących łuków:

- Jeżeli $t \in In(p) - Out(p)$, to dodajemy łuk (p', t) , taki że $W'(p', t) = W(t, p)$.
- Jeżeli $t \in Out(p) - In(p)$, to dodajemy łuk (t, p') , taki że $W'(t, p') = W(p, t)$.
- Jeżeli $t \in In(p) \cap Out(p) \wedge W(t, p) - W(p, t) > 0$, to dodajemy łuk (p', t) , taki że $W'(p', t) = W(t, p) - W(p, t)$.
- Jeżeli $t \in In(p) \cap Out(p) \wedge W(t, p) - W(p, t) < 0$, to dodajemy łuk (t, p') , taki że $W'(t, p') = W(p, t) - W(t, p)$.

Rozważmy ponownie sieć \mathcal{N} przedstawioną na rys. 6.2. Dla sieci tej zbudujemy równoważną jej sieć uogólnioną. Wszystkie miejsca sieci \mathcal{N} mają zdefiniowaną ograniczoną pojemność $K = (6, 4, 2)$, zatem dla każdego miejsca konieczne jest dodanie miejsca uzupełniającego. Rozważmy na przykład miejsce uzupełniające dla miejsca $p2$. Przy znakowaniu początkowym miejsce to będzie zawierało 4 znaczniki, ponieważ $M_0(p2) = 0$ i $K(p2) = 4$. Wykonanie przejścia $t1$ dodaje jeden znacznik do miejsca $p2$. Aby był spełniony warunek (6.8), konieczne jest dodanie łuku $(p2', t1)$ o wagie 1. Podobnie należy jeszcze dołączyć do sieci łuki $(p2', t2)$, $(t3, p2')$ i $(t4, p2')$ odpowiednio o wagach 4, 1 i 4. Sieć uogólnioną \mathcal{N}' , która jest równoważna sieci \mathcal{N} , przedstawiono na rys. 6.4.

Dodane miejsca $p1'$, $p2'$ i $p3'$ przejmują rolę funkcji pojemności miejsc. Rozważmy na przykład przejście $t1$. Jego wykonanie w sieci \mathcal{N}' jest możliwe, jeżeli miejsce $p2'$ zawiera co najmniej jeden znacznik, tzn. jeżeli istnieje co najmniej jedno wolne miejsce w czytelni. Przedstawionej w tym przykładzie konstrukcji użyliśmy już wcześniej. Miejsce uzupełniające zastosowano do ograniczenia pojemności magazynu w modelu systemu producent-konsument (zob. rys. 4.5).

Niech dane będą sieć miejsc i przejście $\mathcal{N} = (P, T, A, K, W, M_0)$ oraz równoważna jej sieć uogólniona $\mathcal{N}' = (P', T, A', W', M'_0)$. W dalszych rozważaniach przyjmijmy, że sieć \mathcal{N} pracuje zgodnie ze średnią regułą wykonywania przejść. Dla sieci \mathcal{N} i \mathcal{N}' można wskazać funkcję, która wzajemnie jednoznacznie przekształca zbiór znakowań osiągalnych sieci \mathcal{N} na zbiór znakowań osiągalnych sieci \mathcal{N}' . Przyjmijmy, że jeżeli miejsce $p \in P$ ma ograniczoną pojemność, to jego miejsce uzupełniające,



Rys. 6.4. Model wzajemnego wykluczania dla problemu czytelników i pisarzy; sieć uogólniona równoważna sieci przedstawionej na rys. 6.2

dodane w sieci \mathcal{N}' , ma oznaczenie p' . Funkcja f jest określona następująco:

$$\begin{aligned} f: \mathcal{R}(M_0) &\rightarrow \mathcal{R}(M'_0) \wedge \\ (f(M) = M') &\Rightarrow (\forall p \in P: M'(p) = M(p) \wedge \\ \forall p' \in P' - P: M'(p') &= K(p) - M(p)). \end{aligned} \quad (6.9)$$

Prawdziwe są następujące wnioski wynikające z przedstawionego algorytmu konstruowania sieci \mathcal{N}' .

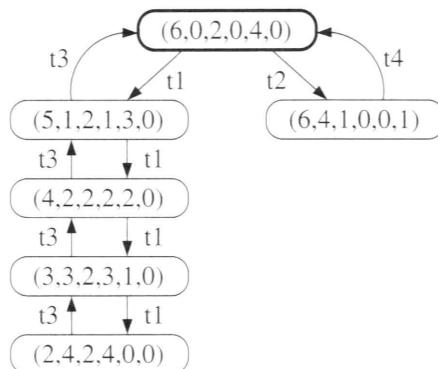
Wniosek 6.1. Jeżeli miejsce $p \in P$ ma ograniczoną pojemność, to miejsca $p, p' \in P'$ są $K(p)$ -ograniczone. Jeżeli miejsce $p \in P$ ma nieograniczoną pojemność, to miejsce to jest ograniczone wtedy i tylko wtedy, gdy miejsce $p \in P'$ jest ograniczone. ■

Wniosek 6.2. Przejście t sieci \mathcal{N} jest aktywne przy znakowaniu $M \in \mathcal{R}(M_0)$ wtedy i tylko wtedy, gdy przejście t sieci \mathcal{N}' jest aktywne przy znakowaniu $f(M) \in \mathcal{R}(M'_0)$, przy czym funkcja f jest określona zgodnie z wzorem (6.9). ■

Wniosek 6.3. Sieci \mathcal{N} i \mathcal{N}' mają identyczne zbiory ciągów przejść, które można wykonać, rozpoczynając od znakowania początkowego, tzn.

$$\mathcal{L}(M_0) = \mathcal{L}(M'_0). \quad (6.10)$$

Ostatni z przedstawionych wniosków jest bezpośrednią konsekwencją wniosku 6.2. Wynika z niego, że obie sieci mają takie same własności zależne od ciągów wykonywanych przejść, tzn. sieci te mają taki sam poziom żywotności, te same własności dotyczące odwracalności sieci czy też jej sprawiedliwości.



Rys. 6.5. Graf osiągalności dla sieci przedstawionej na rys. 6.4

Na rysunku 6.5 przedstawiono graf osiągalności dla sieci występującej na rys. 6.4 (zbiór miejsc P' jest uporządkowany następująco: $P' = \{p_1, p_2, p_3, p_1', p_2', p_3'\}$). Jak sieć N na rys. 6.2, tak i rozważana sieć N' jest żywa, żywa ze względu na miejsca, odwracalna, ale nie jest siecią sprawiedliwą. Sieć ta jest ograniczona. Górnymi ograniczeniami poszczególnych miejsc są odpowiednio liczby 6, 4, 2, 4, 4 i 1.

6.4. Podsumowanie

W tym rozdziale omówiliśmy podstawowe pojęcia dotyczące sieci miejsc i przejść. Są one najczęściej spotykaną w literaturze klasą sieci Petriego. W porównaniu z sieciami uogólnionymi miejsca w PT-sieciach mogą mieć ograniczoną pojemność, co ma istotny wpływ na ich własności dynamiczne. Przejścia w PT-sieciach są aktywne, o ile ich wykonanie nie spowodowałoby przekroczenia pojemności ich miejsc wyjściowych. Sieci uogólnione są szczególnym przypadkiem sieci miejsc i przejść. Wszystkie metody analizy przedstawione w rozdz. 5 mają zastosowanie również w odniesieniu do PT-sieci.

Sieci priorytetowe

Sieci priorytetowe są rozszerzeniem sieci miejsc i przejść (zob. prace: [19], [24], [43], [50] i [55]). Dzięki wprowadzeniu funkcji priorytetów do definicji sieci można określić częściowy porządek w zbiorze przejść, eliminując lub ograniczając sytuacje konfliktowe. Jeżeli aktywne są dwa przejścia, które rywalizują o wspólne znaczniki, to będzie wybrane przejście o wyższym priorytecie. Zastosowanie priorytetów przejść pozwala na dokładniejsze modelowanie systemów współbieżnych, w których często pojawia się na przykład konieczność przypisania priorytetów poszczególnym procesom.

W tym rozdziale zajmiemy się formalną definicją sieci priorytetowych oraz przykładami modelowania z zastosowaniem tych sieci. Będziemy rozważać wyłącznie sieci priorytetowe z nieograniczoną pojemnością miejsc, ale podobnie jak w przypadku PT-sieci możemy wprowadzić do nich funkcję pojemności miejsc. Przykłady innych modyfikacji sieci można znaleźć na przykład w pracy [74].

7.1. Dynamika sieci priorytetowych

W porównaniu z definicją 4.1 sieci uogólnionej definicja sieci priorytetowej zawiera dodatkowo funkcję określającą priorytety przejść.

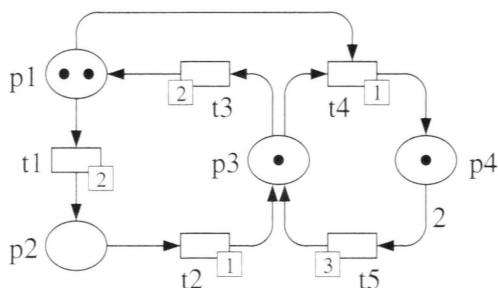
Definicja 7.1. Uporządkowaną szóstkę $\mathcal{N} = (P, T, A, W, I, M_0)$ nazywamy *siecią priorytetową*, jeżeli spełnione są warunki:

- (1) $\mathcal{N} = (P, T, A)$ jest siecią.
- (2) $W: A \rightarrow \mathbb{N}$ jest funkcją wag łuków.
- (3) $I: T \rightarrow \mathbb{Z}_+$ jest funkcją priorytetów przypisującą każdemu z przejść sieci liczbę całkowitą nieujemną, zwaną *priorytetem przejścia*.
- (4) $M_0: P \rightarrow \mathbb{Z}_+$ jest *znakowaniem początkowym* sieci \mathcal{N} . ■

Funkcja I wyznacza w zbiorze przejść sieci częściowy porządek, który redukuje liczbę sytuacji konfliktowych. W sieciach priorytetowych obowiązują takie same definicje jak w sieciach uogólnionych, wszakże z jednym wyjątkiem: przejście, które jest aktywne przy danym znakowaniu sieci, może zostać wykonane tylko wtedy, kiedy nie ma w tej sieci innego aktywnego przejścia o wyższym priorytecie, które pozostawałoby z danym przejściem w konflikcie. Gdy dwa przejścia mają taki sam wysoki priorytet i są aktywne, wówczas wybór przejścia do wykonania jest niedeterministyczny.

Definicja 7.2. Przejście $t \in T$ jest *aktywne* przy znakowaniu M (jest M -aktywne), jeżeli każde z jego miejsc wejściowych zawiera co najmniej tyle znaczników, ile wynosi waga łuku prowadzącego od tego miejsca do przejścia t , oraz priorytet przejścia t jest nie mniejszy niż priorytet dowolnego przejścia $t' \neq t$, z którym przejście t jest w konflikcie. ■

Jeżeli przejście t jest aktywne przy znakowaniu M , to w wyniku jego wykonania powstaje nowe znakowanie M' zdefiniowane tak samo jak dla sieci uogólnionych (zob. wzór (4.1)).



Rys. 7.1. Przykład sieci priorytetowej

Na rysunku 7.1 przedstawiono przykład sieci priorytetowej. Wartości priorytetów występują jako dodatkowe etykiety dołączane do poszczególnych przejść. Przy zadanym znakowaniu początkowym $M_0 = (2, 0, 1, 1)$ aktywne są przejścia t_1 i t_3 . Przejście t_4 nie jest aktywne, ponieważ ma niższy priorytet.

Sieci uogólnione można traktować jako szczególny przypadek sieci priorytetowych, w których wszystkie przejścia mają jednakowe priorytety. Właściwości sieci priorytetowych można analizować tymi samymi metodami, których używa się, badając sieci uogólnione. Wprowadzenie priorytetów pozwala na bardziej *naturalne* modelowanie systemów, w których trzeba korzystać z deterministycznego wyboru,

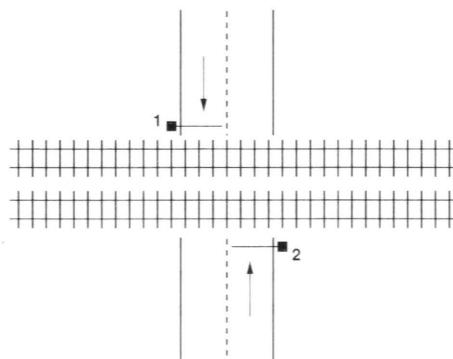
jednak nie pozostaje bez wpływu na własności sieci. Warto zwrócić uwagę na fakt, że graf osiągalności dla sieci priorytetowej będzie zazwyczaj mniejszy niż graf odpowiadającej jej sieci uogólnionej (tj. sieci, która powstaje z sieci priorytetowej przez usunięcie wartości priorytetów). Różnica może dotyczyć zarówno liczby węzłów, jak i łuków grafu. Ze względu na priorytety przejść, w grafie osiągalności nie wystąpią niektóre łuki, a w efekcie mogą nie wystąpić niektóre znakowania.

7.2. Modelowanie z zastosowaniem sieci priorytetowych

W tym podrozdziale omówimy przykłady zastosowania sieci priorytetowych do modelowania systemów współbieżnych. Użycie priorytetów przejść pozwala tworzyć modele dokładniej odzwierciedlające rzeczywisty system. Poniżej analizujemy dwa przykłady sieci priorytetowych, model sterowania rogatkami na przejeździe kolejowym i model wywłaszczenia procesów dla systemu operacyjnego czasu rzeczywistego.

7.2.1. Model przejazdu kolejowego

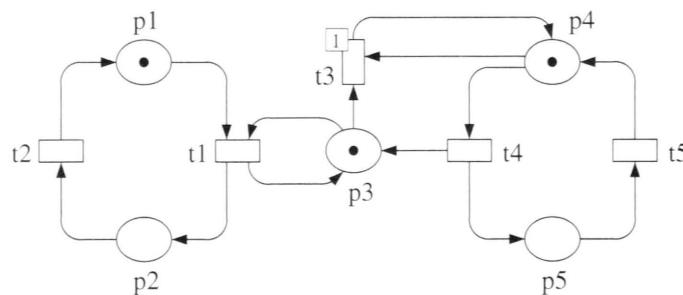
Rozważmy system sterowania rogatkami na przejeździe kolejowym, którego schemat przedstawiono na rys. 7.2. Na drogach dojazdowych umieszczone dwie rogatki oznaczone numerami 1 i 2. Ponieważ powinny one zawsze znajdować się w tym samym stanie (podniesione lub opuszczone), więc będą rozważane jako jeden element systemu i reprezentowane przez jedno miejsce. Znacznik w tym miejscu będzie oznaczał, że rogatki są podniesione i możliwy jest ruch samochodów przez przejazd.



Rys. 7.2. Schemat przejazdu kolejowego

Model rozważanego systemu przedstawiono na rys. 7.3. Oto znaczenie poszczególnych elementów sieci:

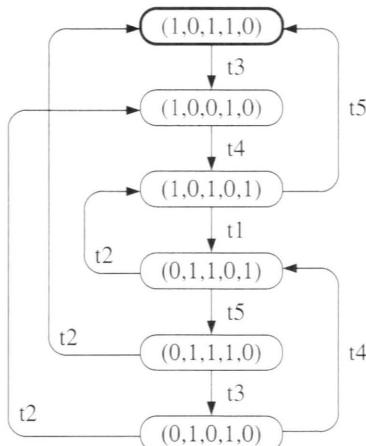
- Miejsce p_1 reprezentuje drogi dojazdowe do przejazdu, a znacznik w tym miejscu – samochód znajdujący się przed przejazdem. Przejazd przez tory jest reprezentowany przez przejście t_1 .
- Miejsce p_3 reprezentuje rogatek. Znacznik w miejscu p_3 oznacza, że rogatek są podniesione. Opuszczenie rogatek jest reprezentowane przez przejście t_3 .
- Miejsce p_4 reprezentuje tory kolejowe, a znacznik w tym miejscu – pociąg zbliżający się do przejazdu kolejowego. Przejazd pociągu i podniesienie rogatek po przejeździe jest reprezentowane przez przejście t_4 .
- Miejsca p_2 i p_5 reprezentują otoczenie. Znaczniki w tych miejscach oznaczają odpowiednio samochód i pociąg po przejechaniu przez przejazd kolejowy.



Rys. 7.3. Model sterowania rogatkami na przejeździe kolejowym

Prezentowany model opisuje zarówno ruch kolejowy, jak i samochodowy. W przypadku ruchu samochodów przejście oznaczające przejazd pojazdu (t_1) testuje obecność znacznika w miejscu p_3 (podniesione rogatek). Pojawienie się pociągu przed przejazdem kolejowym wymusza konieczność opuszczenia rogatek (wykonanie przejścia t_3). Przejście t_3 ma najwyższy priorytet w modelu (pozostałe przejścia mają przypisane priorytety równe 0). Użycie priorytetów przejść umożliwia przypisanie pociągowi pierwszeństwa i wymuszenie opuszczenia rogatek bez względu na natężenie ruchu samochodowego. Brak priorytetów przejść mógłby spowodować sytuację konfliktową między przejściami t_1 i t_3 , gdyby zaś w modelu było więcej znaczników oznaczających samochody, wówczas mogłoby dojść do zagłodzenia przejścia t_3 .

Do analizy sieci priorytetowych można stosować metody przedstawione w rozdz. 5, zwłaszcza grafy osiągalności (pokrycia) oraz niezmienniki sieci. Na rysunku 7.4 przedstawiono graf osiągalności dla rozważanej sieci. Warto zwrócić uwagę, że przy znakowaniu początkowym $M_0 = (1, 0, 1, 1, 0)$ aktywne jest tylko przejście t_3 . Pomimo że dla przejść t_1 i t_4 istnieje odpowiednia liczba znaczników w ich miejscowościach wejściowych, nie są one aktywne ze względu na niższe priorytety tych przejść w porównaniu z priorytetem przejścia t_3 .



Rys. 7.4. Graf osiągalności dla sieci modelującej sterowanie rogatkami na przejeździe kolejowym

Rozważana sieć jest żywa, bezpieczna i żywa ze względu na miejsca. Graf osiągalności sieci jest grafem silnie spójnym, jest ona zatem odwracalna. Sieć nie jest sprawiedliwa. W zbiorze $\mathcal{L}(M_0)$ można wskazać sprawiedliwe ciągi przejść, ale własność ta nie dotyczy wszystkich ciągów, na przykład nie jest sprawiedliwy ciąg przejść $\alpha = t_3, t_4, t_1, t_2, t_1, t_2 \dots$

Macierz incydencji rozważanej sieci to:

$$N = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 1 & -1 \end{pmatrix}. \quad (7.1)$$

Rozwiązywanie układu równań (5.26) jest następujące:

$$x = \begin{pmatrix} \alpha_1 \\ \alpha_1 \\ 0 \\ \alpha_2 \\ \alpha_2 \end{pmatrix}. \quad (7.2)$$

Przykładami niezmienników miejsc są wektory $I_1 = (1, 1, 0, 0, 0)$, $I_2 = (0, 0, 0, 1, 1)$ oraz $I_3 = I_1 + I_2 = (1, 1, 0, 1, 1)$. Rozważana sieć nie jest pokryta niezmiennikami miejsc (mimo że jest ograniczona) i nie jest zachowawcza. Sieć jest częściowo zachowawcza dla zbioru miejsc będącego nośnikiem niezmiennika I_3 .

Rozwiązywanie układu równań (5.63) jest następujące:

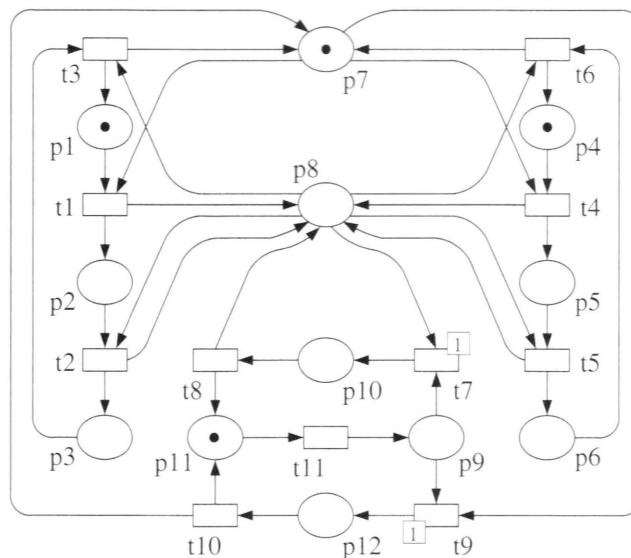
$$x = \begin{pmatrix} \alpha_1 \\ \alpha_1 \\ \alpha_2 \\ \alpha_2 \\ \alpha_2 \end{pmatrix}. \quad (7.3)$$

Przykładami niezmienników przejść są wektory $I_1 = (1, 1, 0, 0, 0)$ oraz $I_2 = (0, 0, 1, 1, 1)$. Oba niezmienniki są realizowalne. Sieć jest pokryta niezmiennikami przejść. Niezmiennikiem pokrywającym sieć jest wektor $I_3 = I_1 + I_2$.

7.2.2. Model wywłaszczenia procesów dla systemu operacyjnego

Rozważmy uproszczony model systemu operacyjnego czasu rzeczywistego. Przyjmijmy, że procesor jest zasobem współdzielonym przez dwa procesy. Ponadto może zdarzyć się wywłaszczenie procesora przez przerwanie. Aktualnie wykonywane zadanie zostanie wówczas zawieszone na czas obsługi przerwania.

Model rozważanego systemu przedstawiono na rys. 7.5. Występują w nim dwa procesy reprezentowane odpowiednio przez podsieci wyznaczone przez zbiory miejsc $\{p1, p2, p3\}$ i $\{p4, p5, p6\}$. Każdy proces znajduje się albo w stanie oczekiwania na przydzielenie procesora, albo w stanie posiadania do niego dostępu. W drugim przypadku podzielono obliczenia na części, między którymi proces może zostać zawieszony, jeżeli było zgłoszone przerwanie. Znaczenie elementów sieci powiązanych z procesami jest następujące:



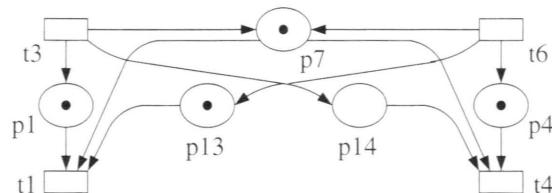
Rys. 7.5. Model wywieszczania procesów dla systemu operacyjnego

- Znaczniki w miejscach p_1 i p_4 reprezentują procesy oczekujące na przydzielanie im procesora. Przejście procesora przez proces jest reprezentowane przez przejścia t_1 i t_4 , zwolnienie procesora – przez przejścia t_3 i t_6 .
- Znaczniki w miejscach p_2 , p_3 , p_5 i p_6 reprezentują procesy będące w stanie posiadania dostępu do procesora. Jeżeli dodatkowo w miejscu p_8 znajduje się znacznik, to proces wykonuje obliczenia, a w przeciwnym przypadku jest zawieszony.
- Miejsce p_7 służy do zapewnienia wzajemnego wykluczania między procesami w dostępie do procesora. Znacznik w tym miejscu reprezentuje wolny procesor, brak znacznika oznacza, że procesor jest zajęty przez jeden z dwóch procesów lub przez proces obsługi przerwania.
- Znacznik w miejscu p_8 oznacza, że procesor jest zajęty przez jeden z procesów. Przy przejściu do kolejnej fazy obliczeń (przejścia t_2 i t_5 lub przejścia związane ze zwolnieniem procesora t_3 i t_6), proces testuje obecność znacznika w tym miejscu. Brak znacznika (efekt wywieszczania procesora przez proces obsługi przerwania) powoduje zawieszenie procesu.

Podsieć wyznaczona przez zbiór miejsc $\{p_9, p_{10}, p_{11}, p_{12}\}$ reprezentuje część systemu związaną ze zgłoszaniem i obsługą przerwań. Znaczenie poszczególnych elementów tej podsieci jest następujące:

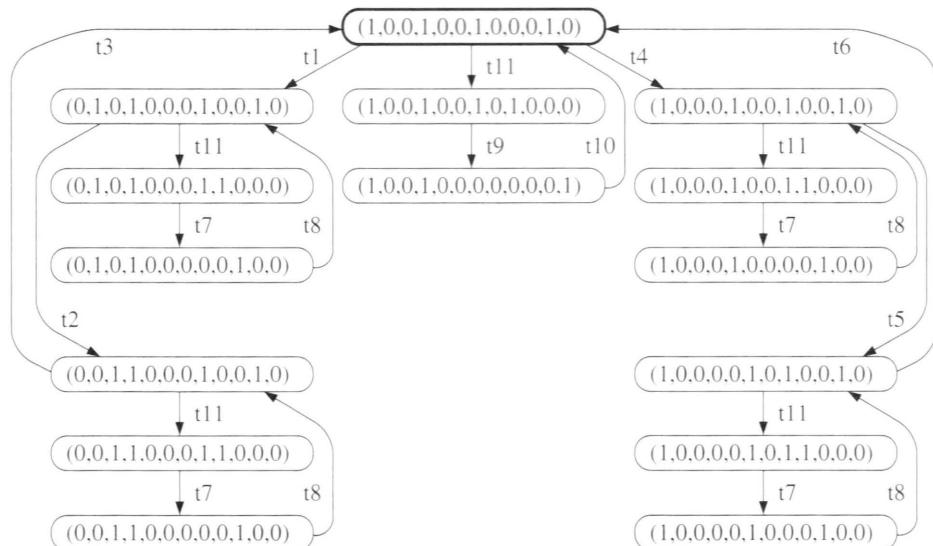
- Znacznik w miejscu p_{11} oznacza, że system jest gotowy do obsługi przerwania. Zgłoszenie przerwania jest reprezentowane przez przejście t_{11} .
- Znacznik w miejscu p_9 oznacza, że zgłoszono przerwanie i system powinien rozpoczęć procedurę jego obsługi. Rozpoczęcie obsługi przerwania reprezentowane jest przez przejścia t_7 i t_9 . Przejście t_7 jest wykonywane wówczas, gdy procesor jest zajęty przez jeden z procesów, a jego wykonanie powoduje zawieszenie tego procesu. Przejście t_9 jest wykonywane wtedy, kiedy procesor nie jest zajęty, a jego wykonanie blokuje możliwość przejęcia procesora przez którykolwiek z procesów aż do czasu zakończenia obsługi przerwania.
- Znacznik w miejscu p_{10} lub p_{12} oznacza, że procesor jest zajęty przez proces obsługi przerwania. Zwolnienie procesora jest reprezentowane przez przejścia t_8 i t_{10} . W zależności od sytuacji, w której rozpoczęto obsługę przerwania, albo będzie odwieszony proces, w posiadaniu którego był procesor (przejście t_8), albo będzie usunięta blokada dostępu do procesora (przejście t_{10}).

Zastosowanie priorytetów w prezentowanym modelu uniemożliwia wystąpienie efektu zagłodzenia procesu obsługi przerwań (przejścia t_7 i t_9) przez inne przejścia sieci. Pomijając obsługę przerwań, istnieje możliwość zagłodzenia jednego procesu przez drugi, gdyż proces może ponowne przejąć procesor bezpośrednio po jego zwolnieniu. Aby zapewnić naprzemienne udostępnianie procesora dwóm rozważanym procesom, należało dodać konstrukcję przedstawioną na rys. 7.6.



Rys. 7.6. Konstrukcja gwarantująca przemienność w dostępie do procesora

Na rysunku 7.7 przedstawiono graf osiągalności dla rozważanej sieci. Jest ona żywa, bezpieczna, żywa ze względu na miejsca i odwracalna. Sieć nie jest sprawiedliwa. W zbiorze $\mathcal{L}(M_0)$ można wskazać ciągi przejść, które nie są sprawiedliwe, na przykład: $\alpha = t_{11}, t_9, t_{10}, t_{11}, t_9, t_{10} \dots$



Rys. 7.7. Graf osiągalności dla sieci modelującej wywłaszczenie procesów w systemie operacyjnym

Oto macierz incydencji rozważanej sieci:

$$N = \begin{pmatrix} -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & -1 & 0 & 1 & 0 & 0 & -1 & 1 & 0 \\ 1 & 0 & -1 & 1 & 0 & -1 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \end{pmatrix} \quad (7.4)$$

Rozwiązaniem ogólnym układu równań (5.26) jest wektor:

$$x = \begin{pmatrix} \alpha_1 + \alpha_3 - \alpha_5 \\ \alpha_1 \\ \alpha_1 \\ \alpha_2 + \alpha_3 - \alpha_5 \\ \alpha_2 \\ \alpha_2 \\ \alpha_5 - \alpha_4 \\ \alpha_3 - \alpha_4 \\ \alpha_4 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \end{pmatrix}. \quad (7.5)$$

Przykładami niezmienników miejsc są wektory:

- $I_1 = (1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0)$,
- $I_2 = (0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0)$,
- $I_3 = (0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)$,
- $I_4 = (0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0)$,
- $I_5 = (1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0)$.

Pierwsze trzy niezmienniki wyznaczają zachowawcze podsieci, reprezentujące odpowiednio lewy i prawy proces oraz proces obsługi przerwań. Każda z tych podsieci zawiera zawsze dokładnie jeden znacznik, który w zależności od miejsca, w którym się znajduje, reprezentuje różne stany danego procesu. Na podstawie niezmiennika I_4 można wysunąć wniosek, że w dowolnym stanie systemu zachodzi dokładnie jedna z czterech sytuacji: procesor jest wolny, procesor jest zajęty przez proces reprezentowany po lewej stronie na rys. 7.5, procesor jest zajęty przez proces reprezentowany po prawej stronie na rys. 7.5 albo procesor jest zajęty przez proces obsługi przerwań.

Rozważana sieć jest pokryta niezmiennikami miejsc. Sieć jest zachowawcza względem wektora wag wyznaczonego przez niezmiennik $I_6 = I_3 + I_4 + I_5 = (1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2)$.

Rozwiązywanie ogólne układu równań (5.63) jest wektorem:

$$x = \begin{pmatrix} \alpha_1 \\ \alpha_1 \\ \alpha_1 \\ \alpha_2 \\ \alpha_2 \\ \alpha_2 \\ \alpha_2 \\ \alpha_4 - \alpha_3 \\ \alpha_4 - \alpha_3 \\ \alpha_3 \\ \alpha_3 \\ \alpha_3 \\ \alpha_4 \end{pmatrix}. \quad (7.6)$$

Przykładami niezmienników przejść są wektory:

- $I_1 = (1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0)$,
- $I_2 = (0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0)$,
- $I_3 = (0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0)$,
- $I_4 = (0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1)$.

Wszystkie niezmienniki są realizowalne. Sieć jest pokryta niezmiennikami przejść. Wskazane niezmienniki przejść odpowiadają niektórym cyklom w grafie osiągalności sieci. Niezmienniki I_1 i I_2 wyznaczają cykle pracy procesów odpowiednio po lewej i prawej stronie na rys. 7.5, a niezmienniki I_3 i I_4 – dwa możliwe cykle pracy procesu obsługi przerwań (z wywłaszczeniem i bez wywłaszczenia procesów).

7.3. Podsumowanie

W tym rozdziale omówiliśmy podstawowe pojęcia dotyczące sieci priorytetowych, które są jednym z wielu rozszerzeń sieci miejsc i przejść (lub sieci uogólnionych). Podstawowym elementem tego rozszerzenia jest wprowadzenie funkcji priorytetów przejść. W efekcie zostaje zadany częściowy porządek w zbiorze przejść, dzięki czemu można wyeliminować lub znacznie ograniczyć występowanie sytuacji konfliktowych. Jeżeli aktywne są dwa przejścia, które rywalizują o wspólne znaczniki, to wybiera się przejście o wyższym priorytecie. Jednocześnie zastosowanie priorytetów umożliwia wierniejsze modelowanie rzeczywistych systemów współbieżnych.



Sieci czasowe

Wprowadzone w rozdz. 4 sieci uogólnione są jedną z najczęściej spotykanych w literaturze klas sieci Petriego, przy czym występują one zazwyczaj pod nazwą sieci miejsc i przejść (PT-sieci). Są one traktowane jako formalny język używany do modelowania systemów współbieżnych. Najważniejsze cechy tej klasy to duża różnorodność metod analizy ich własności i łatwość ich stosowania. Sieci uogólnione mogą również posłużyć do opisu systemów z zależnościami czasowymi. Zastosowania te są jednak znacznie ograniczone, gdyż zależności tych nie można opisywać w sposób ilościowy.

Pierwsze rozszerzenia czasowe sieci Petriego zostały wprowadzone przez Merlinia (zob. prace: [46], [47]) i Ramchandaniego (zob. pracę [54]). Obaj autorzy połączyli ograniczenia czasowe z przejściami, ale zdefiniowali różne semantyki, dzięki czemu zachowanie tych sieci było interpretowane w odmienny sposób. W 1983 roku Coolahan i Roussopoulos zaproponowali w pracy [15] czasowe rozszerzenie sieci Petriego, w którym ograniczenia czasowe połączono z miejscami. W tym samym roku Walter wprowadził (zob. pracę [88]) rozszerzenie czasowe, w którym ograniczenia czasowe przypisano do łuków, a dodatkowo do znaczników sieci dołączono tzw. *pieczętki czasowe*. Te trzy główne rozszerzenia sieci Petriego były później rozwijane przez wielu autorów. Modele czasowe wprowadzono również do sieci wysokiego poziomu. Przykłady takich rozszerzeń będą omówione w rozdz. 10 i 13. Więcej informacji na temat różnych rozszerzeń czasowych sieci Petriego można znaleźć w pracach: [8], [59] i [60].

W tym rozdziale omówimy dwie klasy sieci czasowych, w których ograniczenia czasowe są powiązane z przejściami sieci. *Proste sieci czasowe* (zob. pracę [9]) zdefiniowano jako rozszerzenie sieci znakowanych, a sieci przedziałowe (zob. pracę [47]), jako rozszerzenie sieci uogólnionych. W obu przypadkach definicja została wz bogaca ona o funkcję opóźnień określony na zbiorze przejść. Dla prostych sieci czasowych opóźnienie wykonania przejścia jest nieujemną liczbą wymierną określającą czas wykonania tego przejścia liczony od chwili, w której było ono uaktywnione. W przypadku sieci przedziałowych podaje się przedział liczb wy-

miernych określający minimalne i maksymalne opóźnienie wykonania przejścia, przy czym opóźnienie maksymalne nie musi być liczbą skońzoną.

8.1. Proste sieci czasowe

W prostych sieciach czasowych każde przejście ma przypisaną nieujemną liczbę wymierną, która jest interpretowana jako czas jego wykonania.

Definicja 8.1. Uporządkowaną piątkę $\mathcal{N} = (P, T, A, M_0, \sigma)$ nazywamy *prostą siecią czasową*, jeżeli spełnione są warunki:

- (1) $\mathcal{N} = (P, T, A)$ jest siecią.
- (2) $M_0: P \rightarrow \mathbb{Z}_+$ jest *znakowaniem początkowym* sieci \mathcal{N} .
- (3) $\sigma: T \rightarrow \mathbb{Q}_+$ jest *funkcją opóźnień*, przypisującą każdemu przejściu sieci liczbę wymierną nieujemną $\sigma(t)$, zwaną *opóźnieniem statycznym* przejścia t . ■

Funkcja opóźnień zmienia zachowanie prostych sieci czasowych w porównaniu z sieciami uogólnionymi w sposób następujący. Jeżeli przejście t staje się aktywne (tzn. wszystkie jego miejsca wejściowe zawierają znaczniki), to musi zostać wykonane po dokładnie $\sigma(t)$ jednostkach czasu, chyba że przed upływem tego czasu przestanie być aktywne w wyniku wykonania innego przejścia. Sieci znakowane mogą być traktowane jako szczególny przypadek prostych sieci czasowych z funkcją σ , która przypisuje każdemu przejściu zerowe opóźnienie.

Opóźnienie statyczne σ jest niezależne od wartości *globalnego zegara*. Dla dowolnego przejścia $t \in T$ możemy zdefiniować również *opóźnienie dynamiczne* $\delta(t)$, które zależy od aktualnej wartości zegara. Jeżeli przejście t jest aktywne, to opóźnienie dynamiczne określa, ile czasu pozostało do wykonania przejścia t . Jeżeli przejście t nie jest aktywne przy danym znakowaniu, to przyjmujemy $\delta(t) = -$ (symbol $-$ będzie oznaczał brak wartości). Jeżeli przyjmiemy, że zbiór przejść sieci jest zbiorem uporządkowanym, to opóźnienia statyczne i dynamiczne możemy zapisywać w postaci wektorów σ i δ . Między opóźnieniem statycznym i dynamicznym zachodzi następujący związek (dla przejść aktywnych):

$$0 \leq \delta(t) \leq \sigma(t). \quad (8.1)$$

Niech dana będzie prosta sieć czasowa $\mathcal{N} = (P, T, A, M_0, \sigma)$. Przyjmijmy ponadto założenie, że żadne przejście w sieci \mathcal{N} nigdy nie jest aktywne współbieżnie ze

sobą. Innymi słowy, nie istnieje przejście $t \in T$ i znakowanie M osiągalne ze znakowania początkowego, przy którym wszystkie miejsca wejściowe przejścia t zawierają więcej niż jeden znacznik.

Definicja 8.2. Stanem sieci N nazywamy parę (M, δ) , przy czym M jest znakowaniem, a δ jest wektorem opóźnień dynamicznych.

Stanem początkowym nazywamy parę (M_0, δ_0) , przy czym M_0 jest znakowaniem początkowym oraz $\delta_0(t) = \sigma(t)$, jeżeli przejście t jest aktywne przy znakowaniu początkowym i $\delta_0(t) = -$ w przeciwnym przypadku. ■

Aktywność przejścia t przy zadanym znakowaniu nie gwarantuje jego wykonania. Jeżeli w sieci istnieje przejście, które ma mniejsze opóźnienie dynamiczne niż t , to może spowodować utratę aktywności przejścia t , zanim będzie ono mogło być wykonane.

Definicja 8.3. Przejście $t \in T$ jest *wykonalne* w stanie (M, δ) , jeżeli jest aktywne przy znakowaniu M i ma minimalne opóźnienie dynamiczne spośród wszystkich przejść aktywnych przy tym znakowaniu. ■

Jeżeli przejście t jest wykonalne w stanie (M, δ) , to nowy stan (M', δ') , uzyskany w wyniku wykonania przejścia t , jest określony w sposób następujący:

$$M'(p) = \begin{cases} M(p) - 1 & \text{gdy } p \in In(t) - Out(t), \\ M(p) + 1 & \text{gdy } p \in Out(t) - In(t), \\ M(p) & \text{w pozostałych przypadkach;} \end{cases} \quad (8.2)$$

$$\delta'(t_i) = \begin{cases} \sigma(t_i) & \text{w przypadku (1),} \\ \delta(t_i) - \delta(t) & \text{w przypadku (2),} \\ - & \text{w przypadku (3).} \end{cases} \quad (8.3)$$

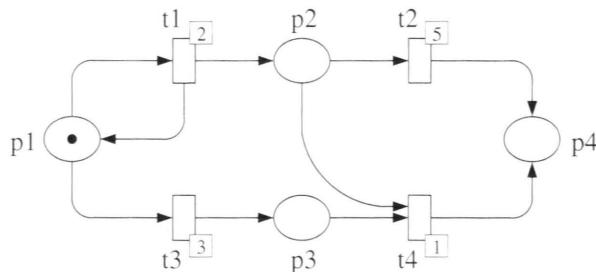
- (1) Przejście t_i jest aktywne przy znakowaniu M' , ale nie było aktywne przy znakowaniu M lub $t_i = t$ i jest ponownie aktywne;
- (2) Przejście $t_i \neq t$ było aktywne przy znakowaniu M i pozostaje aktywne przy znakowaniu M' ;
- (3) Przejście t_i nie jest aktywne przy znakowaniu M' .

Jeżeli przejście t jest wykonalne w stanie (M, δ) i w wyniku jego wykonania otrzymujemy stan (M', δ') , to fakt ten zapisujemy następująco:

$$(M, \delta) \xrightarrow{t} (M', \delta'). \quad (8.4)$$

Dowolny stan sieci \mathcal{N} , który możemy otrzymać ze stanu (M, δ) po wykonaniu skończonego ciągu przejść, nazywamy *stanem osiągalnym* ze stanu (M, δ) . Zbiór wszystkich stanów osiągalnych ze stanu (M, δ) oznaczamy symbolem $\mathcal{R}(M, \delta)$. Wyróżniamy zbiór stanów osiągalnych ze stanu początkowego, który oznaczamy symbolem $\mathcal{R}(M_0, \delta_0)$. Z każdym stanem (M', δ') osiągalnym ze stanu (M, δ) jest związany ciąg przejść, których wykonanie prowadzi od stanu (M, δ) do (M', δ') . Zbiór wszystkich ciągów przejść, które można wykonać, począwszy od stanu (M, δ) , będziemy oznaczać symbolem $\mathcal{L}(M, \delta)$. Zbiór $\mathcal{L}(M, \delta)$ zawiera zarówno skończone, jak i nieskończonościowe ciągi przejść. Dla sieci czasowych możemy mówić również o znakowaniach osiągalnych i w takim przypadku będziemy używać oznaczenia $\mathcal{R}(M)$.

Jeżeli w stanie (M, δ) są wykonalne dwa przejścia, ale wykonanie dowolnego z nich powoduje utratę aktywności przez drugie, to mówimy, że przejścia te pozostają ze sobą w *konflikcie dynamicznym* (lub po prostu w *konflikcie*). W takim przypadku wybór przejścia, które zostanie wykonane, jest wyborem niedeterministycznym.



Rys. 8.1. Prosta sieć czasowa

Na rysunku 8.1 przedstawiono przykład prostej sieci czasowej. Opóźnienia statyczne przypisane przejściom sieci są reprezentowane przez dodatkowe etykiety, umieszczone przy poszczególnych przejściach. Stan początkowy rozważanej sieci (M_0, δ_0) jest określony następująco: $M_0 = (1, 0, 0, 0)$, $\delta_0 = (2, -, 3, -)$. W stanie początkowym aktywne są przejścia $t1$ i $t3$, ale wykonalne jest tylko przejście $t1$; jego wykonanie prowadzi do stanu (M_1, δ_1) , przy czym $M_1 = (1, 1, 0, 0)$ i $\delta_1 = (2, 5, 1, -)$. W nowym stanie aktywne są trzy przejścia, ale wykonalne jest tylko przejście $t3$, prowadzące do kolejnego stanu (M_2, δ_2) , przy czym $M_2 = (0, 1, 1, 0)$ i $\delta_2 = (-, 4, -, 1)$. W stanie (M_2, δ_2) jest wykonalne tylko przejście $t4$. Wykonanie tego przejścia prowadzi do ostatniego już stanu (M_3, δ_3) , przy czym $M_3 = (0, 0, 0, 1)$ i $\delta_3 = (-, -, -, -)$, a znakowanie M_3 jest znakowaniem martwym.

Stan (M, δ) prostej sieci czasowej \mathcal{N} , taki że znakowanie M jest znakowaniem martwym nazywamy *stanem martwym*.

Rozważmy skończony ciąg przejść $\alpha = t_1, t_2, \dots, t_n$ prowadzących od stanu (M_1, δ_1) , taki że:

$$(M_1, \delta_1) \xrightarrow{t_1} (M_2, \delta_2) \xrightarrow{t_2} \cdots \xrightarrow{t_n} (M_{n+1}, \delta_{n+1}). \quad (8.5)$$

Definicja 8.4. Czasem wykonania ciągu przejść α nazywamy sumę dynamicznych opóźnień przejść t_i w stanach (M_i, δ_i) , którą oznaczamy symbolem $\delta(\alpha)$, tzn:

$$\delta(\alpha) = \sum_{i=1}^n \delta_i(t_i). \quad (8.6)$$

■

Definicja 8.5. Niech dane będą stany $(M, \delta), (M', \delta')$, takie że $(M', \delta') \in \mathcal{R}(M, \delta)$. Czasem przejścia od stanu (M, δ) do (M', δ') nazywamy czas wykonania dowolnego ciągu przejść α prowadzącego od stanu (M, δ) do (M', δ') . Czas przejścia oznaczamy symbolem $\delta((M, \delta), (M', \delta'))$.

Czas wykonania ciągu przejść α jest określony jednoznacznie, ale nie można tego powiedzieć o czasie przejścia między stanami. Jeżeli istnieje wiele ciągów przejść prowadzących od stanu (M, δ) do (M', δ') , to otrzymujemy również wiele potencjalnie różnych czasów przejść. Najczęściej poszukuje się minimalnego i maksymalnego czasu przejścia między stanami.

Rozważmy ponownie przykład sieci czasowej przedstawionej na rys. 8.1. Dla ciągu przejść $\alpha = t1, t3, t4$, takiego że:

$$(M_0, \delta_0) \xrightarrow{t1} (M_1, \delta_1) \xrightarrow{t3} (M_2, \delta_2) \xrightarrow{t4} (M_3, \delta_3), \quad (8.7)$$

czas wykonania tego ciągu obliczamy następująco: $\delta(\alpha) = 2 + 1 + 1 = 4$. Ponieważ istnieje tylko jeden ciąg przejść od stanu (M_0, δ_0) do (M_3, δ_3) , więc czas $\delta(\alpha)$ jest jednocześnie minimalnym i maksymalnym czasem przejścia między tymi stanami.

Przyjęliśmy powyżej założenie, że w rozważanych sieciach nie występują przejścia współbieżne ze sobą. Jeżeli pominiemy to ograniczenie, to gdy pewne przejście t jest w stanie (M, δ) wpółbieżne ze sobą, wówczas każdą z jego aktywności należy analizować indywidualnie. W takiej sytuacji wpółrzędna wektora δ odpowiadająca przejściu t nie będzie pojedynczą liczbą, ale listą opóźnień dynamicznych, przechowującą po jednej wartości dla każdej aktywności przejścia t .

□ 8.2. Przedziałowe sieci czasowe

W prostych sieciach czasowych czas wykonania przejścia t ma stałą wartość. Dla *sieci przedziałowych* (często nazywanych krótko *sieciami czasowymi*) wartość ta nie jest stała, ale jest ograniczona do pewnego (być może prawostronne nieskończonego) przedziału. Sieci przedziałowe zdefiniowano jako rozszerzenie sieci uogólnionych. Są one bardziej wyraziste niż proste sieci czasowe, ale analiza własności sieci przedziałowych jest znacznie trudniejsza.

Definicja 8.6. Uporządkowaną szóstkę $\mathcal{N} = (P, T, A, W, M_0, \sigma)$ nazywamy *przedziałową siecią czasową*, jeżeli są spełnione warunki:

- (1) $\mathcal{N} = (P, T, A)$ jest siecią.
- (2) $W: A \rightarrow \mathbb{N}$ jest funkcją wag łuków.
- (3) $M_0: P \rightarrow \mathbb{Z}_+$ jest znakowaniem początkowym sieci \mathcal{N} .
- (4) $\sigma: T \rightarrow \mathbb{Q}_+ \times (\mathbb{Q}_+ \cup \{\infty\})$ jest funkcją opóźnień, przypisującą każdemu z przejść parę nieujemnych liczb wymiernych, takich że: $\sigma_{\min}(t) \leq \sigma_{\max}(t)$. Liczby $\sigma_{\min}(t)$ i $\sigma_{\max}(t)$ nazywamy odpowiednio *minimalnym* i *maksymalnym opóźnieniem statycznym* przejścia t . ■

Funkcjonowanie przedziałowych czasowych sieci Petriego można nieformalnie opisać następująco. Każde z przejść ma przypisane dwie wartości $\sigma_{\min}(t)$ i $\sigma_{\max}(t)$. Przyjmijmy, że przejście t od chwili jego uaktywnienia pozostaje już przez cały czas aktywne. Wartości $\sigma_{\min}(t)$ i $\sigma_{\max}(t)$ interpretujemy następująco: $\sigma_{\min}(t)$ określa minimalny czas, który musi upływać od chwili aktywowania przejścia t do jego wykonania, a $\sigma_{\max}(t)$ określa maksymalny czas, w którym przejście t może czekać na wykonanie.

Minimalne i maksymalne opóźnienia statyczne są niezależne od wartości *globalnego zegara*. Opóźnienie dynamiczne δ zależy od aktualnej wartości zegara. Jeżeli przejście t jest aktywne, to opóźnienie dynamiczne określa, ile co najmniej oraz ile co najwyżej pozostało czasu do wykonania przejścia t . Jak opóźnienie statyczne, tak i opóźnienie dynamiczne przypisuje każdemu z aktywnych przejść parę nieujemnych liczb wymiernych $(\delta_{\min}(t), \delta_{\max}(t))$ (druga z tych wartości może być liczbą niewłaściwą), takich że: $\delta_{\min}(t) \leq \delta_{\max}(t)$. Jeżeli przejście t nie jest aktywne przy danym znakowaniu, to przyjmujemy $\delta(t) = -$ (symbol $-$ będzie oznaczał brak wartości).

Jeżeli przyjmiemy, że zbiór przejść sieci jest zbiorem uporządkowanym, to opóźnienia statyczne i dynamiczne możemy zapisywać w postaci wektorów σ i δ (każdy

współczynnik tych wektorów jest parą liczb wymiernych lub symbolem $-$, drugi element pary może być równy ∞). Między opóźnieniem statycznym i dynamicznym zachodzi następujący związek (dla przejść aktywnych):

$$0 \leq \delta_{\min}(t) \leq \sigma_{\min}(t) \wedge 0 \leq \delta_{\max}(t) \leq \sigma_{\max}(t). \quad (8.8)$$

Niech dana będzie sieć przedziałowa $\mathcal{N} = (P, T, A, W, M_0, \sigma)$.

Definicja 8.7. Stanem sieci \mathcal{N} nazywamy parę (M, δ) , przy czym M jest znakowaniem sieci \mathcal{N} , a δ jest wektorem opóźnień dynamicznych.

Stanem początkowym nazywamy parę (M_0, δ_0) , przy czym M_0 jest znakowaniem początkowym oraz $\delta_0(t) = \sigma_0(t)$, jeżeli przejście t jest aktywne przy znakowaniu początkowym i $\delta_0(t) = -$ w przeciwnym przypadku. ■

Podobnie jak w sieciach prostych, aktywność przejścia t przy zadanym znakowaniu nie gwarantuje jego wykonania, może ono bowiem utracić aktywność w wyniku wykonania innego przejścia.

Definicja 8.8. Niech (M, δ) będzie stanem sieci \mathcal{N} , który został osiągnięty przy wartości τ zegara globalnego. Przejście $t \in T$ jest *wykonane* w stanie (M, δ) w czasie $\tau + \theta$, jeżeli jest aktywne przy znakowaniu M oraz:

$$\delta_{\min}(t) \leq \theta \leq \min_{t_i \in T} \delta_{\max}(t_i), \quad (8.9)$$

przy czym $\min_{t_i \in T} \delta_{\max}(t_i)$ oznacza minimum z wartości $\delta_{\max}(t_i)$ przejść aktywnych przy znakowaniu M (podczas wyznaczania minimum pomijamy przejścia nieaktywne przy znakowaniu M , tzn. przejścia, dla których $\delta(t_i) = -$). ■

Jeżeli przejście t jest wykonane w stanie (M, δ) w czasie $\tau + \theta$, to nowy stan (M', δ') uzyskany w wyniku wykonania przejścia t w czasie $\tau + \theta$ jest określony w sposób następujący:

$$M'(p) = \begin{cases} M(p) - W(p, t) & \text{gdy } p \in In(t) - Out(t), \\ M(p) + W(t, p) & \text{gdy } p \in Out(t) - In(t), \\ M(p) - W(p, t) + W(t, p) & \text{gdy } p \in In(t) \cap Out(t), \\ M(p) & \text{w pozostałych przypadkach;} \end{cases} \quad (8.10)$$

$$\delta'(t_i) = \begin{cases} \sigma(t_i) & \text{w przypadku (1),} \\ (\max(0, \delta_{\min}(t_i) - \theta), \delta_{\max}(t_i) - \theta) & \text{w przypadku (2),} \\ - & \text{w przypadku (3).} \end{cases} \quad (8.11)$$

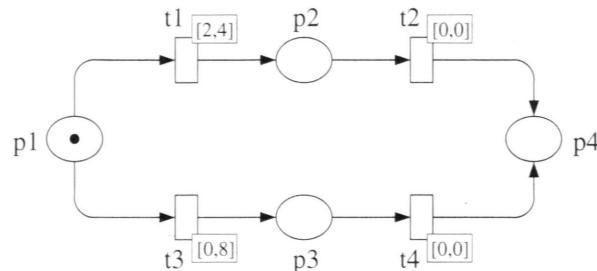
- (1) Przejście t_i jest aktywne przy znakowaniu M' , ale:
 - nie było aktywne przy znakowaniu M ,
 - lub było aktywne przy znakowaniu M , ale nie jest aktywne przy znakowaniu $M - t^-$,
 - lub $t_i = t$ i jest ono ponownie aktywne.
- (2) Przejście $t_i \neq t$ było aktywne przy znakowaniu M i jest aktywne przy znakowaniu $M - t^-$.
- (3) Przejście t_i nie jest aktywne przy znakowaniu M' .

Jeżeli przejście t jest wykonalne w stanie (M, δ) w czasie $\tau + \theta$ i w wyniku jego wykonania otrzymujemy stan (M', δ') , to fakt ten zapisujemy następująco:

$$(M, \delta) \xrightarrow{(t, \theta)} (M', \delta'). \quad (8.12)$$

Konflikt dynamiczny między przejściami definiuje się analogicznie jak w prostych sieciach czasowych.

W sieciach przedziałowych zachowują dotychczasowe znaczenie terminy: stan osiągalny, zbiór stanów osiągalnych, ciąg przejść, zbiór ciągów przejść, a także oznaczenia $\mathcal{R}(M, \delta)$ i $\mathcal{L}(M, \delta)$. Łatwo zauważyć, że w ogólnym przypadku zbiór $\mathcal{R}(M_0, \delta_0)$ jest zbiorem nieskończonym. Przejście t aktywne w stanie (M, δ) może zostać wykonane w dowolnym czasie między $\delta_{\min}(t)$ i $\min_{t_i \in T} \delta_{\max}(t_i)$; oznacza to, że stan taki może mieć nieskończenie wiele stanów bezpośrednio z niego osiągalnych.



Rys. 8.2. Przedziałowa sieć czasowa

Na rysunku 8.2 przedstawiono przykład przedziałowej sieci czasowej. Przy znakowaniu początkowym aktywne są przejścia $t1$ i $t3$. Stan początkowy (M_0, δ_0) jest określony następująco: $M_0 = (1, 0, 0, 0)$, $\delta_0 = ([2, 4], -, [0, 8], -)$ (przyjmujemy początkową wartość zegara globalnego $\tau = 0$). Przejście $t1$ jest wykonalne

w stanie początkowym w dowolnym czasie z przedziału $[2, 4]$, przejście t_3 zaś, w dowolnym czasie z przedziału $[0, 4]$. Jeżeli przejście t_1 zostanie wykonane w czasie $\theta = 3$, to nowy stan (M_1, δ_1) będzie określony następująco: $M_1 = (0, 1, 0, 0)$ i $\delta_1 = (-, [0, 0], -, -)$ (przejście t_2 musi zostać natychmiast wykonane).

Rozważmy skończony ciąg przejść $\alpha = t_1, t_2, \dots, t_n$ prowadzących od stanu (M_1, δ_1) , taki że:

$$(M_1, \delta_1) \xrightarrow{(t_1, \theta_1)} (M_2, \delta_2) \xrightarrow{(t_2, \theta_2)} \dots \xrightarrow{(t_n, \theta_n)} (M_{n+1}, \delta_{n+1}). \quad (8.13)$$

Definicja 8.9. Czasem wykonania ciągu przejść α nazywamy sumę:

$$\delta(\alpha) = \sum_{i=1}^n \theta_i. \quad (8.14) \blacksquare$$

Czas przejścia między stanami jest zdefiniowany identycznie jak w prostych sieciach czasowych.

Rozważmy ponownie przykład sieci czasowej przedstawionej na rys. 8.2. Niech dane będą: stan (M_1, δ_1) , przy czym $M_1 = (0, 1, 0, 0)$, $\delta_1 = (-, [0, 0], -, -, -)$ i stan (M_2, δ_2) , przy czym $M_2 = (0, 0, 0, 1)$ i $\delta_2 = (-, -, -, -)$. Dla ciągu przejść $\alpha = t_1, t_2$, takiego że:

$$(M_0, \delta_0) \xrightarrow{(t_1, 4)} (M_1, \delta_1) \xrightarrow{(t_2, 0)} (M_2, \delta_2), \quad (8.15)$$

czas wykonania tego ciągu obliczamy jako: $\delta(\alpha) = 4 + 0 = 4$. Czas przejścia między stanami (M_0, δ_0) i (M_2, δ_2) nie jest określony jednoznacznie. Czas $\delta(\alpha) = 4$ jest maksymalnym czasem przejścia między tymi stanami, minimalny czas przejścia wynosi 0.

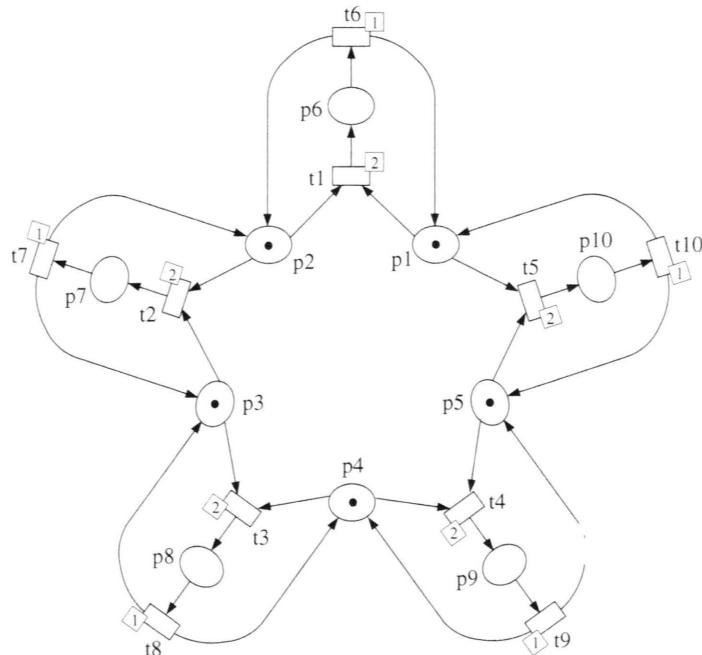
8.3. Modelowanie z zastosowaniem sieci czasowych

W tym podrozdziale omówimy przykłady zastosowania sieci czasowych do modelowania systemów czasu rzeczywistego: model problemu pięciu filozofów w formie prostej czasowej sieci Petriego oraz ponownie model prostego protokołu komunikacyjnego, ale mający postać sieci przedziałowej.

8.3.1. Model problemu pięciu filozofów

Rozważmy ponownie rozwiązywanie problemu pięciu filozofów przedstawione na rys. 3.18. W rozwiązyaniu tym przyjęto założenie, że filozof może sięgnąć po wi-

delce tylko wówczas, gdy oba widelce znajdują się na stole. W takiej sytuacji filozof może wziąć widelce, ale tylko oba jednocześnie. Rozbudujemy ten model o zależności czasowe. Przyjmiemy, że każdy z filozofów poświęca dwie jednostki czasu na myślenie, po czym podnosi widelce (jeżeli są dostępne) i przez jedną jednostkę czasu posila się. Rozwiążanie takie przedstawiono na rys. 8.3.



Rys. 8.3. Sieć czasowa modelująca problem pięciu filozofów

Znaczenie poszczególnych elementów sieci jest takie samo, jak było określone w p. 3.5.2. Jak sieć znakowana, tak i rozważana prosta sieć czasowa jest odwrotna, żywa, żywa ze względu na miejsca i bezpieczna. Sieć jest zachowawcza względem wektora wag $w = (1, 1, 1, 1, 1, 2, 2, 2, 2, 2)$, lecz nie jest siecią sprawiedliwą.

Stan początkowy tej sieci jest następujący:

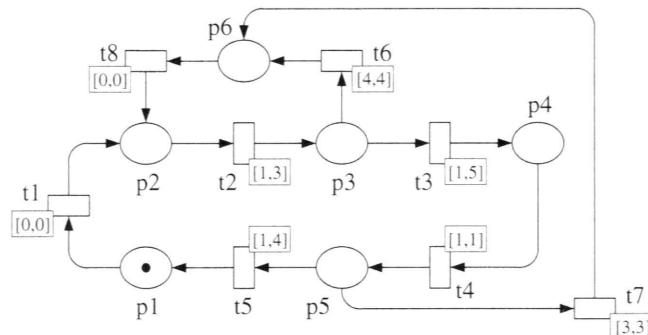
$$\begin{aligned} M_0 &= (1, 1, 1, 1, 1, 0, 0, 0, 0, 0), \\ \delta_0 &= (2, 2, 2, 2, 2, -, -, -, -, -). \end{aligned} \quad (8.16)$$

W stanie tym są wykonalne przejścia od $t1$ do $t5$. Jeżeli przyjmiemy, że nie wszyscy filozofowie poświęcają tyle samo czasu na myślenie, na przykład opóźnienie

statyczne dla przejść t_1 i t_2 wynosi 3, to tylko trzy przejścia będą wykonalne w stanie początkowym. Wybór opóźnień statycznych dla przejść od t_1 do t_5 ma zasadniczy wpływ na żywotność sieci. Jeżeli w porównaniu z rys. 8.3 zmienimy opóźnienie statyczne dla przejść t_1 i t_3 na 3, to przejścia te, a również przejścia t_6 i t_8 , będą martwe.

8.3.2. Model protokołu komunikacyjnego

Rozważmy ponownie model protokołu komunikacyjnego opisanego w podrozdz. 3.5. Rozwiążanie przedstawione na rys. 3.19 pozwalało zaznaczyć możliwość wystąpienia przeterminowania, ale bez ilościowego ujęcia aspektu czasu. Zastosowanie sieci przedziałowej umożliwia zbudowanie modelu protokołu, w którym będą przedstawione również zależności czasowe. Model ten znajduje się na rys. 8.4.



Rys. 8.4. Model protokołu komunikacyjnego

Przy dowolnym znakowaniu w rozważanej sieci znajduje się dokładnie jeden znacznik. W zależności od miejsca, w którym występuje znaczek, system znajduje się w jednym z sześciu stanów. Znaczenie poszczególnych stanów jest takie samo jak dla sieci na rys. 3.19. Znaczenie poszczególnych przejść również nie ulega zmianie, ale przypomnijmy je ze względu na zależności czasowe:

- t_1 – wygenerowanie nowej wiadomości, które jest wykonywane natychmiast po zakończeniu poprzedniego cyklu;
- t_2 – wysłanie wiadomości przez nadawcę – zajmuje od 1 do 3 jednostek czasu;
- t_3 – odebranie wiadomości przez odbiorcę – czas przypisany do przejścia t_3 łącznie z czasem przypisany do przejścia t_2 reprezentują czas trwania transmisji danych, który wahaj się od 2 do 7 jednostek czasu;

- t_4 – wysłanie potwierdzenia przez *odbiorcę* – zajmuje dokładnie 1 jednostkę czasu;
- t_5 – odebranie potwierdzenia przez *nadawcę* – czas przypisany do przejścia t_5 łącznie z czasem przypisanym do przejścia t_4 reprezentują czas trwania transmisji potwierdzenia, który waha się od 2 do 4 jednostek czasu;
- t_6 – zagubienie wiadomości – jeżeli przejście t_3 staje się aktywne i jego wykonanie nie nastąpi w ciągu 4 jednostek czasu, to będzie wykonane przejście t_6 ; jest to interpretowane jako zagubienie wiadomości;
- t_7 – zagubienie potwierdzenia – jeżeli przejście t_5 staje się aktywne i jego wykonanie nie nastąpi w ciągu 3 jednostek czasu, to będzie wykonane przejście t_7 ; jest to interpretowane jako zagubienie potwierdzenia;
- t_8 – wysłanie sygnału o konieczności ponowienia transmisji.

Przedstawiona sieć jest maszyną stanową. Jest ona żywa, bezpieczna, żywa ze względu na miejsca, ale nie jest siecią sprawiedliwą. Analizą innych własności sieci zależnych od czasu zajmiemy się w rozdz. 9.

8.4. Podsumowanie

W tym rozdziale omówiliśmy dwie klasy czasowych sieci Petriego. Rozbudowanie definicji sieci o model czasu pozwala na rozszerzenie zakresu ich zastosowań. Między innymi sieci czasowych można używać do modelowania systemów czasu rzeczywistego. Definicje obu prezentowanych klas zawierają funkcję opóźnień określających dla każdego z przejść czas, w którym, licząc od chwili uaktywnienia danego przejścia, może ono zostać wykonane. Dla prostych sieci czasowych opóźnienie jest nieujemną liczbą wymierną wyznaczającą dokładnie czas wykonania tego przejścia. Dla sieci przedziałowych podaje się przedział liczb wymiernych określający minimalne i maksymalne opóźnienie.

Wraz z wprowadzeniem zależności czasowych okazało się, że do opisania stanu sieci konieczne jest podanie nie tylko znakowania, ale również wektora opóźnień dynamicznych. Zmiany te mają zasadniczy wpływ na formalną analizę modelu. W przedstawionych w rozdz. 5 metodach analizy sieci uogólnionych nie uwzględnia się zależności czasowych i w związku z tym na ogół nie można tych metod stosować bezpośrednio do analizy sieci czasowych.

Analiza sieci czasowych

Analiza sieci czasowych jest zadaniem bardziej złożonym w porównaniu z analizą sieci, w których nie uwzględnia się ograniczeń czasowych. Pomijając własności czasowe modelu, można zastosować metody przedstawione w rozdz. 5, na przykład: grafy osiągalności, niezmienniki sieci, metody wyznaczania zatrzasków i pułapek itp. Uwzględnienie zależności czasowych występujących w modelu wymaga innego podejścia do sprawdzania własności sieci. Najbardziej popularną metodą jest przedstawienie zbioru osiągalnych znakowań (stanów) w postaci grafu osiągalności. Grafy takie definiuje się analogicznie jak dla sieci nieczasowych, przy czym muszą one uwzględniać upływ czasu (zob. prace: [5], [7], [9]). W przypadku sieci przedziałowych zbudowanie pełnego grafu osiągalności jest zazwyczaj niemożliwe ze względu na nieskończoną przestrzeń stanów. Zamiast grafu osiągalności stosowany jest zazwyczaj graf klas stanów, którego węzły reprezentują zbiory stanów sieci.

W tym rozdziale omówimy grafy osiągalności dla prostych sieci czasowych i grafy klas stanów dla sieci przedziałowych. Korzystając z tych metod, można analizować wszystkie podstawowe własności sieci, w tym własności czasowe, którymi zajmowaliśmy się w rozdz. 8.

9.1. Grafy osiągalności dla prostych sieci czasowych

Każdy węzeł grafu osiągalności skonstruowanego dla prostej sieci czasowej reprezentuje jeden osiągalny stan systemu, tzn. parę wektorów, znakowanie sieci i wektor opóźnień dynamicznych. Etykiety łuków grafu osiągalności również są parami. Pierwszy element takiej pary jest nazwą przejścia, które powoduje zmianę stanu, drugi oznacza dynamiczne opóźnienie wykonywanego przejścia w stanie wyjściowym.

Algorytm konstruowania grafu osiągalności jest następujący. Jako pierwszy węzeł grafu przyjmujemy stan początkowy sieci. Dla każdego węzła dodanego do gra-

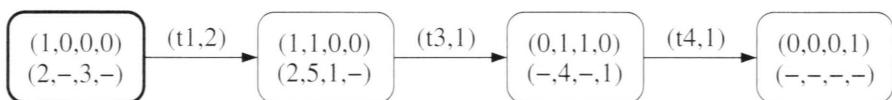
fu wyznaczamy wszystkie przejścia aktywne przy znakowaniu reprezentowanym przez ten węzeł. Jeżeli takie przejścia nie istnieją, to rozważany węzeł reprezentuje stan martwy. W przeciwnym wypadku analizujemy aktywne przejścia, wśród których przynajmniej jedno jest wykonalne. Dla każdego wykonalnego przejścia wyznaczamy nowy stan, który otrzymujemy w wyniku wykonania tego przejścia. Jeżeli takiego węzła nie ma jeszcze w grafie osiągalności, to dodajemy nowy węzeł i łuk. Jeżeli węzeł taki istnieje, to dodajemy tylko nowy łuk, który otrzymuje etykietę złożoną z nazwy wykonywanego przejścia i jego opóźnienia dynamicznego w stanie reprezentowanym przez węzeł początkowy tego łuku.

Niech dana będzie sieć czasowa $\mathcal{N} = (P, T, A, M_0, \sigma)$.

Definicja 9.1. Grafem osiągalności sieci \mathcal{N} nazywamy graf $\mathcal{G} = (V, A, \gamma)$ etykietowany nad zbiorem $T \times \mathbb{Q}_+$, spełniający warunki:

- (1) $V = \mathcal{R}(M_0, \delta_0)$ – jest zbiorem węzłów grafu.
- (2) A jest zbiorem łuków grafu \mathcal{G} postaci $((M, \delta), (t, \delta(t)), (M', \delta'))$, przy czym: $(M, \delta), (M', \delta') \in \mathcal{R}(M_0, \delta_0)$, $(M, \delta) \xrightarrow{t} (M', \delta')$, a $\delta(t)$ jest opóźnieniem dynamicznym przejścia t w stanie (M, δ) .
- (3) $\forall a = ((M, \delta), (t, \delta(t)), (M', \delta')) \in A: \gamma(a) = ((M, \delta), (M', \delta'))$. ■

Rozważmy sieć przedstawioną na rys. 8.1. Stan początkowy (M_0, δ_0) , dla którego $M_0 = (1, 0, 0, 0)$ i $\delta_0 = (2, -3, -)$, jest reprezentowany przez pierwszy węzeł grafu osiągalności. Przy znakowaniu początkowym aktywne są przejścia $t1$ i $t3$, ale wykonalne jest tylko przejście $t1$ prowadzące do stanu (M_1, δ_1) , przy czym $M_1 = (1, 1, 0, 0)$ i $\delta_1 = (2, 5, 1, -)$. Nowy stan jest reprezentowany przez drugi węzeł grafu. Opóźnienie dynamiczne przejścia $t1$ w stanie początkowym jest równe 2 ($\delta(t1) = 2$), a zatem łuk łączący te dwa węzły grafu osiągalności będzie miał etykietę $(t1, 2)$. Pełny graf osiągalności dla rozważanej sieci przedstawiono na rys. 9.1.



Rys. 9.1. Graf osiągalności dla prostej sieci czasowej

Jeżeli sieć jest ograniczona, to graf osiągalności jest grafem skończonym.

Niech $\mathcal{N} = (P, T, A, M_0, \sigma)$ będzie prostą siecią czasową. Równoważną siecią nieczasową do sieci \mathcal{N} nazywamy sieć znakowaną $\mathcal{N}' = (P, T, A, M_0)$. Własności

sieci \mathcal{N}' nie przenoszą się bezpośrednio na własności sieci \mathcal{N} . Grafy osiągalności sieci czasowej i odpowiadającej jej sieci nieczasowej mogą się znacznie różnić. Z jednej strony, ze względu na zależności czasowe pewne znakowania mogą być dla sieci czasowej nieosiągalne. Z drugiej strony, może występować kilka stanów, które różnią się jedynie wektorem opóźnień dynamicznych.

Graf osiągalności dla prostych sieci czasowych pozwala stosować metody analizy sieci, które są analogiczne do metod analizowania grafu osiągalności dla sieci uogólnionych. Analiza znakowań reprezentowanych przez węzły grafu osiągalności umożliwia rozstrzygnięcie kwestii dotyczących ograniczoności miejsc sieci (można sformułować twierdzenie analogiczne do twierdzenia 5.1).

Analizując drogi w grafie osiągalności, można sprawdzić własności związane z osiągalnością znakowań, stanów, żywotnością sieci, a także własności czasowe sieci. W przypadku sieci czasowych, oprócz takich pojęć, jak: znakowanie odtwarzalne, znakowanie własne itp., należy wprowadzić pojęcia: *stan odwracalny*, *stan własny* itd.

Definicja 9.2. Stan początkowy (M_0, δ_0) nazywamy *odtwarzalnym*, jeżeli istnieje stan $(M, \delta) \in \mathcal{R}(M_0, \delta_0)$ różny od stanu (M_0, δ_0) , z którego stan początkowy jest ponownie osiągalny. Sieć \mathcal{N} nazywamy *odtwarzalną*, jeżeli stan początkowy jest odtwarzalny. Sieć \mathcal{N} nazywamy *odwracalną*, jeżeli stan początkowy jest osiągalny z każdego stanu $(M, \delta) \in \mathcal{R}(M_0, \delta_0)$.

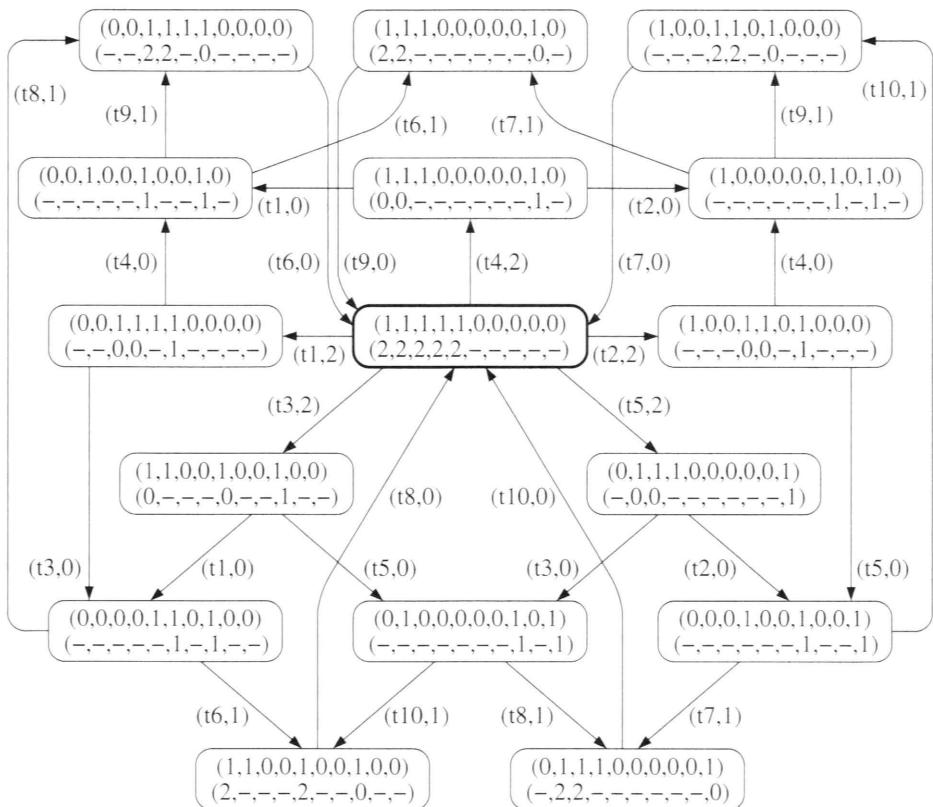
Stan (M, δ) sieci \mathcal{N} nazywamy *stanem własnym*, jeżeli jest on osiągalny z dowolnego stanu $(M', \delta') \in \mathcal{R}(M_0, \delta_0)$. ■

Stosując pojęcia wprowadzone w definicji 9.2 dla prostych sieci czasowych, można sformułować twierdzenia analogiczne do twierdzeń 5.2, 5.3 i 5.4 (zob. podrozdz. 5.1).

Drogi w grafie osiągalności odpowiadają wykonywanym ciągom przejść. Etykiety łuków grafu osiągalności (a dokładniej mówiąc, drugie elementy par stanowiących etykiety) są używane do wyznaczania czasu wykonania ciągu przejść czy też czasu przejścia od stanu (M, δ) do stanu $(M', \delta') \in \mathcal{R}(M, \delta)$. Wyznaczanie najkrótszego i najdłuższego czasu przejścia między dwoma stanami sieci sprowadza się do zastosowania standardowych algorytmów przeglądania grafów i poszukiwania najdłuższej i najkrótszej ważonej drogi, łączącej dwa węzły grafu (jako wagi łuków przyjmujemy drugie elementy par stanowiących etykiety).

Na rysunku 9.2 przedstawiono graf osiągalności dla prostej sieci czasowej modelującej problem pięciu filozofów (zob. rys. 8.3). Zbiór znakowań osiągalnych dla

rozważanej sieci zawiera tylko 11 elementów (zob. graf osiągalności na rys. 5.2), ale ze względu na zależności czasowe zbiór stanów osiągalnych zawiera aż 16 elementów. Łatwo zauważać, że wszystkie znakowania, które występowały w grafie osiągalności przedstawionym na rys. 5.2, pojawiły się również w rozważanym grafie.



Rys. 9.2. Graf osiągalności dla sieci modelującej problem pięciu filozofów

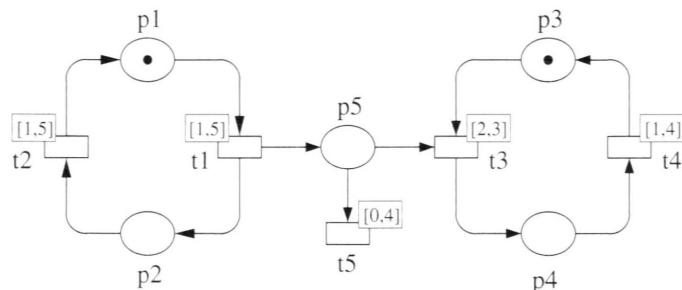
Analizując proste sieci czasowe, można stosować metody redukcji grafu osiągalności, które są analogiczne do metod dla sieci nieczasowych, tzn. grafy silnie spójnych składowych, symetrie sieci i trwałe zbiory przejść. W przypadku redukcji grafu osiągalności ze względu na symetrie i trwałe zbiory przejść metody te zmodyfikowano, tak aby uwzględniały zależności czasowe modelu.

Symetrie sieci są definiowane jako funkcje, w których dodatkowo bierze się pod uwagę opóźnienia statyczne przejść, tzn. opóźnienia statyczne przejścia i przejścia mu równoważnego muszą być takie same. Ponadto zamiast równoważności znakowania sieci, definiuje się równoważność stanów (dodatkowo uwzględnia się wektory opóźnień dynamicznych). Algorytm konstruowania grafu osiągalności zredukowanego ze względu na symetrię dla prostych sieci czasowych jest analogiczny do algorytmu przedstawionego w podrozdz. 5.4. Należy jednak wziąć pod uwagę, że w tym przypadku węzły grafu są stanami i łuki mają etykiety, będące parami liczb.

Trwałe zbiory przejść dla prostych sieci czasowych tak zdefiniowano, aby była w nich uwzględniona wykonalność przejść (trwały zbiór przejść musi zawierać co najmniej jedno przejście wykonalne). Ponadto, dokonując redukcji grafu, nie wolno stracić informacji o czasach przejścia między stanami sieci.Więcej informacji na ten temat można znaleźć w pracach [9] i [68].

9.2. Grafy klas stanów

Zbudowanie grafu osiągalności dla sieci przedziałowych, którego węzły reprezentowałyby pojedyncze stany, jest na ogół zadaniem niemożliwym do wykonania. Jak wspomniałem w podrozdz. 8.2, zbiór $\mathcal{R}(M_0, \delta_0)$ jest w ogólnym przypadku zbiorem nieskończonym. W literaturze pojawiło się kilka propozycji metod reprezentowania przestrzeni stanów sieci czasowych w postaci skończonego grafu skierowanego (zob. prace: [5], [6], [7], [53] i [92]). W tym rozdziale zajmiemy się jedną z pierwszych metod, którą zaproponowano w pracy [5].



Rys. 9.3. Sieć czasowa

Rozważmy przykład sieci przedstawionej na rys. 9.3. Stan początkowy sieci (M_0, δ_0) jest określony następująco: $M_0 = (1, 0, 1, 0, 0)$ i $\delta_0 = ([1, 5], -, -, -, -)$. W stanie początkowym aktywne jest tylko przejście t_1 i może ono być wykonane

w dowolnym czasie θ_1 spełniającym warunek: $1 \leq \theta_1 \leq 5$. Niezależnie od czasu, w którym zostanie wykonane przejście $t1$, otrzymujemy nowy stan (M_1, δ_1) , przy czym $M_1 = (0, 1, 1, 0, 1)$ i $\delta_1 = (-, [1, 5], [2, 3], -, [0, 4])$. Od tej pory symbolu τ będziemy używać do oznaczenia czasu, w którym sieć osiągnęła rozważany stan.

W stanie (M_1, δ_1) są aktywne przejścia $t2$, $t3$ i $t5$. Przyjrzyjmy się bliżej przejściu $t2$. Może ono zostać wykonane w dowolnym czasie $\tau + \theta_2$, spełniającym warunek: $1 \leq \theta_2 \leq 3$. W zależności od czasu, w którym będzie wykonane przejście $t2$, otrzymamy w wyniku różne stany, które będą mieć wspólne znakowanie $M_2 = (1, 0, 1, 0, 1)$, ale różne wektory opóźnień dynamicznych, na przykład: $\delta_{2,1} = ([1, 5], -, [1, 2], -, [0, 3])$ dla $\theta_2 = 1$, $\delta_{2,2} = ([1, 5], -, [0, 1], -, [0, 2])$ dla $\theta_2 = 2$ itd. Wszystkie takie stany będą reprezentowane jako jedna *klasa stanów*.

Klasa stanów jest parą (M, D) , przy czym M jest znakowaniem sieci czasowej, D zaś jest dziedziną czasu wykonania dla tej klasy. Dziedzinę D definiuje się jako zbiór rozwiązań układu nierówności liniowych, które określają ograniczenia czasów wykonania przejść aktywnych w danym stanie.

Dla rozważanej sieci początkowa klasa stanów (M_0, D_0) jest postaci:

$$\begin{aligned} M_0 &= (1, 0, 1, 0, 0), \\ D_0: & 1 \leq \theta_1 \leq 5. \end{aligned} \tag{9.1}$$

Druga klasa stanów, będąca wynikiem wykonania przejścia $t1$, jest postaci:

$$\begin{aligned} M_1 &= (0, 1, 1, 0, 1), \\ D_1: & \begin{cases} 1 \leq \theta_2 \leq 5, \\ 2 \leq \theta_3 \leq 3, \\ 0 \leq \theta_5 \leq 4. \end{cases} \end{aligned} \tag{9.2}$$

Rozważmy klasę (M_1, D_1) . Przy znakowaniu M_1 są aktywne 3 przejścia. Jeżeli chcemy wyznaczyć dopuszczalne czasy wykonania przejścia $t2$, to do układu nierówności D_1 należy dodać nierówności: $\theta_2 \leq \theta_3$ i $\theta_2 \leq \theta_5$. Stąd też wynika podany już wcześniej dopuszczalny zakres czasu wykonania przejścia $t2$.

Przymijmy, że przejście $t2$ będzie wykonane w pewnym czasie θ_2 spełniającym warunek:

$$1 \leq \theta_2 \leq 3. \tag{9.3}$$

Po wykonaniu przejścia $t2$ pozostają aktywne przejścia $t3$ i $t5$, a dopuszczalne nowe czasy wykonania tych przejść, θ'_3 i θ'_5 , muszą spełniać warunki:

$$\begin{aligned} 2 \leq \theta'_3 + \theta_2 \leq 3, \\ 0 \leq \theta'_5 + \theta_2 \leq 4, \end{aligned} \tag{9.4}$$

które można zapisać również jako:

$$\begin{aligned} 2 - \theta_2 &\leq \theta'_3 \leq 3 - \theta_2, \\ 0 - \theta_2 &\leq \theta'_5 \leq 4 - \theta_2 \end{aligned} \quad (9.5)$$

lub:

$$\begin{aligned} 2 - \theta'_3 &\leq \theta_2 \leq 3 - \theta'_3, \\ 0 - \theta'_5 &\leq \theta_2 \leq 4 - \theta'_5. \end{aligned} \quad (9.6)$$

Biorąc pod uwagę fakt, że zmienne θ_i muszą przyjmować wartości nieujemne, po eliminacji z przedstawionych nierówności zmiennej θ_2 otrzymujemy zależności:

$$\begin{aligned} 0 &\leq \theta'_3 \leq 2, \\ 0 &\leq \theta'_5 \leq 3, \\ 2 - \theta'_3 &\leq 4 - \theta'_5, \\ 0 - \theta'_5 &\leq 3 - \theta'_3. \end{aligned} \quad (9.7)$$

Pierwsze dwie nierówności wynikają z zależności (9.3) i (9.6). Pozostałe nierówności wynikają z zależności (9.6). Porządkując nierówności i biorąc pod uwagę fakt, że ostatnia nierówność jest zbędna, otrzymujemy ostatecznie zależności:

$$\begin{aligned} 0 &\leq \theta'_3 \leq 2, \\ 0 &\leq \theta'_5 \leq 3, \\ \theta'_5 - \theta'_3 &\leq 2. \end{aligned} \quad (9.8)$$

Po dodaniu nierówności dla nowego aktywnego przejścia $t1$ i formalnej zamianie nazw zmiennych klasa (M_2, D_2) , która jest wynikiem wykonania przejścia $t2$, ma postać:

$$M_2 = (1, 0, 1, 0, 1),$$

$$D_2: \left\{ \begin{array}{l} 0 \leq \theta_3 \leq 2, \\ 0 \leq \theta_5 \leq 3, \\ \theta_5 - \theta_3 \leq 2, \\ 1 \leq \theta_1 \leq 5. \end{array} \right. \quad (9.9)$$

Jeżeli klasa (M', D') jest osiągana z (M, D) w wyniku wykonania przejścia t_i w czasie θ_i , to nowe znakowanie M' będzie wyznaczone zgodnie z wzorem (8.10), a dziedzina D' będzie wyznaczona na podstawie następującego algorytmu:

1. Do układu nierówności D dodajemy nierówności postaci $\theta_i \leq \theta_j$ dla wszystkich zmiennych θ_j występujących w układzie D , dla których $i \neq j$.

2. Każdą ze zmiennych θ_j występujących w układzie D , dla których $i \neq j$, zastępujemy sumą czasu wykonania przejścia t_i i nowej zmiennej θ'_j , tzn.: $\theta_j = \theta_i + \theta'_j$.
3. Eliminujemy z układu nierówności zmienną θ_i , wprowadzając nowe przedziały wykonania dla przejść, które pozostały aktywne (tzn. podwójne nierówności ograniczające zmienne θ'_j), oraz nierówności określające zależności między zmiennymi θ'_j .
4. Tak jak w przypadku zmiennej θ_i , z układu nierówności eliminujemy wszystkie zmienne odnoszące się do przejść, które przestały być aktywne po wykonaniu przejścia t_i .
5. Do tworzonego układu dodajemy nierówności charakteryzujące przejścia, które zostały uaktywnione po przejściu do klasy (M', D') . Dla każdego z tych przejść dodajemy nierówności postaci: $\sigma_{\min}(t_j) \leq \theta_j \leq \sigma_{\max}(t_j)$.
6. Dokonujemy zamiany nazw zmiennych, usuwając znaki '.

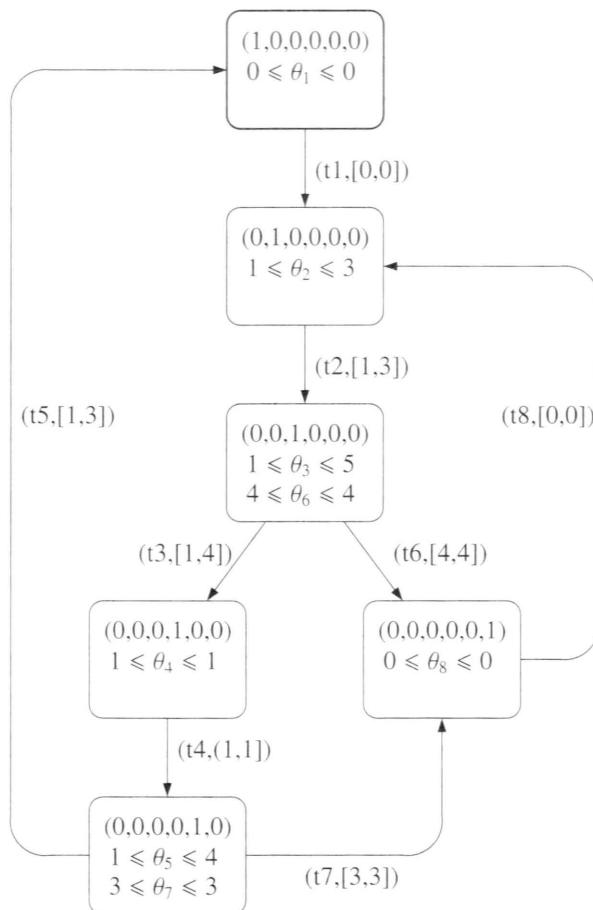
Otrzymany ostatecznie układ nierówności zawiera dokładnie tyle zmiennych, ile jest aktywnych przejść przy znakowaniu M' . Układ ten zawiera nierówności postaci $\delta_{\min}(t_j) \leq \theta_j \leq \delta_{\max}(t_j)$ dla każdego z przejść, które są aktywne przy znakowaniu M' , oraz nierówności postaci $\theta_j - \theta_k \leq b_{jk}$, przy czym $j \neq k$ i $b_{jk} \in \mathbb{R}_+$ dla każdej pary przejść t_j, t_k , które były aktywne przy znakowaniu M i pozostają aktywne przy znakowaniu M' .

Twierdzenie 9.1. Liczba klas stanów sieci N jest skończona wtedy i tylko wtedy, gdy sieć jest ograniczona. ■

Dowód powyższego twierdzenia można znaleźć w pracy [5].

Zbiór klas stanów jest przedstawiany w postaci *drzewa klas stanów*, którego korzeniem jest klasa początkowa. Dwie klasy stanów są połączone łukiem, jeżeli istnieje przejście t , którego wykonanie prowadzi od pierwszej z tych klas do drugiej, tzn. jeżeli przejście t jest wykonalne przy znakowaniu należącym do pierwszej klasy stanów. Etykieta łuku składa się z dwóch elementów, nazwy przejścia i przedziału określającego czas, w którym przejście to może zostać wykonane. Każdy węzeł takiego drzewa ma skończoną liczbę następców, co najwyżej po jednym następcu dla każdego z przejść, które są aktywne przy znakowaniu należącym do danej klasy. Sklejając równoważne klasy, tzn. mające takie same znakowania i dziedziny, otrzymujemy *graf klas* dla przedziałowej sieci czasowej.

Na rysunku 9.4 przedstawiono graf klas dla sieci czasowej modelującej protokół komunikacyjny (zob. rys. 8.4). Warto zwrócić uwagę, że w tym przypadku graf klas dla sieci czasowej i graf osiągalności dla sieci nieczasowej (zob. rys. 5.3) są izomorficzne.



Rys. 9.4. Graf klas dla sieci modelującej protokół komunikacyjny

Jeżeli graf klas dla sieci czasowej jest skończony, to można go zastosować do analizy własności sieci, podobnie jak grafy osiągalności dla prostych sieci czasowych. Graf klas jawnie przedstawia wszystkie znakowania osiągalne sieci. Wykorzystując etykiety łuków grafu, można rozstrzygać kwestie żywotności sieci, a także własności czasowe. Graf nie przedstawia jawnie wszystkich stanów sieci, dlatego kwestie odwracalności można rozstrzygać bezpośrednio tylko w odniesieniu do znakowań sieci, a nie stanów.

Etykiety łuków grafu klas (a dokładniej mówiąc, drugie elementy par stanowiących etykiety) są używane do wyznaczania minimalnego i maksymalnego czasu

przejścia między dwiema klasami stanów. Przy wyznaczaniu minimalnego czasu przejścia stosuje się lewe, maksymalnego zaś prawe końce przedziałów.

Rozważmy ponownie sieć modelującą protokół komunikacyjny (zob. rys 8.4). Stosując graf klas, można wyznaczyć minimalny i maksymalny czas, który musi upływać od wygenerowania komunikatu przez nadawcę (rozpoczynając od stanu początkowego) do odebrania potwierdzenia jego otrzymania. Minimalny czas jest powiązany z wykonaniem cyklu złożonego z przejść t_1, t_2, t_3, t_4, t_5 i wynosi 4 jednostki czasu. Maksymalny czas, przy założeniu, że nie zagubiono wiadomości ani potwierdzenia, wynosi 11 jednostek czasu. Jeżeli dopuszczać możliwość zagubienia wiadomości lub potwierdzenia, to maksymalny czas jest nieograniczony.

Przedstawione w tym podrozdziale metody reprezentowania grafu klas i analizy sieci czasowych opierają się na założeniu, że przejścia nie są wielokrotnie aktywne, tzn. nie są aktywne współbieżnie ze sobą. Jeżeli dopuszczać taką możliwość, to analiza nieco się komplikuje. Należy wówczas rozważyć każdą aktywność przejścia niezależnie i przy wyznaczaniu klas stanów z każdą z tych aktywności powiązać niezależną zmienną (zob. pracę [5]).



9.3. Podsumowanie

W tym rozdziale omówiliśmy metody analizy prostych i przedziałowych sieci czasowych. Konieczność uwzględnienia zależności czasowych w tworzonym modelu powoduje, że dla tych klas sieci należy konstruować grafy osiągalności, których węzły reprezentują nie znakowania, ale stany. W większości przypadków (a zwłaszcza dla sieci przedziałowych) oznacza to konieczność rozważania nieskończonej przestrzeni stanów, a więc uniemożliwia zbudowanie pełnego grafu osiągalności. Analizując sieci przedziałowe, zamiast klasycznych grafów osiągalności korzysta się z grafów klas stanów, które są skończone dla sieci ograniczonych. Jednocześnie grafy takie pozwalają analizować wszystkie podstawowe własności sieci, w tym również własności czasowe.

Sieci kolorowane

Sieci kolorowane (zob. prace: [31], [32] i [33]) są zaliczane do tzw. sieci Petriego wysokiego poziomu (zob. prace: [29], [34] i [65]). Rozwijano tę klasę sieci w celu uzyskania narzędzi do modelowania bardziej złożonych systemów, które można spotkać w zastosowaniach inżynierskich. W tym celu połączono możliwości sieci niskiego poziomu z właściwościami języków programowania. Z sieci Petriego zaczerpnięto mechanizmy służące do modelowania operacji współbieżnych, a z języków programowania – sposoby definiowania typów danych oraz zasady działania na nich.

Zasadniczą różnicą między sieciami uogólnionymi a sieciami kolorowanymi jest możliwość definiowania, w przypadku tych drugich, różnych typów znaczników, które odpowiadają typom danych występującym w popularnych językach programowania. Każde miejsce sieci ma przypisany tzw. *kolor*, czyli typ danych, którego wartości miejsca to może przechowywać. Znakowanie miejsca w sieci kolorowanej jest wielozbiorem zbudowanym nad zbiorem określającym kolor takiego miejsca. Łuki i przejścia sieci mają przypisane wyrażenia (mogące zawierać zmienne), które umożliwiają manipulowanie znacznikami różnych typów.

W tym rozdziale zajmiemy się kolorowanymi sieciami Petriego. Przede wszystkim omówimy pojęcie wielozbioru oraz formalną definicję niehierarchicznych sieci kolorowanych. Ze względu na zasadnicze różnice występujące między sieciami uogólnionymi i kolorowanymi, konieczne było wprowadzenie definicji podstawowych własności omawianej klasy sieci; na przykład przy omawianiu pojęć dotyczących ograniczoności sieci należy brać pod uwagę różne typy i wartości znaczników występujących w poszczególnych miejscach sieci.

Sieci kolorowane są pierwszą z prezentowanych w tej książce klas sieci, dla której można konstruować hierarchiczne modele. Z tego powodu w jednym z podrozdziałów omówimy mechanizmy hierarchizacji CP-sieci. Wykorzystanie hierarchicznego opisu pozwala stosować sieci kolorowane do modelowania bardziej złożonych systemów współbieżnych. Ponadto rozdział zawiera przykłady modelowania z zastosowaniem tej klasy sieci.

10.1. Wielozbiory

Podstawowa różnica między *zbiorem* i *wielozbiorem* polega na tym, że w wielozbiorze może wielokrotnie występować ten sam element. Jak dla zbiorów przyjmuje się istnienie pewnego uniwersum, tak dla wielozbiorów zakładamy istnienie pewnego *zbioru bazowego*. Stanowi on podstawę do rozważania rodziny wielozbiorów zbudowanych nad danym zbiorem.

Definicja 10.1. *Wielozbiorem* A^* nad niepustym zbiorem A nazywamy dowolną funkcję

$$A^*: A \rightarrow \mathbb{Z}_+ \quad (10.1)$$

■

Niech A będzie dowolnym zbiorem. Symbolem 2^{A^*} będziemy oznaczać rodzinę¹ wszystkich wielozbiorów zbudowanych nad zbiorem A .

Rozważmy zbiór $A = \{a, b, c, d, e\}$. Przykładami wielozbiorów nad zbiorem A są wielozbiory $A_1^* = \{a, a, c, c, c, d, e, e\}$ i $A_2^* = \{a, b\}$. Wielozbiory można zapisywać również algebraicznie: $A_1^* = 2a + 3c + d + 2e$ i $A_2^* = a + b$. Dowolny wielozbiór nad zbiorem A można zatem zapisać jako:

$$A^* = \sum_{x \in A} A^*(x)x, \quad (10.2)$$

przy czym $A^*(x)$ oznacza *współczynnik* (liczbę wystąpień) elementu x w wielozbiorze A^* . Element $x \in A$ należy do wielozbioru A^* , jeżeli $A^*(x) > 0$.

Korzystając z zapisu algebraicznego, można łatwo zdefiniować podstawowe operacje na wielozbiorach.

Definicja 10.2. Niech A^* , A_1^* i A_2^* będą dowolnymi wielozbiorami nad zbiorem A i niech $n \in \mathbb{Z}_+$.

Sumą wielozbiorów A_1^* i A_2^* nazywamy wielozbiór:

$$A_1^* + A_2^* = \sum_{x \in A} (A_1^*(x) + A_2^*(x))x. \quad (10.3)$$

¹ W literaturze naukowej nie ma jednoznacznego symbolu stosowanego do oznaczania rodziny wielozbiorów nad określonym zbiorem A . Symbol wprowadzony przeze mnie można tak uzasadnić: symbol 2^A oznacza rodzinę podzbiorów zbioru A . W lingwistyce matematycznej symbolu * używa się do oznaczenia domknięcia Kleenego. Jeśli przyjmiemy, że A jest alfabetem, zaś *konkatenację* słów będziemy interpretować jako dodawanie wielozbiorów, to 2^{A^*} możemy zinterpretować jako zbiór wszystkich wielozbiorów, które można zbudować nad zbiorem A .

Iloczynem wielozbioru A^ przez stałą n nazywamy wielozbiór:*

$$n \cdot A^* = \sum_{x \in A} (n \cdot A^*(x))x. \quad (10.4)$$

Relację równości wielozbiorów definiujemy następująco:

$$A_1^* = A_2^* \Leftrightarrow \forall x \in A : A_1^*(x) = A_2^*(x). \quad (10.5)$$

Relację mniejszości wielozbiorów definiujemy następująco:

$$A_1^* \leq A_2^* \Leftrightarrow \forall x \in A : A_1^*(x) \leq A_2^*(x). \quad (10.6)$$

Jeżeli $A_2^* \leq A_1^*$, to różnicę wielozbiorów A_1^* i A_2^* nazywamy wielozbiór:

$$A_1^* - A_2^* = \sum_{x \in A} (A_1^*(x) - A_2^*(x))x. \quad (10.7)$$

Rozmiarem wielozbioru A^ nazywamy liczbę:*

$$|A^*| = \sum_{x \in A} A^*(x). \quad (10.8)$$

■



10.2. Niehierarchiczne sieci kolorowane

Jak wspomniałem na początku tego rozdziału, sieci kolorowane umożliwiają definiowanie i manipulowanie różnymi typami znaczników. Jednym z elementów sieci jest zbiór typów (*kolorów*), które mogą być przypisane jej miejscom. Każde miejsce ma przypisany jeden typ i może przechowywać tylko te znaczniki, które są wartościami tego typu. Konsekwencją takiego podejścia jest modyfikacja na przykład funkcji wag łuków, tak by było możliwe określenie nie tylko liczby, ale również wartości znaczników pobieranych czy też dodawanych do miejsc sieci. Wagi takie mogą być wyrażeniami zawierającymi zmienne. W zależności od wartościowania tych zmiennych wielozbory znaczników usuwanych (dodawanych) w wyniku różnych wykonan tego samego przejścia mogą się różnić nie tylko wartościami znaczników, ale również rozmiarami tych wielozbiorów. Przepływ znaczników jest dodatkowo sterowany za pomocą wyrażeń logicznych (tzw. *zastrzeżeń*), przy pisywanych przejściom sieci. W odróżnieniu od sieci uogólnionych dopuszcza się w sieciach kolorowanych istnienie kilku łuków o tym samym początku i końcu. Wszystkie te różnice powodują, że sieć kolorowana jest definiowana jako struktura zawierająca aż 9 elementów.

W odniesieniu do sieci kolorowanych zachowują ważność przedstawione w rozdz. 2 definicje pójęć związanych z grafami, na przykład zbiór poprzedników węzła, zbiór nastęników itp. Dodatkowo wprowadzimy kilka oznaczeń umożliwiających opisywanie wyrażeń zawierających zmienne i manipulowanie nimi.

Definicja sieci kolorowanych nie wymaga wiążania z nią ściśle określonej składni zapisu wyrażeń używanych do opisu elementów sieci. Przyjęto jedynie założenie, że składnia taka istnieje łącznie z odpowiednią semantyką i można w jednoznaczny sposób zdefiniować następujące pojęcia.

Dla dowolnej zmiennej v symbolem $\mathcal{T}(v)$ będzie oznaczany typ tej zmiennej, tzn. zbiór wszystkich wartości, które ta zmienna może przyjmować.

Niech x będzie wyrażeniem. Symbolem $\mathcal{V}(x)$ oznaczamy zbiór zmiennych występujących w wyrażeniu x oraz symbolem $\mathcal{T}(x)$ – typ wyrażenia, tzn. zbiór wszystkich wartości, które można otrzymać w wyniku wartościowania wyrażenia x .

Dla dowolnego zbioru zmiennych V definiujemy typ zbioru zmiennych jako

$$\mathcal{T}(V) = \{\mathcal{T}(v) : v \in V\}. \quad (10.9)$$

Symbol Bool będzie oznaczany zbiorem wartości logicznych (zawierający wartości $\{\text{false}, \text{true}\}$).

Dla łuku a symbole $P(a)$ i $T(a)$ będą oznaczać ten z węzłów łuku, który jest odpowiednio miejscem i przejściem.

Definicja 10.3. Strukturę $\mathcal{N} = (\Sigma, P, T, A, \gamma, C, G, E, M_0)$ nazywamy siecią kolorowaną, jeżeli są spełnione warunki:

- (1) Σ jest niepustym, skończonym zbiorem typów (kolorów), z których każdy jest zbiorem niepustym.
- (2) P jest niepustym, skończonym zbiorem miejsc.
- (3) T jest niepustym, skończonym zbiorem przejść.
- (4) A jest niepustym, skończonym zbiorem łuków, który spełnia warunek: $P \cap T = P \cap A = T \cap A = \emptyset$.
- (5) $\gamma: A \rightarrow (P \times T) \cup (T \times P)$ jest funkcją zaczepienia, która przypisuje każdemu łukowi uporządkowaną parę węzłów.
- (6) $C: P \rightarrow \Sigma$ jest funkcją typów (kolorów), która określa, jakiego typu znaczniki każde z miejsc może zawierać.
- (7) G jest funkcją zatrzeżeń (dozorów), przypisującą każdemu z przejść wyrażenie, takie że: $\forall t \in T: \mathcal{T}(G(t)) \subseteq \text{Bool} \wedge \mathcal{T}(\mathcal{V}(G(t))) \subseteq \Sigma$, tzn. wyrażenie, które może zawierać zmienne typów należących do Σ i którego dowolne wartościowanie daje w wyniku wartość logiczną.

- (8) E jest funkcją wag łuków, przypisującą każdemu z łuków wyrażenie, takie że: $\forall a \in A: \mathcal{T}(E(a)) \subseteq 2^{C(P(a))^*} \wedge \mathcal{T}(\mathcal{V}(E(a))) \subseteq \Sigma$, tzn. wyrażenie, które może zawierać zmienne typów należących do Σ i którego dowolne wartościowanie daje w wyniku wielozbiór nad typem przypisanym do miejsca $P(a)$.
- (9) M_0 jest znakowaniem początkowym, takim że dla dowolnego miejsca $p \in P$ $M_0(p) \in 2^{C(p)^*}$, tzn. M_0 jest funkcją, która każdemu miejscu przyporządkowuje wielozbiór nad typem przypisanym do tego miejsca. ■

Zdefiniowanie wag łuków w postaci wyrażeń mogących zawierać zmienne ma wpływ na sposób definiowania aktywności przejścia. Aby przejście było aktywne, jego miejsca wejściowe muszą zawierać nie tylko odpowiednią liczbę, ale również odpowiednie wartości znaczników. Może tak się zdarzyć, że przy danym znakowaniu sieci to samo przejście będzie wykonywane z różnymi zestawami znaczników w zależności od wartościowania odpowiednich zmiennych.

Niech dana będzie sieć kolorowana $\mathcal{N} = (\Sigma, P, T, A, \gamma, C, G, E, M_0)$.

Niech $V = P \cup T$ będzie zbiorem węzłów sieci. Zbiór łuków otaczających węzeł x definiujemy następująco:

$$A(x) = \{a \in A : \exists x' \in V \wedge (\gamma(a) = (x, x') \vee \gamma(a) = (x', x))\}. \quad (10.10)$$

Symbol $\mathcal{V}(t)$ będziemy oznaczać zbiór zmiennych, które występują w zastrzeżeniu przejścia t lub w wyrażeniach przypisanych łukom ze zbioru $A(t)$.

Definicja 10.4. Wiązaniem przejścia $t \in T$ nazywamy odwzorowanie b określone na zbiorze $\mathcal{V}(t)$, takie że każdej zmiennej przypisano wartość należącą do typu tej zmiennej i jednocześnie jest spełnione zastrzeżenie przejścia t (tzn. wynikiem wartościowania zastrzeżenia jest wartość *true*).

Zbiór wszystkich wiązań przejścia t oznaczamy symbolem $\mathcal{B}(t)$. Symbolem \mathcal{B} będziemy oznaczać zbiór par (t, b) , przy czym $t \in T$ i $b \in \mathcal{B}(t)$. ■

Definicja 10.5. Znakowaniem sieci kolorowanej \mathcal{N} nazywamy dowolną funkcję M określoną na zbiorze miejsc sieci, taką że:

$$\forall p \in P: M(p) \in 2^{C(p)^*}. \quad (10.11)$$

Niech $x, y \in V$ będą węzłami sieci. Symbolem $E(x, y)$ będziemy oznaczać wyrażenie powstałe przez zsumowanie wyrażeń na tych łukach sieci, których x jest początkiem, y zaś końcem. (Wyznaczenie sumy tych wyrażeń jest możliwe, ponieważ wszystkie te wyrażenia są tego samego typu). Wynik wartościowania wyrażenia $E(x, y)$ przy ustalonym wiązaniu b będziemy oznaczać symbolem $E(x, y)_b$. Symbol $E(p, t)_b$ oznacza w rzeczywistości wielozbiór znaczników, które są usuwane z miejsca p , jeżeli przejście t jest wykonywane przy wiązaniu b . Analogicznie symbol $E(t, p)_b$ oznacza wielozbiór znaczników dodawanych do miejsca p .

Definicja 10.6. Przejście $t \in T$ jest *aktywne* przy znakowaniu M i wiązaniu $b \in \mathcal{B}(t)$ (jest M -aktywne przy wiązaniu b), jeżeli spełniony jest warunek:

$$\forall p \in In(t): E(p, t)_b \leq M(p), \quad (10.12)$$

tz. jeżeli każde miejsce wejściowe przejścia t zawiera przy wiązaniu b odpowiednią liczbę i odpowiednie wartości znaczników.

Przejście t jest *aktywne* przy znakowaniu M , jeżeli jest aktywne przy znakowaniu M i pewnym wiązaniu $b \in \mathcal{B}(t)$. ■

Jeżeli przejście t jest aktywne przy znakowaniu M i wiązaniu b , to nowe znakowanie M' , które uzyskano w wyniku wykonania przejścia t , jest określone w sposób następujący:

$$M'(p) = \begin{cases} M(p) - E(p, t)_b & \text{gdy } p \in In(t) - Out(t), \\ M(p) + E(t, p)_b & \text{gdy } p \in Out(t) - In(t), \\ M(p) - E(p, t)_b + E(t, p)_b & \text{gdy } p \in In(t) \cap Out(t), \\ M(p) & \text{w pozostałych przypadkach.} \end{cases} \quad (10.13)$$

Jeżeli przejście t jest aktywne przy znakowaniu M i wiązaniu b oraz w wyniku jego wykonania otrzymujemy znakowanie M' , to fakt ten zapisujemy jako:

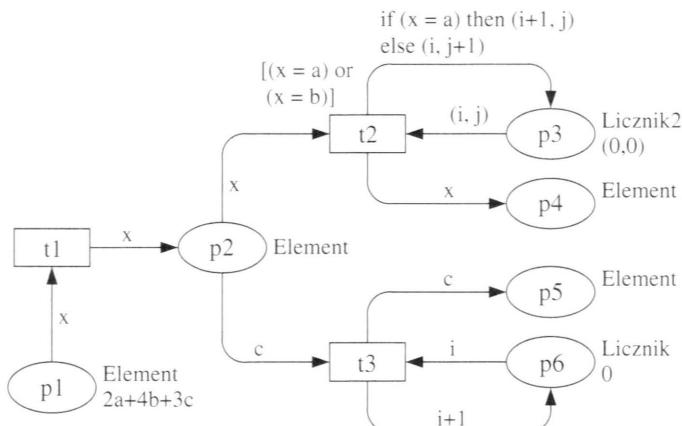
$$M \xrightarrow{(t,b)} M'. \quad (10.14)$$

Jeżeli przy rozważaniu zmiany znakowania wiązanie przejścia t nie będzie miało znaczenia, to będziemy stosować zapis uproszczony:

$$M \xrightarrow{t} M'. \quad (10.15)$$

Na rysunku 10.1 przedstawiono przykład niehierarchicznej sieci kolorowanej \mathcal{N} . Etykiety oznaczające nazwy węzłów sieci, umieszczone wewnątrz figur reprezentujących te węzły. W rozważanej sieci użyto trzech typów danych

$(\Sigma = \{Licznik, Element, Licznik2\})$. Są to: *Licznik* – podzbiór zbioru liczb całkowitych, *Element* – typ wyliczeniowy i *Licznik2* – iloczyn kartezjański (para elementów typu *Licznik*). Definicje tych typów umieszczone w tzw. *węźle deklaracji*. Etykiety z nazwami typów przypisanych poszczególnym miejscom umieszczone obok tych miejsc.



```

color Licznik = int with 0..100;
color Element = with a | b | c;
color Licznik2 = product Licznik * Licznik;
var i, j : Licznik;
var x : Element;

```

Rys. 10.1. Przykład niehierarchicznej sieci kolorowanej

Przymając, że zbiór miejsc sieci jest uporządkowany (zgodnie z wartościami indeksów miejsc), znakowanie początkowe sieci można przedstawić następująco: $M_0 = (2a + 4b + 3c, \emptyset, (0,0), \emptyset, \emptyset, \emptyset)$, tzn. miejsce $p1$ zawiera 9 znaczników, miejsca $p3$ i $p6$ zawierają po jednym znaczniku, a pozostałe miejsca są puste. Jeżeli znakowanie początkowe miejsca nie jest zbiorem pustym, to obok tego miejsca występuje etykieta opisująca to znakowanie.

Przejście $t2$ ma przypisane zastrzeżenie w postaci wyrażenia $[(x = a) \text{ or } (x = b)]$. Biorąc pod uwagę wyrażenie $(E(p2, t2) = x)$ przypisane do łuku $(p2, t2)$, oznacza to, że aby przejście $t2$ było aktywne, w miejscu $p2$ musi się znajdować co najmniej jeden znacznik o wartości a lub b . W przypadku pozostałych przejść przyjmuje się domyślne zastrzeżenie, tzn. stałą *true*.

Przy znakowaniu początkowym M_0 aktywne jest tylko przejście $t1$, ale przy trzech różnych wiązaniach²: $b_1 = (a/x)$, $b_2 = (b/x)$ i $b_3 = (c/x)$. W rzeczywistości można by mówić aż o 9 wiązaniach ze względu na liczbę dostępnych znaczników. Ponadto przejście t mogłoby zostać wykonane wspólnie samo ze sobą nawet przy tych samych wiązaniach.

Jeżeli zostanie wykonane przejście $t1$ przy wiązaniu b_1 , to otrzymamy nowe znakowanie $M_1 = (a + 4b + 3c, a, (0, 0), \emptyset, \emptyset, 0)$. Przy znakowaniu M_1 aktywne jest (oprócz $t1$) przejście $t2$ przy wiązaniu $b = (a/x, 0/i, 0/j)$. Warto zwrócić uwagę, że zmienne rozważa się lokalnie względem danego przejścia. Współbieżnie aktywne przejścia $t1$ i $t2$ mogą być rozważane przy wiązaniach, dla których za zmienną x podstawiono różne wartości, na przykład przejście $t2$ – przy wiązaniu $b = (a/x, 0/i, 0/j)$, a przejście $t1$ – przy wiązaniu $b_2 = (b/x)$.

W zbiorze $\mathcal{R}(M_0)$ znakowań osiągalnych ze znakowania początkowego znajduje się znakowanie martwe $M = (\emptyset, \emptyset, (2, 4), 2a + 4b, 3c, 3)$, które jest niejako stanem końcowym tej sieci. Po osiągnięciu stanu M otrzymuje się posortowany wielozbiór $2a + 4b + 3c$ (miejsce $p4$ zawiera znaczniki o wartościach a i b , miejsce $p5$ – znaczniki o wartości c) oraz uzyskuje się liczbę wystąpień znaczników o poszczególnych wartościach.

Rozważmy pewną modyfikację sieci \mathcal{N} . Sieć \mathcal{N}' powstaje z sieci \mathcal{N} w wyniku usunięcia miejsca $p1$. W takiej sytuacji przejście $t1$ będzie aktywne przy dowolnym znakowaniu należącym do zbioru $\mathcal{R}(M_0)$ (dla sieci \mathcal{N}' , $M_0 = (\emptyset, (0, 0), \emptyset, \emptyset, 0)$). Każde wykonanie tego przejścia powoduje umieszczenie w miejscu $p2$ jednego znacznika ze zbioru *Element* o losowo wybranej wartości. W zbiorze znakowań osiągalnych sieci \mathcal{N}' nie ma znakowania martwego. Ponadto zbiór $\mathcal{R}(M_0)$ jest zbiorem nieskończonym. W wyniku pracy tej sieci liczba znaczników w miejscach $p2$, $p4$ i $p5$ może wzrastać nieograniczenie, ale w miejscach $p3$ i $p6$ będzie zawsze pozostała dokładnie jeden znacznik.

10.3. Ograniczoność sieci

Większość zdefiniowanych w rozdz. 3 pojęć dotyczących własności sieci dotyczy również sieci kolorowanych. W odniesieniu do sieci kolorowanych są nadal ważne definicje takich pojęć jak: znakowanie osiągalne, zbiór znakowań osiągalnych ($\mathcal{R}(M)$), znakowanie odtwarzalne, sieć odtwarzalna, znakowanie odwracalne, sieć odwracalna oraz znakowanie własne.

² Wiązania są funkcjami określonymi na zbiorach skończonych; będziemy dla nich stosować zapis: $b = (\text{wartość}_1/\text{zmienna}_1, \text{wartość}_2/\text{zmienna}_2, \dots)$, tzn. za zmienną *zmienna₁* podstawiamy wartość *wartość₁* itd.

Z każdym znakowaniem M' osiągalnym ze znakowania M jest związany ciąg przejść, którego wykonanie prowadzi od znakowania M do M' . Ponieważ dla sieci kolorowanych wykonanie przejścia jest zawsze rozważane w kontekście pewnego wiązania, więc ciągi przejść są zapisywane jako ciągi par (t_i, b_i) , przy czym $t_i \in T$ oraz $b_i \in \mathcal{B}(t_i)$. Zbiór wszystkich takich ciągów przejść, które można wykonać, rozpoczynając od znakowania M , będziemy oznaczać symbolem $\mathcal{L}(M)$. Zbiór $\mathcal{L}(M)$ zawiera zarówno skończone, jak i nieskończone ciągi przejść.

Ograniczenie miejsc sieci kolorowanej można rozważyć w odniesieniu nie tylko do liczby znaczników, które mogą się zgromadzić w danym miejscu, ale również do wartości tych znaczników. Aby dokładniej określić, jakie znaczniki mogą zgromadzić się w danym miejscu sieci, podaje się dla miejsc zarówno górne, jak i dolne ograniczenia liczbowe oraz górne i dolne ograniczenia wielozbiorowe. Te ostatnie służą do jakościowego opisania możliwych znakowań miejsc sieci.

Definicja 10.7. Niech $\mathcal{N} = (\Sigma, P, T, A, \gamma, C, G, E, M_0)$ będzie siecią kolorowaną.

1. Miejsce $p \in P$ nazywamy *ograniczonym*, jeżeli jest spełniony warunek:

$$\exists k \in \mathbb{N} \quad \forall M \in \mathcal{R}(M_0): |M(p)| \leq k, \quad (10.16)$$

tz. przy dowolnym znakowaniu osiągalnym ze znakowania początkowego liczba znaczników w miejscu p jest nie większa niż k . Miejsce takie nazywamy również k -ograniczonym, a liczbę k jego górnym ograniczeniem. Miejsce 1-ograniczone nazywamy miejscem *bezpiecznym*. Miejsce p nazywamy *nieograniczonym*, jeżeli nie jest ograniczone. Sieć \mathcal{N} nazywamy k -ograniczoną, jeżeli wszystkie jej miejsca są k -ograniczone. Sieć 1-ograniczoną nazywamy siecią *bezpieczną*.

2. Liczbę $k \in \mathbb{N}$ nazywamy *dolnym ograniczeniem* miejsca p , jeżeli:

$$\forall M \in \mathcal{R}(M_0): |M(p)| \geq k. \quad (10.17)$$

3. Wielozbior $X \in 2^{C(p)^*}$ nazywamy *górnym ograniczeniem wielozbiorowym* miejsca p , jeżeli:

$$\forall M \in \mathcal{R}(M_0): M(p) \leq X. \quad (10.18)$$

4. Wielozbior $X \in 2^{C(p)^*}$ nazywamy *dolnym ograniczeniem wielozbiorowym* miejsca p , jeżeli:

$$\forall M \in \mathcal{R}(M_0): X \leq M(p). \quad (10.19)$$

Tabela 10.1. Ograniczenia miejsc sieci przedstawionej na rys. 10.1

miejsca	1	2	3	4
p_1	0	9	0	$2a + 4b + 3c$
p_2	0	9	0	$2a + 4b + 3c$
p_3	1	1	0	$(0, 0) + (0, 1) + (0, 2) + (0, 3) + (0, 4) + (1, 0) + (1, 1) + (1, 2) + (1, 3) + (1, 4) + (2, 0) + (2, 1) + (2, 2) + (2, 3) + (2, 4)$
p_4	0	6	0	$2a + 4b$
p_5	0	3	0	$3c$
p_6	1	1	0	$0 + 1 + 2 + 3$

W tabeli 10.1 przedstawiono górne i dolne ograniczenia liczbowe (kolumny 1 i 2) oraz wielozbiorowe (kolumny 3 i 4) miejsc sieci z rys. 10.1.

W przypadku sieci N' (opisana powyżej modyfikacja sieci na rys. 10.1) miejsca p_2 , p_4 i p_5 są nieograniczone. Ponadto dla żadnego z miejsc sieci N' nie istnieje skończone górnne ograniczenie wielozbiorowe.

Definicja 10.8. Miejsce $p \in P$ nazywamy *znakowanym*, jeżeli istnieje znakowanie $M \in \mathcal{R}(M_0)$, takie że $|M(p)| > 0$.

Zbiór miejsc $P' \subseteq P$ nazywamy *czystym*, jeżeli przy dowolnym znakowaniu $M \in \mathcal{R}(M_0)$ żaden znacznik nie leży na miejscu należącym do tego zbioru. W przeciwnym przypadku zbiór P' nazywamy *znakowanym*. ■

Wagi łuków sieci kolorowanej mogą być wyrażeniami zawierającymi zmienne. W zależności od wiązania wielozbioru znaczników usuwanych (lub dodawanych) w wyniku różnych wykonań tego samego przejścia mogą się różnić ze względu na rozmiary tych wielozbiorów. W związku z tym można zdefiniować pewne własności charakteryzujące liczbę znaczników *przepływających* przez łuki sieci.

Definicja 10.9. Niech dany będzie łuk $a \in A$ i przejście $t \in T$.

(1) Wyrażenie $E(a)$ nazywamy *jednolitym z wielokrotnością* $n \in \mathbb{N}$, jeżeli spełniony jest warunek:

$$\forall b \in \mathcal{B}(T(a)): |E(a)_b| = n, \quad (10.20)$$

tzn. dla dowolnego wiązania b przejścia $T(a)$ wartościowanie wyrażenia $E(a)$ jest wielozborem o rozmiarze n .

- (2) Przejście t nazywamy *jednolitym*, jeżeli wszystkie otaczające je łuki mają przypisane jednolite wyrażenia.
(3) Przejście t nazywamy *zachowawczym*, jeżeli jest spełniony warunek:

$$\forall b \in \mathcal{B}(t): \sum_{p \in In(t)} |E(p, t)_b| = \sum_{p \in Out(t)} |E(t, p)_b|, \quad (10.21)$$

tzn. jeżeli wykonanie przejścia t przy dowolnym wiązaniu $b \in \mathcal{B}(t)$ nie zmienia liczby znaczników w sieci.

- (4) Przejście t ma własność *maszyny stanowej*, jeżeli jest spełniony warunek:

$$\forall b \in \mathcal{B}(t): \sum_{p \in In(t)} |E(p, t)_b| = \sum_{p \in Out(t)} |E(t, p)_b| = 1. \quad (10.22)$$

Jeżeli wszystkie przejścia sieci mają własność maszyny stanowej, to mówimy, że sieć kolorowana ma własność *maszyny stanowej*. ■

Rozważmy ponownie sieć kolorowaną przedstawioną na rys. 10.1. Wszystkie łuki tej sieci mają przypisane jednolite wyrażenia z wielokrotnością 1. Wszystkie przejścia sieci są jednolite i zachowawcze, przy czym przejście t_1 ma dodatkowo własność maszyny stanowej.

Dla sieci kolorowanych można sformułować twierdzenie analogiczne do twierdzenia 3.1.

Twierdzenie 10.1. Jeżeli sieć kolorowana \mathcal{N} ma własność maszyny stanowej i jest znakowana k znacznikami, to jest k -ograniczona. ■

Definicja 10.10. Sieć \mathcal{N} nazywamy *zachowawczą*, jeżeli spełniony jest warunek:

$$\forall M \in \mathcal{R}(M_0): \sum_{p \in P} |M(p)| = \sum_{p \in P} |M_0(p)|, \quad (10.23)$$

tzn. jeżeli łączna liczba znaczników w sieci pozostaje stała dla każdego znakowania osiągalnego ze znakowania początkowego.

Przyjmijmy, że zbiór miejsc sieci kolorowanej \mathcal{N} jest zbiorem uporządkowanym, tzn. $P = \{p_1, p_2, \dots, p_n\}$. Sieć \mathcal{N} nazywamy *zachowawczą względem wektora wag* (w_1, w_2, \dots, w_n) , gdzie $w_i > 0$ dla $i = 1, 2, \dots, n$, jeżeli spełniony jest warunek:

$$\forall M \in \mathcal{R}(M_0): \sum_{i=1}^n w_i |M(p_i)| = \sum_{i=1}^n w_i |M_0(p_i)|, \quad (10.24)$$

tzn. jeżeli łączna ważona liczba znaczników w sieci pozostaje stała dla każdego znakowania osiągalnego ze znakowania początkowego. ■

Jeżeli możliwe jest podanie wektora wag tylko dla pewnego podzbioru zbioru P , to mówimy, że sieć jest *częściowo zachowawcza*. Bezpośrednio z powyższych definicji wynika następujący wniosek.

Wniosek 10.1. Sieć \mathcal{N} jest siecią zachowawczą wtedy i tylko wtedy, gdy wszystkie przejścia sieci są zachowawcze. ■

□ 10.4. Żywotność sieci

Żywotność sieci kolorowanych definiuje się podobnie jak żywotność sieci niskiego poziomu. Należy tylko wziąć pod uwagę fakt, że wykonanie przejścia dla sieci kolorowanych bada się zawsze w kontekście pewnego wiązania. Określone w definicji 3.8 własności żywotności można analogicznie sformułować dla przejścia rozważanego przy ustalonym wiązaniu.

Definicja 10.11. Niech t będzie przejściem sieci kolorowanej \mathcal{N} i niech b będzie wiązaniem przejścia t .

- (1) Przejście t nazywamy *martwym* przy wiązaniu b , jeżeli para (t, b) nie występuje w żadnym ciągu przejść należącym do zbioru $\mathcal{L}(M_0)$.
- (2) Przejście t nazywamy *potencjalnie wykonalnym* ($\mathcal{L}1$ -żywym) przy wiązaniu b , jeżeli istnieje ciąg przejść należący do zbioru $\mathcal{L}(M_0)$, w którym para (t, b) występuje co najmniej jeden raz.
- (3) Przejście t nazywamy *$\mathcal{L}2$ -żywym* przy wiązaniu b , jeżeli dla dowolnej liczby naturalnej k istnieje ciąg przejść należący do zbioru $\mathcal{L}(M_0)$, w którym para (t, b) występuje co najmniej k razy.
- (4) Przejście t nazywamy *$\mathcal{L}3$ -żywym* przy wiązaniu b , jeżeli istnieje ciąg przejść należący do zbioru $\mathcal{L}(M_0)$, w którym para (t, b) występuje nieskończonie wiele razy.
- (5) Przejście t nazywamy *żywym* ($\mathcal{L}4$ -żywym) przy wiązaniu b , jeżeli jest ono potencjalnie wykonalne przy wiązaniu b dla każdego znakowania $M \in \mathcal{R}(M_0)$. ■

Bardzo często istotne jest sprawdzenie, czy dane przejście będzie miało szansę na ponowne wykonanie bez względu na to, przy jakim wiązaniu to będzie możliwe. Definicję 10.11 można tak sformułować, by uwzględniała przejścia bez odwoływania się do konkretnych wiązań.

Definicja 10.12. Niech $t \in T$ będzie przejściem sieci kolorowanej \mathcal{N} .

- (1) Przejście t nazywamy *martwym* ($\mathcal{L}0$ -żywym), jeżeli żadna para (t, b) , przy czym $b \in \mathcal{B}(t)$, nie występuje w żadnym ciągu przejść należącym do zbioru $\mathcal{L}(M_0)$.
- (2) Przejście t nazywamy *potencjalnie wykonalnym* ($\mathcal{L}1$ -żywym), jeżeli istnieje wiązanie $b \in \mathcal{B}(t)$ oraz ciąg przejść należący do zbioru $\mathcal{L}(M_0)$, w którym para (t, b) występuje co najmniej jeden raz.
- (3) Przejście t nazywamy $\mathcal{L}2$ -żywym, jeżeli dla dowolnej liczby naturalnej k istnieją wiązania $b_1, b_2, \dots, b_n \in \mathcal{B}(t)$ oraz ciąg przejść należący do zbioru $\mathcal{L}(M_0)$, w którym pary (t, b_i) , przy czym $i = 1, 2, \dots, n$, występują łącznie co najmniej k razy.
- (4) Przejście t nazywamy $\mathcal{L}3$ -żywym, jeżeli istnieją wiązania $b_1, b_2, \dots, b_n \in \mathcal{B}(t)$ oraz ciąg przejść należący do zbioru $\mathcal{L}(M_0)$, w którym pary (t, b_i) , przy czym $i = 1, 2, \dots, n$, występują łącznie nieskończonie wiele razy.
- (5) Przejście t nazywamy żywym ($\mathcal{L}4$ -żywym), jeżeli t jest potencjalnie wykonalne dla każdego znakowania $M \in \mathcal{R}(M_0)$.

Sieć \mathcal{N} nazywamy $\mathcal{L}k$ -żywą, jeżeli każde przejście sieci jest $\mathcal{L}k$ -żywe dla $k = 0, 1, \dots, 4$. Sieć \mathcal{N} nazywamy dokładnie $\mathcal{L}k$ -żywą, jeżeli jest ona $\mathcal{L}k$ -żywa, ale nie jest $\mathcal{L}(k+1)$ -żywa. Sieć \mathcal{N} nazywamy strukturalnie żywą, jeżeli istnieje znakowanie początkowe M_0 , przy którym sieć ta jest żywa. ■

Warunki żywotności sformułowane w definicji 10.12 są zdecydowanie słabsze niż warunki w definicji 10.11. Żywotność przejścia dopuszcza istnienie wiązań tego przejścia, które nigdy nie zostaną wykonane. Można wprowadzić bardziej restrykcyjną definicję żywotności przejść.

Definicja 10.13. Niech $t \in T$ będzie przejściem sieci \mathcal{N} . Przejście t nazywamy silnie $\mathcal{L}k$ -żywym, jeżeli jest ono $\mathcal{L}k$ -żywe przy dowolnym swoim wiązaniu b , dla $k = 0, 1, \dots, 4$. Sieć \mathcal{N} nazywamy silnie $\mathcal{L}k$ -żywą, jeżeli każde przejście sieci jest silnie $\mathcal{L}k$ -żywe dla $k = 0, 1, \dots, 4$. Sieć \mathcal{N} nazywamy dokładnie silnie $\mathcal{L}k$ -żywą, jeżeli jest ona silnie $\mathcal{L}k$ -żywa, ale nie jest silnie $\mathcal{L}(k+1)$ -żywa. ■

Znakowanie M nazywamy żywym, jeżeli sieć $\mathcal{N} = (\Sigma, P, T, A, \gamma, C, G, E, M)$ jest żywa, natomiast znakowanie M nazywamy martwym, jeżeli taka sieć \mathcal{N} jest martwa.

Implikacje przedstawione we wzorze (3.11) zachowują ważność zarówno wtedy, gdy rozważamy żywotność przejść przy konkretnym wiązaniu, jak i żywotność przejść czy też silną żywotność przejść.

Definicja 10.14. Miejsce $p \in P$ nazywamy żywym, jeżeli dla dowolnego znakowania $M \in \mathcal{R}(M_0)$ istnieje znakowanie $M' \in \mathcal{R}(M)$, takie że $|M'(p)| > 0$. Sieć \mathcal{N} nazywamy żywą ze względu na miejsca, jeżeli wszystkie jej miejsca są żywe. ■

Podobnie jak w sieciach niskiego poziomu, żywotność miejsca oznacza, że zawsze ma ono szansę ponownie zawierać znaczniki, a żywotność przejścia oznacza, że zawsze ma ono szansę ponownie być aktywne. Gdy przejście nie jest przejściem żywym, wówczas nie oznacza to, że jest martwe. Może się bowiem zdarzyć, że przy zadanym znakowaniu początkowym, przejście będzie mogło być wykonane kilka razy, a dopiero później straci szansę na to, by ponownie być aktywne. Martwe przejścia można usunąć z sieci bez zmiany jej zachowania; sieć i tak działa jakby ich nie było. Jeżeli sieć nie jest żywa to nie znaczy, że jest martwa. Jest to związane z faktem, że poszczególne jej przejścia mogą mieć różne poziomy żywotności.

Jeżeli przy pewnym znakowaniu M są aktywne dwa przejścia, ale wykonanie dowolnego z nich powoduje utratę aktywności przez drugie, to mówimy, że przejścia te pozostają ze sobą w konflikcie. Warto zwrócić uwagę, że aby w sieci kolorowanej wystąpiła sytuacja konfliktowa, muszą być spełnione mocniejsze warunki niż w przypadku sieci niskiego poziomu. Jeżeli dwa przejścia mają wspólne miejsce wejściowe, to aby doszło do sytuacji konfliktowej, muszą one rywalizować o ten sam znacznik.

Klasyfikacja żywotności przedstawiona w definicjach 10.11 i 10.12 nie odzwierciedla dobrze regularności działania sieci, nawet w odniesieniu do sieci żywych. Szansa na ponowne wykonanie przejścia nie oznacza jednak, że przejście to kiedykolwiek zostanie wykonane. Może się zdarzyć, że ze względu na występujące konflikty, przejście jest nieskończonym wiele razy aktywne, ale nigdy nie zostaje wykonane. Dla sieci kolorowanych można wprowadzić pojęcie sprawiedliwości (por. definicję 3.10), które oznacza żądanie równomiernego wykonywania sieci.

Definicja 10.15. Ciąg przejść $\alpha \in \mathcal{L}(M_0)$ nazywamy sprawiedliwym, jeżeli jest on skończony lub jest nieskończony i dla każdego przejścia $t \in T$ istnieją wiązania $b_1, b_2, \dots, b_n \in \mathcal{B}(t)$, takie że pary (t, b_i) , przy czym $i = 1, 2, \dots, n$, występują łącznie nieskończonym wiele razy w tym ciągu. Ciąg przejść $\alpha \in \mathcal{L}(M_0)$ nazywamy silnie sprawiedliwym, jeżeli jest on skończony lub jest nieskończony i każda para $(t, b) \in \mathcal{B}$ występuje nieskończonym wiele razy w tym ciągu.

Sieć \mathcal{N} nazywamy sprawiedliwą, jeżeli każdy ciąg przejść $\alpha \in \mathcal{L}(M_0)$ jest sprawiedliwy. Sieć \mathcal{N} nazywamy silnie sprawiedliwą, jeżeli każdy ciąg przejść $\alpha \in \mathcal{L}(M_0)$ jest silnie sprawiedliwy. ■

10.5. Hierarchiczne sieci kolorowane

Hierarchiczne sieci kolorowane umożliwiają tworzenie dużych modeli za pomocą łączenia skończonej liczby sieci niehierarchicznych, zwanych *stronami*. Strony sieci mogą być łączone przy użyciu dwóch konstrukcji, określanych jako *podstawiane przejścia* i *fuzje miejsc*.

Podstawianie przejść pozwala połączyć z przejściem sieci i otaczającymi go łukami bardziej złożoną sieć (hierarchiczną lub nie). Strona zawierająca podstawiane przejście jest nazywana *nadstroną*, a strona zawierająca podsieć, *podstroną*. Podstrona może zawierać kolejne podstawiane przejścia, można więc mówić o bezpośrednich i pośrednich podstronach³. Strukturę sieci przedstawia się na *grafie hierarchii stron*. Węzły tego grafu reprezentują strony sieci, łuki zaś, etykietowane nazwami podstawianych przejść, reprezentują zależności nadstrona–podstrona.

Fuzja jest zbiorem miejsc, o których zakładamy, że są nieroróżnicalne, tzn. pojedynczo reprezentują jeden element (obiekt), lecz są narysowane jako indywidualne miejsca. Jeśli jakiś znacznik jest dodawany do (lub usuwany z) jednego z miejsc należących do danej fuzji, to jest automatycznie dodawany do (lub usuwany z) każdego z pozostałych miejsc tej fuzji. Miejsca biorące udział w fuzji mogą należeć do jednej lub do wielu stron. Jeżeli wszystkie miejsca należą do jednej strony, która występuje w sieci tylko raz, to istnienie fuzji miejsc pozwala jedynie uniknąć zbyt wielu przecinających się łuków. Jeżeli natomiast miejsca należą do różnych stron, to ich fuzja umożliwia połączenie tych stron i skonstruowanie spójnego modelu.

Można wyróżnić trzy typy fuzji: *fuzję globalną (FG)*, która dopuszcza, by miejsca biorące w niej udział należały do różnych stron, oraz *fuzję stronicową (FP)* i *fuzję instancji strony (FI)*. Fuzje typu *FP* i *FI* dopuszczają, by miejsca biorące w nich udział należały tylko do jednej strony, przy czym w pierwszym przypadku miejsca te mogą należeć do różnych instancji (kopii) tej strony, a zatem poszczególne instancje współdzielą dane miejsce wynikowe; w drugim przypadku miejsca mogą należeć tylko do jednej instancji danej strony, a zatem każda instancja tej strony ma własne miejsce wynikowe.

Strony hierarchicznej sieci kolorowanej są sieciami niehierarchicznymi postaci:

$$(\Sigma_s, P_s, T_s, A_s, \gamma_s, C_s, G_s, E_s, M_{0s}). \quad (10.25)$$

³ Określenie *podstrona* będzie się odnosić do bezpośrednich podstron, chyba że zostanie powiedziane inaczej.

Niech S oznacza zbiór stron sieci hierarchicznej. Zbiory Σ , P , T i A można zdefiniować globalnie dla całej sieci:

$$\Sigma = \bigcup_{s \in S} \Sigma_s, \quad P = \bigcup_{s \in S} P_s, \quad T = \bigcup_{s \in S} T_s, \quad A = \bigcup_{s \in S} A_s. \quad (10.26)$$

Podobnie można zdefiniować również globalne funkcje: γ, C, G, E, M_0 , na przykład:

$$C: P \rightarrow \Sigma \wedge \forall s \in S \quad \forall p \in P_s: C(p) = C_s(p). \quad (10.27)$$

Pojedyncza strona może w sieci występuwać kilkakrotnie. Poszczególne *kopie* takiej strony nazywane są jej *instancjami*. W sieci hierarchicznej istnieje więc wielozbiór stron, przy czym współczynniki poszczególnych elementów tego wielozbioru informują o liczbie instancji danej strony w sieci. Każda instancja strony ma własną nazwę i numer. Jeżeli dana strona występuje w sieci więcej niż jeden raz, to każda jej instancja zawiera własne znakowanie, które jest niezależne od znakowań w pozostałych instancjach tej strony. Ponieważ pojedyncza strona może mieć kilka instancji w ramach pojedynczej hierarchicznej sieci, więc każde miejsce na takiej stronie, każde przejście i każdy łuk mają wówczas również wiele instancji.

Przyjrzymy się dokładniej mechanizmowi podstawiania przejść. Niech t będzie podstawianym przejściem. Miejsca otaczające przejście t (miejsca ze zbioru $In(t) \cup Out(t)$) nazywamy *miejscami gniazdowymi* (lub krótko *gniazdami*). W zależności od sposobu połączenia gniazda z podstawianym przejściem można mówić o *gniazdach wejściowych* ($In(t) - Out(t)$), *gniazdach wyjściowych* ($Out(t) - In(t)$) i *gniazdach wejściowo-wyjściowych* ($In(t) \cap Out(t)$).

Gniazda łączą się z *miejscami portowymi* (lub krótko *portami*) z bezpośredniej podstrony. Porty są wyróżnionymi miejscami, którym przypisuje się dodatkowo typ portu. W wyniku takiego postępowania mówimy o: *portach wejściowych*, *portach wyjściowych*, *portach wejściowo-wyjściowych* i *portach nieokreślonych*, oznaczając je odpowiednio jako *In*, *Out*, *I/O* oraz *Gen* (ang. *general*).

Strona może zawierać porty nawet wtedy, kiedy nie jest podstroną żadnej innej strony. Określenie wówczas miejsca strony jako miejsca portowego nie wpływa w żaden sposób na semantykę sieci.

Opis połączenia podstawianego przejścia z przypisaną mu podstroną jest w rzeczywistości określeniem relacji dwuargumentowej, której elementami są pary (*gniazdo, port*). Relacja taka łączy miejsca tego samego typu, tzn. gniazdo wejściowe z portem wejściowym itd. Wyjątek stanowią porty nieokreślone, które mogą być łączone z dowolnymi gniazdami. Możliwe jest połączenie kilku portów z jednym

gniazdem i na odwrót. Najczęściej jednak każde gniazdo ma wzajemnie jednoznacznie przypisany jeden port.

Definicja 10.16. Strukturę $N = (S, SN, SA, PN, PT, PA, FS, FT, PP)$ nazywamy *hierarchiczną siecią kolorowaną*, jeżeli są spełnione warunki:

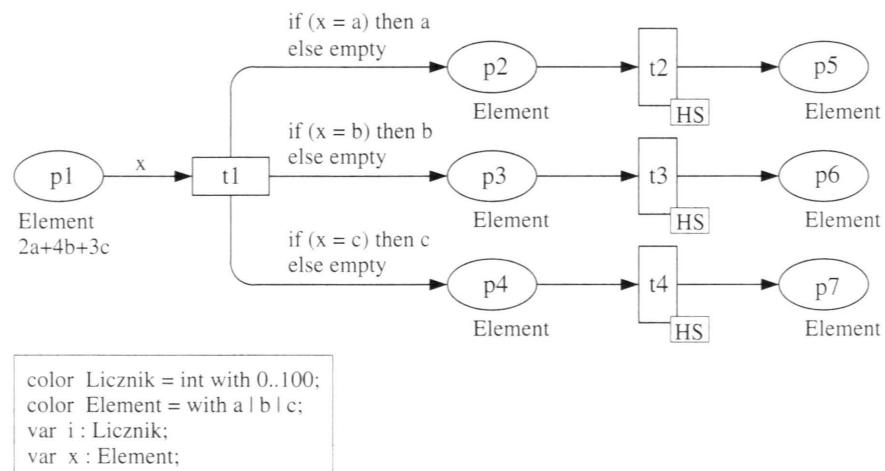
- (1) S jest zbiorem stron, z których każda jest siecią niehierarchiczną i elementy zbiorów P_s , T_s i A_s dla dowolnych dwóch stron są parami rozłączne.
- (2) $SN \subseteq T$ jest zbiorem podstawianych przejść.
- (3) SA jest funkcją przydziału stron, która każdemu z podstawianych przejść przypisuje odpowiadającą mu podstronę w ten sposób, że żadna strona nie jest własną podstroną (zarówno bezpośrednią, jak i pośrednią).
- (4) $PN \subseteq P$ jest zbiorem portów.
- (5) PT jest funkcją typów portów, która każdemu portowi przypisuje odpowiedni typ (In , Out , I/O , Gen).
- (6) PA jest funkcją przydziału portów, która każdemu z podstawianych przejść przypisuje relację dwuargumentową spełniającą warunki:
 - Gniazda są połączone z portami, tzn.: $\forall t \in SN: PA(t) \subseteq (In(t) \cup Out(t)) \times PN_{SA(t)}$, przy czym $PN_{SA(t)}$ oznacza porty na bezpośredniej podstronie przejścia t .
 - Gniazda i porty są takich samych typów, przy czym porty nieokreślone można łączyć z dowolnymi gniazdami.
 - Połączone miejsca mają taki sam typ i znakowania początkowe.
- (7) $FS \subseteq 2^P$ jest skończonym zbiorem fuzji miejsc. Miejsca należące do danej fuzji muszą mieć identyczne typy i takie same znakowania początkowe.
- (8) FT jest funkcją typów fuzji, przypisującą każdej fusji jej typ (FG , FP , FI), przy czym fusja może mieć przypisany typ FP lub FI tylko wtedy, gdy wszystkie jej miejsca należą do jednej strony.
- (9) PP jest wielozborem stron głównych. ■

Aby można było dokładnie określić miejsce każdej instancji strony s w hierarchicznym modelu, każda instancja jest opisywana przez trzy elementy: nazwę strony głównej, numer instancji tej strony i ciąg podstawianych przejść prowadzących od tej strony głównej do strony s . W opisie stron głównych ostatni element takiej trójki jest ciągiem pustym.

Dynamikę sieci hierarchicznych definiuje się analogicznie do dynamiki sieci niehierarchicznych, choć zapis formalny tych definicji jest nieco bardziej skomplikowany. Należy wziąć pod uwagę fakt, że przy określaniu aktywności przejść miejsca

należące do fuzji miejsc lub tworzące parę (*gniazdo, port*) są traktowane jak pojedyncze miejsca⁴.

Rozważmy przykład prostej hierarchicznej sieci kolorowanej, będącej modyfikacją sieci pokazanej na rys. 10.1. Jedyną stroną główną tego modelu (*Sortowanie*) przedstawiono na rys. 10.2 ($PP = \{\text{Sortowanie}\}$). Przejście t_1 jest używane do sortowania znaczników pobieranych z miejsca p_1 . Przy każdym wykonaniu przejścia t_1 jest pobierany jeden znacznik, który w zależności od jego wartości jest umieszczany w jednym z miejsc p_2 , p_3 lub p_4 . Pozostałe przejścia są przejściami podstawianymi; zaznaczono to za pomocą etykiet *HS* (ang. *hierarchy substitution*), które umieszczone obok podstawianych przejść ($SN = \{t_2, t_3, t_4\}$). Wszystkie te przejścia połączone z podstroną (*Zliczanie*) przedstawioną na rys. 10.3 ($SA(t_2) = SA(t_3) = SA(t_4) = Zliczanie$, $S = \{\text{Sortowanie}, \text{Zliczanie}\}$).

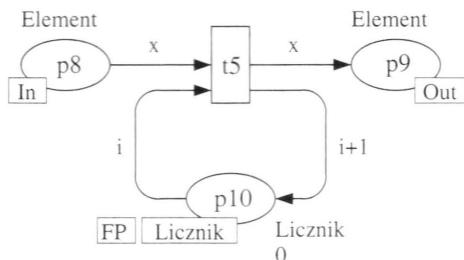


Rys. 10.2. Strona główna *Sortowanie*

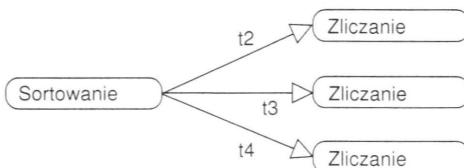
Przedstawiona na rys. 10.3 podstrona *Zliczanie* zawiera dwa miejsca portowe: port wejściowy p_8 i port wyjściowy p_9 . Ponieważ w modelu występują trzy instancje rozważanej podstrony, to każde z miejsc p_8 , p_9 i p_{10} ma trzy instancje. Podobnie trzy instancje ma przejście t_5 i każdy z łuków tej podstrony.

Rozważmy podstawiane przejście t_2 . Miejsce p_2 jest gniazdem wejściowym i jest skojarzone z portem wejściowym p_8 , a właściwie z jedną z instancji tego

⁴ W formalnym zapisie definicji aktywności przejść wprowadza się relację równoważności określoną na zbiorze instancji miejsc i zamiast na pojedynczych miejscach, operuje się na klasach abstrakcji tej relacji (zob. pracę [32]).

Rys. 10.3. Podstrona *Zliczanie*

portu⁵. Podobnie gniazdo wyjściowe p_5 jest skojarzone z portem wyjściowym p_9 ($PA(t_2) = \{(p_2, p_8), (p_5, p_9)\}$). W przypadku podstawianego przejścia t_3 gniazda p_3 i p_6 są skojarzone z innymiinstancjami portów p_8 i p_9 .



Rys. 10.4. Graf hierarchii stron

Miejsce p_{10} pełni funkcję licznika przesłanych znaczników. Miejsce to należy do fuzji stronicowej o nazwie *Licznik*, a zatem jest współdzielone przez wszystkie instancje strony *Zliczanie*. Oznacza to, że wszystkie instancje przejścia t_5 korzystają ze wspólnego miejsca p_{10} , czyli zliczane są wszystkie znaczniki, nie zaś poszczególne ich wartości. Aby uzyskać efekt podobny do przypadku sieci na rys. 10.1, należy usunąć fuzję miejsc *Licznik* lub zmienić jej typ na fuzję instancji strony. Graf hierarchii stron dla rozważanego modelu przedstawiono na rys. 10.4.

10.6. Modelowanie z zastosowaniem sieci kolorowanych

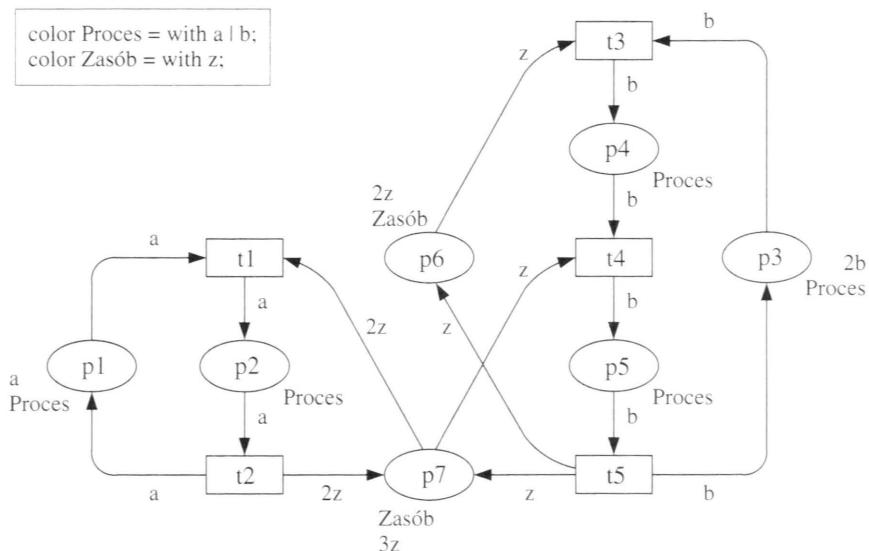
W tym podrozdziale omówimy przykłady zastosowania sieci kolorowanych do modelowania systemów współbieżnych. Najpierw zajmiemy się przykładem systemu,

⁵ Jeżeli nie będzie to prowadziło do nieporozumień, to będziemy używać określeń: miejsce, przejście, strona itd., zamiast określeń: instancja miejsca, instancja przejścia, instancja strony itd.

w którym współbieżnie działające procesy rywalizują o współdzielone zasoby. Model ten, stworzony w postaci sieci niehierarchicznej ze względu na niewielką liczbę osiąganych stanów, wykorzystamy przy omawianiu grafów osiągalności dla sieci kolorowanych. Hierarchiczną sieć kolorowaną zastosujemy do zbudowania modelu funkcjonowania płatnej autostrady.

10.6.1. Model współzawodnictwa procesów

Rozważmy system przedstawiony na rys. 10.5, w którym o dwa źródła zasobów rywalizują trzy procesy: jeden proces typu *a* i dwa procesy typu *b*. Proces *a* może znajdować się w jednym z dwóch etapów obliczeniowych (miejscza *p1* i *p2*), przy czym jeżeli znajduje się on w drugim etapie (miejsce *p2*), to wykorzystuje dwa zasoby z pobrane z miejsca *p7*. Procesy typu *b* mogą się znajdować w trzech różnych etapach obliczeniowych (miejscza *p3*, *p4* i *p5*), przy czym w zależności od etapu korzystają z różnej liczby zasobów pobieranych z miejsc *p6* i *p7*.



Rys. 10.5. Model współzawodnictwa procesów

W rozważanym modelu nie występują zmienne, a zatem każde przejście może być wykonywane tylko w jednym ze swoich wiązań⁶. Zbiór znakowań osiągal-

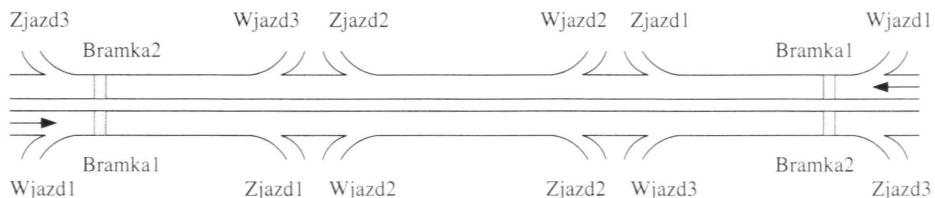
⁶ W takich sytuacjach mamy do czynienia z trywialnymi wiązaniami, gdyż dziedziną dowolnego wiązania *b* jest zbiór pusty.

nich $\mathcal{R}(M_0)$ zawiera 11 elementów (zob. rozdz. 11). Sieć jest żywa, ale nie jest sprawiedliwa, na przykład możemy naprzemienne wykonywać tylko przejścia t_1 i t_2 .

Dolnym ograniczeniem liczbowym dla wszystkich miejsc sieci jest zero, a wielozbiorowym, zbiór pusty. Górnego ograniczenie liczbowe dla miejsc p_1 i p_2 wynosi jeden, dla miejsc od p_3 do p_6 , dwa, a dla miejsca p_7 , trzy. Górnego ograniczenia wielozbiorowe dla kolejnych miejsc sieci wynoszą odpowiednio: $a, a, 2b, 2b, 2b, 2z, 3z$. Wszystkie znakowania osiągalne w sieci są jej znakowaniami własnymi.

10.6.2. Model autostrady

Jako drugi przykład rozważymy model płatnej autostrady, której schemat przedstawiono na rys. 10.6. W modelu tym uwzględniono zarówno ruch pojazdów przemieszczających się po autostradzie, jak również opłaty za przejazd wnoszone przez kierowców.

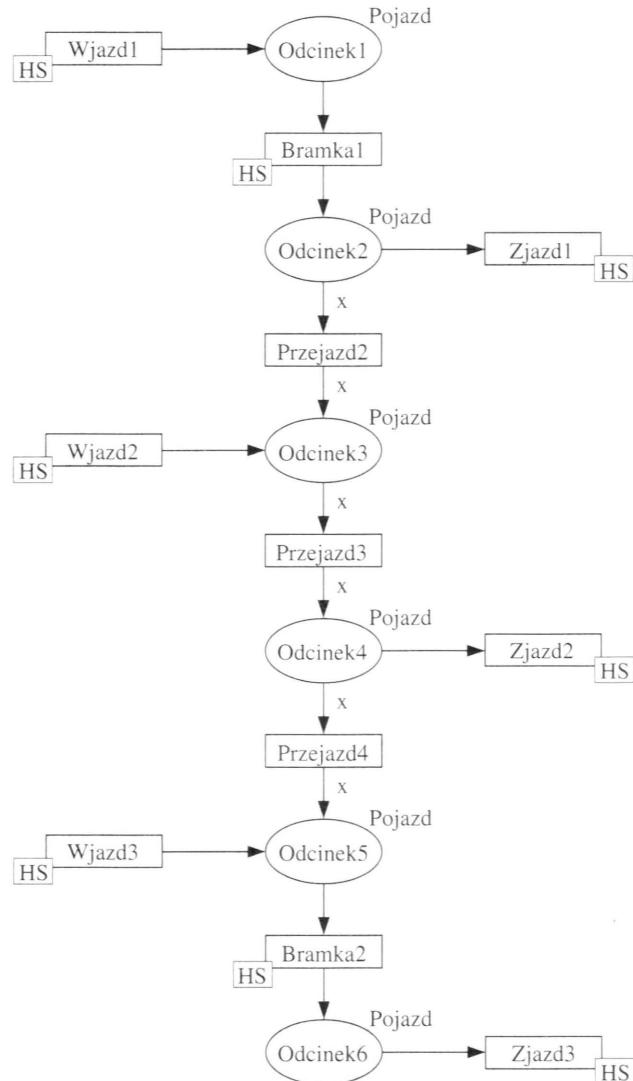


Rys. 10.6. Schemat autostrady

Każda z nitek autostrady zawiera trzy wjazdy i trzy zjazdy. Za pierwszym wjazdem i przed ostatnim zjazdem z autostrady umieszczono bramki, przy których pobierane są opłaty za przejazd autostradą (połowa opłaty przy przejeździe przez każdą z bramek, przy czym samochody ciężarowe uiszczają podwójną opłatę). Pojazdy, które wjeżdżają na autostradę drugim wjazdem i zjeżdżają z niej, korzystając z drugiego zjazdu, nie uiszczają opłaty za przejazd. Wjazdy, zjazdy i bramki poszerzono w taki sposób, by powstał model autostrady składający się z dwóch symetrycznych części.

Stroną główną rozważanego modelu będzie strona reprezentująca jedną z nitek autostrady. Strona ta wystąpi w modelu dwukrotnie ze względu na dwie symetryczne części, z których składa się system. Modelując ruch na autostradzie podzielimy ją na odcinki. Liczba samochodów znajdujących się na danym odcinku będzie reprezentowana przez odpowiednią liczbę znaczników występujących w miejscu reprezentującym ten odcinek. Podział na odcinki będzie wyznaczony przez bramki,

wjazdy i zjazdy, tzn.: *Odcinek1* oznacza fragment autostrady między *Wjazdem1* a *Bramką1*, *Odcinek2* oznacza fragment autostrady między *Bramką1* a *Zjazdem1*, *Odcinek3* oznacza fragment autostrady między *Zjazdem1* a *Wjazdem2* itd. Stronę główną o nazwie *Nitka* przedstawiono na rys. 10.7.



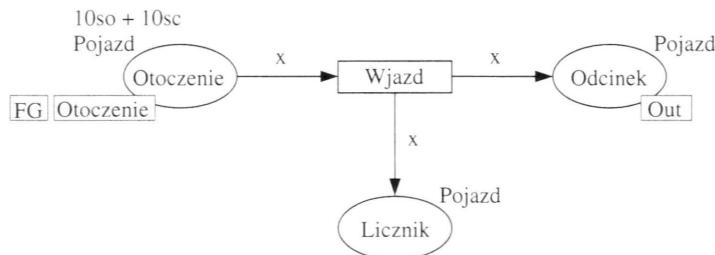
Rys. 10.7. Strona główna modelu autostrady

W modelu występują dwa typy wyliczeniowe i jedna zmienna:

```
color Pojazd = with so | sc;
color Waluta = with m;
var x : Pojazd;
```

Upraszczając model, typy pojazdów poruszających się po autostradzie ograniczono do samochodów osobowych i ciężarowych. Pojazdy te są reprezentowane przez odpowiednie stałe typu *Pojazd*. Stała *m* typu *Waluta* oznacza monetę będącą opłatą za przejazd przez bramkę na autostradzie (opłata za przejazd przez samochód ciężarowy wynosi dwie monety).

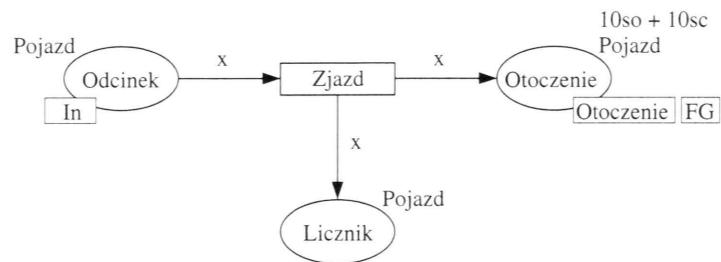
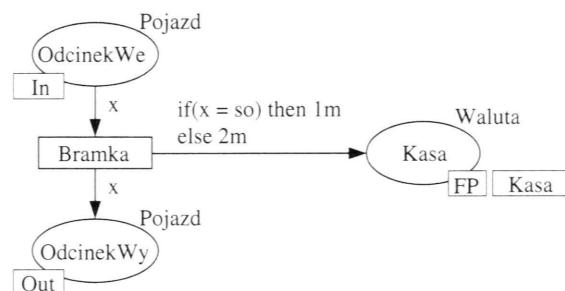
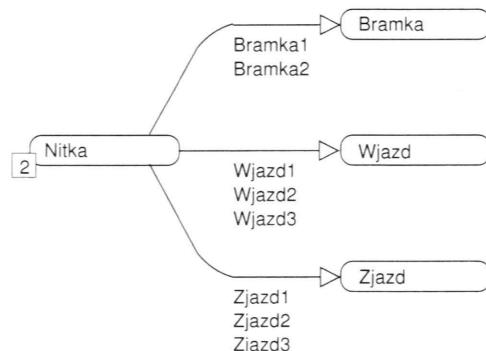
Zajmijmy się stroną główną, zwaną *Nitką*, przedstawioną na rys. 10.7. Przejścia o nazwach: *Przejazd2*, *Przejazd3* i *Przejazd4* oznaczają przejazd odpowiedniego odcinka autostrady i powodują przesunięcie znacznika między dwoma miejscami sieci. Pozostałe przejścia są przejściami podstawianymi.



Rys. 10.8. Podstrona *Wjazd* modelu autostrady

Podstronę *Wjazd* połączoną z przejściami: *Wjazd1*, *Wjazd2* i *Wjazd3* przedstawiono na rys. 10.8. Dodatkowe miejsce, *Otoczenie*, służy do ograniczenia liczby pojazdów poruszających się w modelowanym systemie. Miejsce to należy do fuzji globalnej (o tej samej nazwie) i występuje również na podstronie *Zjazd*. Przy znakowaniu początkowym w miejscu *Otoczenie* umieszczono 20 znaczników reprezentujących wszystkie pojazdy, które będą poruszały się po autostradzie. Miejsce *Liczni*k jest używane do zliczania pojazdów wjeżdżających danym wjazdem na autostradę. Miejsce to nie należy do fuzji miejsc, a zatem każda instancja strony *Wjazd* ma własną instancję tego miejsca (własny licznik). Podobną strukturę ma podstrona *Zjazd* przedstawiona na rys. 10.9.

Na rysunku 10.10 przedstawiono podstronę *Bramka*, która jest połączona z przejściami *Bramka1* i *Bramka2*. Podstrona ta zawiera dodatkowe miejsce, *Kasa*, należące do fuzji stronicowej, którego znakowanie reprezentuje łączną kwotę opłat uiszczych za korzystanie z autostrady.

Rys. 10.9. Podstrona *Zjazd* modelu autostradyRys. 10.10. Podstrona *Bramka* modelu autostrady

Rys. 10.11. Graf hierarchii stron dla modelu autostrady

Na rysunku 10.11 przedstawiono graf hierarchii stron dla rozważanego modelu. Dodatkowa etykieta przy węźle reprezentującym stronę *Nitka* oznacza liczbę instancji tej strony. Liczba instancji dla pozostałych stron jest konsekwencją struktury modelu. Strona *Bramka* ma 4 instancje, a strony *Wjazd* i *Wyjazd* mają po 6 instancji. Strony modelu są ponadto połączone przez dwie fuzje miejsc, dając w wyniku spójny graf sieci.

Zbiór znakowań osiągalnych rozważanej sieci jest zbiorem nieskończonym (miejscą *Licznik* i *Kasa* są nieograniczone). Sieć jest silnie żywa, ale nie jest sprawiedliwa. Dolnym ograniczeniem liczbowym dla wszystkich miejsc sieci jest zero, a wielozbiorowym, zbiór pusty. Górnne ograniczenie liczbowe dla miejsc *Odcinek* wynosi 20, a górne ograniczenie wielozbiorowe dla tych miejsc wynosi $10so + 10sc$. Znakowanie początkowe sieci nie jest odtwarzalne (ze względu na miejsca *Licznik* i *Kasa*).

10.7. Podsumowanie

Opisane w tym rozdziale kolorowane sieci Petriego są najbardziej popularną klasą sieci wysokiego poziomu. Możliwość operowania różnymi typami znaczników ułatwia modelowanie bardziej złożonych systemów, a jednocześnie pozwala tworzyć modele lepiej opisujące rzeczywiste systemy współbieżne. Nie bez znaczenia jest również fakt, że modele mogą mieć postać sieci hierarchicznych. Pozwala to na dokonanie podziału złożonego systemu na części i opracowywanie go na przykład metodą *od ogółu do szczegółu*. Jednocześnie możliwość wielokrotnego wykorzystywania tych samych stron pozwala skrócić czas konstruowania takiego modelu.

Wprowadzenie różnych typów znaczników pociąga za sobą konieczność ponownego zdefiniowania wielu pojęć opisujących własności sieci. Analizując ograniczoność sieci, rozważa się ograniczenie nie tylko liczby znaczników w poszczególnych miejscach, ale również wartości tych znaczników. Podobnie w definicji żywotności czy też sprawiedliwości sieci muszą być uwzględnione wiązania, przy których wykonywane są przejścia. W wyniku tych zmian nie można metod analizy przedstawionych w rozdz. 5 stosować bezpośrednio do sieci kolorowanych.

Analiza sieci kolorowanych

Analizę dowolnej klasy sieci Petriego można opierać na grafach osiągalności, które stanowią uporządkowaną reprezentację zbioru osiągalnych znakowań. Grafy osiągalności dla sieci kolorowanych definiuje się tak samo jak dla sieci uogólnionych (zob. rozdz. 5). Do zredukowania rozmiaru grafu przedstawiającego zbiór osiągalnych znakowań sieci używa się grafów osiągalności z klasami równoważności (tzw. grafów relacyjnych). W celu zbudowania takiego grafu definiuje się dwie relacje równoważności określone odpowiednio na zbiorach $\mathcal{R}(M_0)$ i \mathcal{B} . Węzły i łuki takiego grafu reprezentują klasy abstrakcji tych relacji.

Jak dla sieci uogólnionych, tak i dla CP-sieci definiuje się niezmienniki miejsc i przejść. Niezmienniki takie są zapisywane w postaci równań, które wyrażają pewne stałe zależności między znakowaniami miejsc sieci lub ciągami wykonywanych przejść. Biorąc pod uwagę fakt, że macierz incydencji sieci kolorowanej nie jest macierzą liczbową, wyznaczanie takich niezmienników bez użycia odpowiedniego oprogramowania jest w większości przypadków niemożliwe.

W tym rozdziale omówimy grafy osiągalności, grafy relacyjne oraz niezmienniki miejsc i przejść wykorzystywane do analizy kolorowanych sieci Petriego (zob. prace: [13], [32], [35] i [36]). Ze względu na prostszą formę zapisu, wszystkie przytoczone definicje odnoszą się do niehierarchicznych sieci kolorowanych. Dla sieci hierarchicznych odpowiednie pojęcia definiuje się analogicznie, aczkolwiek zapis tych definicji jest zdecydomniej bardziej skomplikowany, ponieważ trzeba uwzględnić instancje miejsc, przejść itd.

11.1. Grafy osiągalności

Grafy osiągalności dla sieci kolorowanych konstruuje się tak jak dla sieci uogólnionych (zob. podrozdz. 5.1). Węzłami grafu są znakowania sieci, a łuki prowadzące od jednego znakowania do drugiego mają przypisane etykiety zawierające infor-

mację o przejściu (i wiązaniu), którego wykonanie powoduje zmianę znakowania z pierwszego na drugie.

Niech dana będzie sieć kolorowana $\mathcal{N} = (\Sigma, P, T, A, \gamma, C, G, E, M_0)$.

Definicja 11.1. Grafem osiągalności sieci \mathcal{N} nazywamy graf $\mathcal{G} = (V, A, \gamma)$ etykietowany nad zbiorem \mathcal{B} , spełniający warunki:

- (1) $V = \mathcal{R}(M_0)$ – jest zbiorem węzłów grafu.
- (2) $A = \left\{ (M, (t, b), M') : M, M' \in \mathcal{R}(M_0) \wedge M \xrightarrow{(t,b)} M' \right\}$
jest zbiorem łuków, przy czym etykieta (t, b) oznacza, że wykonanie przejścia t przy wiązaniu b powoduje zmianę znakowania z M na M' .
- (3) $\forall a = (M, (t, b), M') \in A : \gamma(a) = (M, M')$. ■

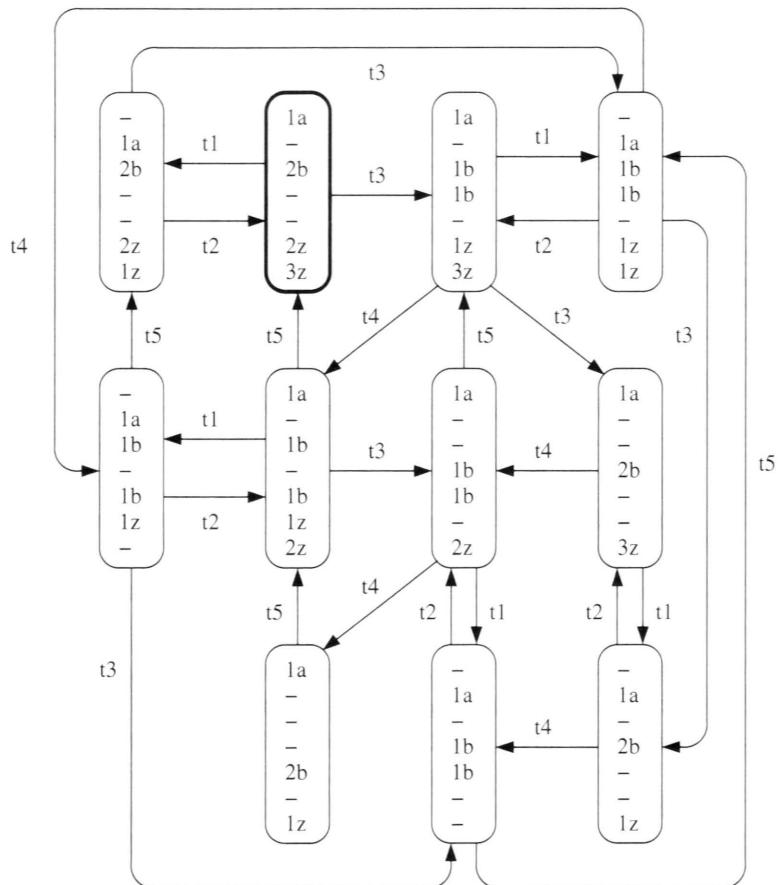
Rozważmy sieć pokazaną na rys. 10.5, będącą modelem współzawodnictwa procesów. Graf osiągalności budujemy podobnie jak dla sieci niskiego poziomu. Pierwszym węzłem jest znakowanie początkowe $M_0 = (a, \emptyset, 2b, \emptyset, \emptyset, 2z, 3z)$. Przy znakowaniu M_0 aktywne są przejścia $t1$ i $t3$. Ponieważ wszystkie przejścia tej sieci mogą być wykonywane tylko przy trywialnych (pustych) wiązaniach, więc łuki grafu osiągalności będą etykietowane wyłącznie nazwami przejść. W wyniku wykonania przejść $t1$ i $t3$ otrzymujemy odpowiednio znakowania $M_1 = (\emptyset, a, 2b, \emptyset, \emptyset, 2z, z)$ i $M_2 = (a, \emptyset, b, b, \emptyset, z, 3z)$, które stanowią kolejne węzły grafu osiągalności. Następnie rozważamy przejścia aktywne przy znakowaniach M_1 i M_2 itd. Pełny graf osiągalności dla tej sieci przedstawiono na rys. 11.1.

Jeżeli wszystkie miejsca mają skończone górne ograniczenia wielozbiorowe, to graf osiągalności jest skończony. Podobnie jak dla sieci niskiego poziomu, analizując węzły grafu osiągalności, można sprawdzić ograniczonosć poszczególnych miejsc.

Twierdzenie 11.1. Niech $\mathcal{G} = (V, A, \gamma)$ będzie grafem osiągalności sieci \mathcal{N} .

- (1) Miejsce p_i sieci \mathcal{N} jest znakowane wtedy i tylko wtedy, gdy w grafie \mathcal{G} istnieje węzeł, dla którego i -ta współrzędna jest różna od zbioru pustego.
- (2) Miejsce p_i sieci \mathcal{N} jest k -ograniczone ($k \in \mathbb{N}$) wtedy i tylko wtedy, gdy dla każdego węzła $M \in V$, $|M(p_i)| \leq k$. Najmniejszym górnym ograniczeniem liczbowym jest liczba $k = \max_{M \in V} |M(p_i)|$.
- (3) Wielozbior $X \in 2^{C(p_i)^*}$ jest górnym ograniczeniem wielozbiorowym miejsca p_i wtedy i tylko wtedy, gdy dla każdego węzła $M \in V$, $M(p_i) \leq X$. ■

Dla sieci kolorowanych zachowuje ważność twierdzenie 5.2. Można również sformułować twierdzenie analogiczne do twierdzenia 5.3.



Rys. 11.1. Graf osiągalności dla modelu współzawodnictwa procesów

Twierdzenie 11.2. Niech $\mathcal{G} = (V, A, \gamma)$ będzie grafem osiągalności sieci \mathcal{N} .

- (1) Przejście t sieci \mathcal{N} jest martwe ($\mathcal{L}0$ -żywe) przy wiązaniu $b \in \mathcal{B}(t)$ wtedy i tylko wtedy, gdy w grafie \mathcal{G} nie występuje łuk z etykietą (t, b) .
- (2) Przejście t sieci \mathcal{N} jest $\mathcal{L}1$ -żywe przy wiązaniu $b \in \mathcal{B}(t)$ wtedy i tylko wtedy, gdy w grafie \mathcal{G} występuje łuk z etykietą (t, b) .
- (3) Przejście t sieci \mathcal{N} jest $\mathcal{L}2$ -żywe przy wiązaniu $b \in \mathcal{B}(t)$ wtedy i tylko wtedy, gdy dla dowolnej liczby $k \in \mathbb{N}$ istnieje droga w grafie \mathcal{G} , w której co najmniej k razy występuje łuk z etykietą (t, b) .

- (4) Przejście t sieci \mathcal{N} jest $\mathcal{L}3$ -żywe przy wiązaniu $b \in \mathcal{B}(t)$ wtedy i tylko wtedy, gdy istnieje droga w grafie \mathcal{G} , w której łuk z etykietą (t, b) występuje nieskończenie wiele razy.
- (5) Jeżeli w grafie \mathcal{G} istnieje cykl, w którym jeden z łuków ma etykietę (t, b) , to przejście t jest $\mathcal{L}3$ -żywe przy wiązaniu b .
- (6) Jeżeli graf \mathcal{G} jest grafem silnie spójnym i występuje w nim łuk z etykietą (t, b) , to przejście t jest żywe ($\mathcal{L}4$ -żywe) przy wiązaniu b .
- (7) Sieć \mathcal{N} jest silnie sprawiedliwa wtedy i tylko wtedy, gdy każda nieskończona droga w grafie \mathcal{G} zawiera nieskończenie wiele wystąpień łuku z etykietą (t, b) dla dowolnego $(t, b) \in \mathcal{B}$.
- (8) Sieć \mathcal{N} jest sprawiedliwa wtedy i tylko wtedy, gdy dla dowolnego przejścia $t \in T$ w każdej nieskończonej drodze w grafie \mathcal{G} występują łącznie nieskończenie wiele razy pary (t, b_i) , przy czym $b_i \in \mathcal{B}(t)$. ■

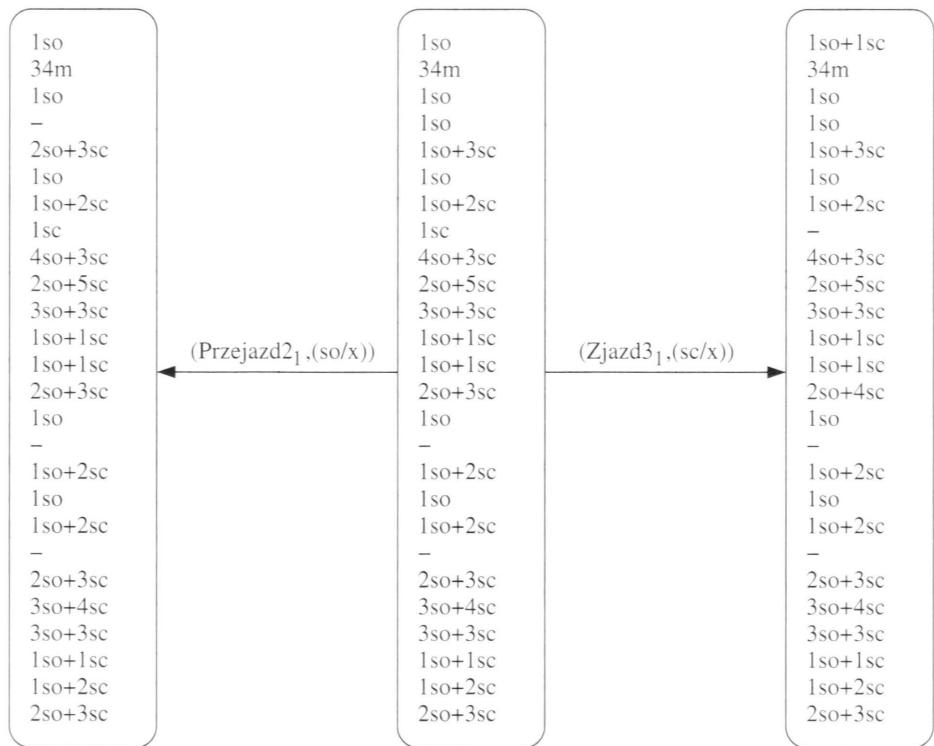
Z powyższych twierdzeń wynika, że sieć przedstawiona na rys. 10.5 jest silnie żywa, odwracalna, ale nie jest sprawiedliwa. Ograniczenia miejsc podane dla tej sieci w p. 10.6.1 znajdują również potwierdzenie przy analizie węzłów grafu osiągalności.

Jeżeli zbiór znakowań sieci kolorowanej osiągalnych ze znakowania początkowego jest zbiorem nieskończonym, to graf osiągalności jest również grafem nieskończonym. Taka sytuacja zachodzi dla modelu autostrady (zob. p. 10.6.2). Przyjmijmy, że zbiór miejsc sieci uporządkowano¹ w sposób następujący: {*Otoczenie*, *Kasa*, *Odcinek*₁, *Odcinek*₂, ..., *Odcinek*₆, *Licznik*₁, *Licznik*₂, ..., *Licznik*₆, *Odcinek*₁₂, *Odcinek*₂, ..., *Odcinek*₆, *Licznik*₇, *Licznik*₈, ..., *Licznik*₁₂}. Fragment grafu osiągalności dla rozważanego modelu przedstawiono na rys. 11.2. Węzeł grafu znajdujący się w środku rysunku ma jeszcze 21 innych następców.

Na podstawie grafu osiągalności sieci kolorowanej można zbudować graf silnie spójnych składowych, tak jak w podrozdz. 5.3. Dla sieci kolorowanych można sformułować twierdzenie analogiczne do twierdzenia 5.9.

Twierdzenie 11.3. Niech \mathcal{N} będzie siecią kolorowaną, a $\mathcal{G} = (V, A, \gamma)$ grafem silnie spójnych składowych grafu osiągalności sieci \mathcal{N} .

¹ Dla uproszczenia w przypadku fuzji miejsc uwzględniamy tylko jedno z miejsc, a w przypadku połączenia gniazdo-port, tylko miejsce gniazdowe. Indeksy przy nazwach miejsc służą do odróżnienia poszczególnych instancji miejsc. Miejsca zwane *Licznik* są numerowane w sposób następujący: najpierw liczniki wjazdów dla pierwszej instancji strony *Nitka* (indeksy 1–3), później liczniki zjazdów (indeksy 4–6). Indeksy od 7 do 12 odnoszą się do liczników wjazdów i zjazdów dla drugiej instancji strony *Nitka*.



Rys. 11.2. Fragment grafu osiągalności dla modelu autostrady

- (1) Niech $M_1, M_2 \in \mathcal{R}(M_0)$. Znakowanie $M_2 \in \mathcal{R}(M_1)$ wtedy i tylko wtedy, gdy w grafie \mathcal{G} istnieje droga od $\mathcal{G}_S(M_1)$ do $\mathcal{G}_S(M_2)$ (droga ta może mieć długość zero).
- (2) Istnieje znakowanie własne sieci \mathcal{N} wtedy i tylko wtedy, gdy graf \mathcal{G} zawiera dokładnie jeden węzeł końcowy. Znakowaniami własnymi sieci są znakowania ze zbioru reprezentowanego przez ten węzeł.
- (3) Sieć \mathcal{N} jest odwracalna wtedy i tylko wtedy, gdy graf \mathcal{G} zawiera dokładnie jeden węzeł.
- (4) Znakowanie $M \in \mathcal{R}(M_0)$ jest znakowaniem martwym sieci \mathcal{N} wtedy i tylko wtedy, gdy węzeł $\mathcal{G}_S(M)$ jest trywialnym węzłem końcowym grafu \mathcal{G} .
- (5) Przejście t sieci \mathcal{N} jest żywe przy wiązaniu $b \in \mathcal{B}(t)$ wtedy i tylko wtedy, gdy każdy węzeł końcowy grafu \mathcal{G} zawiera łuk z etykietą (t, b) .
- (6) Przejście t sieci \mathcal{N} jest żywe wtedy i tylko wtedy, gdy każdy węzeł końcowy grafu \mathcal{G} zawiera łuk z etykietą (t, b) dla pewnego $b \in \mathcal{B}(t)$.

- (7) Przejście t sieci \mathcal{N} jest silnie żywe wtedy i tylko wtedy, gdy dla dowolnego $b \in \mathcal{B}(t)$ każdy węzeł końcowy grafu \mathcal{G} zawiera łuk z etykietą (t, b) . ■

Dla modelu współzawodnictwa procesów (zob. rys. 10.5) graf silnie spójnych składowych zawiera tylko jeden węzeł. W przypadku modelu autostrady (zob. p. 10.6.2) graf silnie spójnych składowych jest nieskończony, podobnie jak graf osiągalności.

11.2. Grafy osiągalności z klasami równoważności

Grafów osiągalności z klasami równoważności (czyli *relacyjnych grafów osiągalności*) używa się w celu zredukowania rozmiaru grafu przedstawiającego zbiór osiągalnych znakowań sieci. Jak grafy pokrycia dla sieci niskiego poziomu, tak i relacyjny graf osiągalności nie przedstawia w sposób jawny wszystkich osiągalnych znakowań sieci. Aby otrzymać graf relacyjny, należy zdefiniować dwie relacje równoważności: jedną określoną na zbiorze osiągalnych znakowań $\mathcal{R}(M_0)$, a drugą na zbiorze \mathcal{B} . Węzły relacyjnego grafu osiągalności reprezentują klasy abstrakcji pierwszej relacji, łuki zaś, klasy abstrakcji drugiej z nich.

Relacyjne grafy osiągalności są zazwyczaj zdecydowanie mniejsze, niż odpowiadające im pełne grafy osiągalności (np. dla nieskończonego grafu osiągalności można uzyskać skończony graf relacyjny). Taki zredukowany graf może nadal posłużyć do analizy wielu właściwości dynamicznych sieci. Wyniki analizy są jednak uwarunkowane właściwym zdefiniowaniem relacji równoważności.

Niech dana będzie sieć kolorowana $\mathcal{N} = (\Sigma, P, T, A, \gamma, C, G, E, M_0)$.

Definicja 11.2. Specyfikacją równoważności sieci \mathcal{N} nazywamy parę $(\sim_{\mathcal{R}}, \sim_{\mathcal{B}})$, przy czym $\sim_{\mathcal{R}}$ jest relacją równoważności określoną na zbiorze $\mathcal{R}(M_0)$, $\sim_{\mathcal{B}}$ zaś jest relacją równoważności określoną na zbiorze \mathcal{B} . ■

Zbiory wszystkich klas abstrakcji relacji $\sim_{\mathcal{R}}$ oraz $\sim_{\mathcal{B}}$ oznaczamy odpowiednio symbolami $\mathcal{R}(M_0)/_{\sim_{\mathcal{R}}}$ oraz $\mathcal{B}/_{\sim_{\mathcal{B}}}$.

Definicja 11.3. Specyfikację równoważności $(\sim_{\mathcal{R}}, \sim_{\mathcal{B}})$ nazywamy *zgodną*, jeżeli dla dowolnych znakowań M_1, M'_1 i M_2 sieci \mathcal{N} i dowolnej pary (t_1, b_1) , takich że:

$$M_1 \xrightarrow{(t_1, b_1)} M'_1 \wedge M_1 \sim_{\mathcal{R}} M_2, \quad (11.1)$$

istnieje znakowanie M'_2 i para (t_2, b_2) , takie że:

$$M_2 \xrightarrow{(t_2, b_2)} M'_2 \wedge (t_1, b_1) \sim_{\mathcal{B}} (t_2, b_2) \wedge M'_1 \sim_{\mathcal{R}} M'_2. \quad (11.2)$$

Zgodność relacji równoważności gwarantuje, że wychodząc od równoważnych znakowań, można obserwować podobne zachowanie sieci. W definicji 11.3 wystarczy ograniczyć się do rozważania znakowań, które są równoważne² dowolnemu znakowaniu ze zbioru $\mathcal{R}(M_0)$

Definicja 11.4. Niech $(\sim_{\mathcal{R}}, \sim_{\mathcal{B}})$ będzie zgodną specyfikacją równoważności sieci \mathcal{N} . *Grafem osiągalności z klasami równoważności (relacyjnym grafem osiągalności) sieci \mathcal{N}* nazywamy graf $\mathcal{G} = (V, A, \gamma)$ etykietowany nad zbiorem $\mathcal{B}/_{\sim_{\mathcal{B}}}$, spełniający warunki:

- (1) $V = \{[M] \in \mathcal{R}(M_0)/_{\sim_{\mathcal{R}}} : [M] \cap \mathcal{R}(M_0) \neq \emptyset\}$ jest zbiorem węzłów grafu.
- (2) A jest zbiorem łuków postaci $([M_1], [(t, b)], [M_2])$, przy czym $[M_1], [M_2] \in V$, $[(t, b)] \in \mathcal{B}/_{\sim_{\mathcal{B}}}$ oraz istnieją takie $M'_1 \in [M_1]$, $M'_2 \in [M_2]$, $(t', b') \in [(t, b)]$, że $M'_1 \xrightarrow{(t', b')} M'_2$.
- (3) $\forall a = ([M_1], [(t, b)], [M_2]) \in A : \gamma(a) = ([M_1], [M_2])$. ■

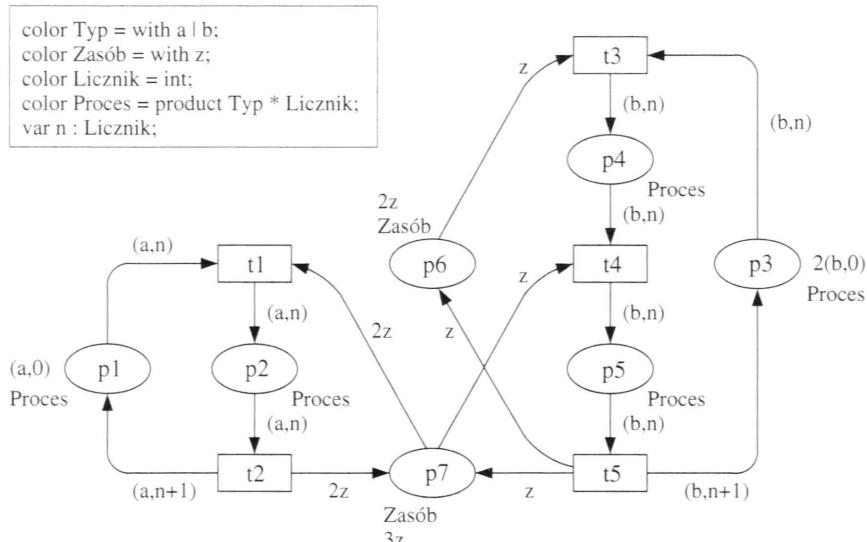
W relacyjnym grafie osiągalności klasy abstrakcji obu relacji równoważności są reprezentowane przez swoich przedstawicieli.

Rozważmy przykład sieci \mathcal{N} przedstawionej na rys. 11.3. Sieć \mathcal{N} jest modyfikacją sieci pokazanej na rys. 10.5. W porównaniu z pierwszą wersją modelu współzawodnictwa procesów, zmieniono definicję typu *Proces*, tak by umożliwić procesom zliczanie cykli ich pracy. Wskutek tego nowa sieć ma nieskończony zbiór osiągalnych znakowań, mimo że charakteryzuje się podobnym zachowaniem jak sieć na rys. 10.5. Uniemożliwia to zbudowanie pełnego grafu osiągalności. Dla rozważanej sieci można podać specyfikację równoważności, przy której graf relacyjny będzie izomorficzny z grafem osiągalności przedstawionym na rys. 11.1. Powiemy, że dwa znakowania sieci są równoważne, jeżeli będą identyczne po pominięciu liczników procesów. Dwie pary $(t_i, b_i), (t_j, b_j)$ uznamy za równoważne, jeżeli $t_i = t_j$ (pominiecie wartości zmiennej n). Łatwo sprawdzić, że opisana specyfikacja równoważności jest zgodna. Graf relacyjny sporządzony dla sieci na rys. 11.3 przedstawiono na rys. 11.4.

Grafy relacyjne nie zawierają informacji o pojedynczych znakowaniach i parach przejście-wiązanie, lecz informacje o klasach abstrakcji dwóch relacji równoważności. Jeżeli w grafie istnieje węzeł M i łuk o etykiecie (t, b) wychodzący z węzła M , to istnieje w klasie $[(t, b)]$ taka para (t', b') , że przejście t' jest aktywne przy

² W pracy [32] rozważane są również specyfikacje równoważności, dla których przyjmuje się słabszą definicję ich zgodności: nie żąda się, by dla znakowań M_1 i M_2 równoważne były znakowania bezpośrednio z nich osiągalne, ale dopiero najbliższe znakowania, które należą do innych klas równoważności niż M_1 i M_2 .

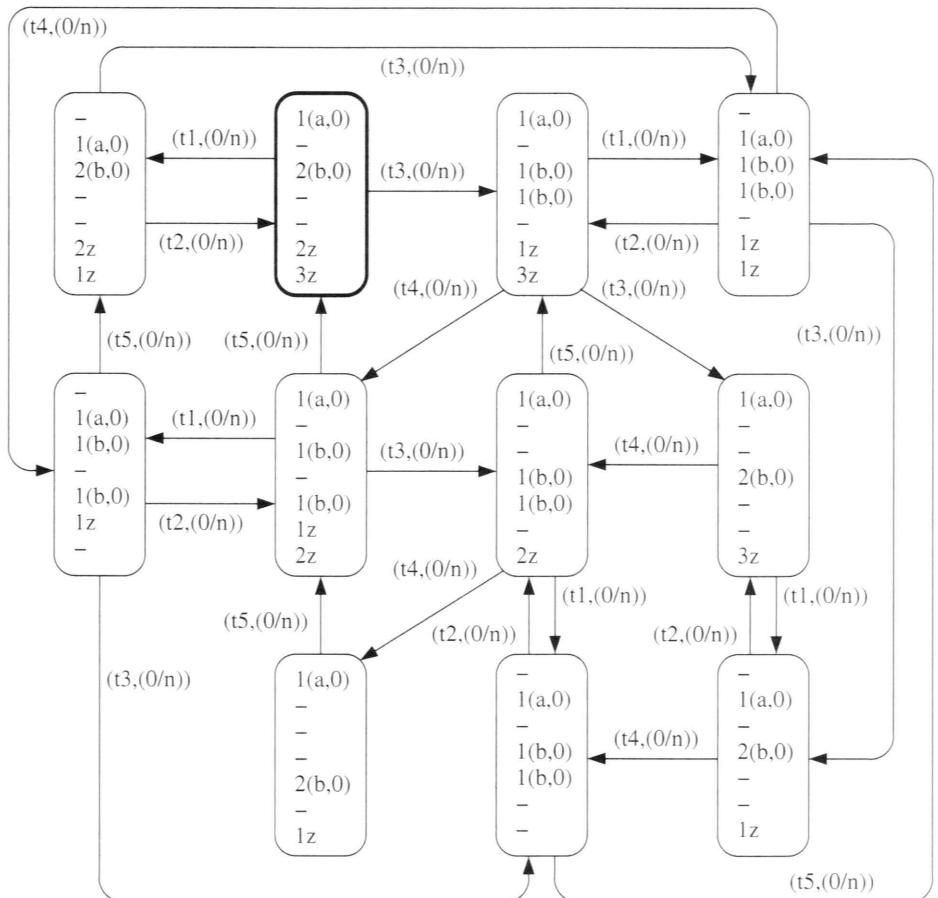
wiązaniu b' , przy pewnym znakowaniu należącym do klasy $[M]$. Dla dowolnego ciągu przejść α istnieje w grafie relacyjnym odpowiadająca mu droga, której łuki etykietowane są klasami abstrakcji (a właściwie, przedstawicielami klas abstrakcji) elementów ciągu α . Na odwrót, każda droga w grafie relacyjnym odpowiada co najmniej jednemu ciągowi przejść.



Rys. 11.3. Model współzawodnictwa procesów (wersja 2)

Możliwość dokonania analizy własności sieci kolorowanej zależy od specyfikacji relacji równoważności. Graf relacyjny przedstawiony na rys. 11.4 pozwala wysnuć podobne wnioski jak graf osiągalności na rys. 11.1. Relację \sim_R tak zdefiniowano, że dwa równoważne znakowania mają tyle samo znaczników w każdym miejscu sieci, można więc wnioskować o liczbowej ograniczości miejsc (analogicznie do twierdzenia 11.1 (1)-(2)). Dolnym ograniczeniem liczbowym dla wszystkich miejsc jest zero, a wielozbiorowym – zbiór pusty. Górnego liczbowego ograniczenie dla miejsc p_1 i p_2 wynosi jeden, dla miejsc od p_3 do p_6 – dwa, a dla miejsca p_7 – trzy. Można ponadto pokazać, że ograniczenia wielozbiorowe miejsc p_6 i p_7 wynoszą odpowiednio $2z$ i $3z$, a pozostałe miejsca nie mają skończonego górnego ograniczenia wielozbiorowego.

Podobnie, użyta w tym przykładzie relacja \sim_B pozwala wyciągnąć wnioski dotyczące żywotności przejść, ale nie żywotności przy ustalonym wiązaniu: sieć jest żywa, ale nie jest silnie żywa. Wszystkie przejścia są potencjalnie wykonalne przy dowolnych swoich wiązaniach. Sieć nie jest sprawiedliwa.



Rys. 11.4. Graf relacyjny dla modelu współzawodnictwa procesów (wersja 2)

Jak dla sieci uogólnionych, tak i dla sieci kolorowanej graf osiągalności można redukować, stosując grafy silnie spójnych składowych, symetrie sieci i trwałe zbiorowe przejścia. Symetrie sieci kolorowanych mogą być traktowane jako szczególny przypadek grafów relacyjnych. Podgrupa symetrii wyznacza zgodną specyfikację równoważności, zatem graf osiągalności zredukowany ze względu na symetrię jest w rzeczywistości przykładem grafu relacyjnego (zob. prace: [32] i [36]). Dla trwałych zbiorów przejścia sama idea ich stosowania pozostaje taka, jak dla sieci uogólnionych (zob. prace: [39], [40], [84] i [85]).

11.3. Niezmienniki sieci kolorowanych

W sieciach uogólnionych niezmienniki miejsc miały postać wektorów o współrzędnych całkowitych. Nieujemne niezmienniki miejsc wyznaczały pewną zachowawczą podsieć, tzn. podsieć, dla której ważona suma znaczników była stała, niezależnie od osiąganego znakowania sieci. Miejsca sieci kolorowanej mogą mieć przypisane różne typy znaczników, dlatego bezpośrednie dodawanie znakowań dwóch lub większej liczby miejsc w ogólnym przypadku nie jest możliwe. Niezmienniki miejsc sieci kolorowanej są zapisywane w postaci równań, które wyrażają stałe (tzn. niezależne od osiąganego znakowania) zależności między znakowaniami poszczególnych miejsc.

Rozważmy sieć przedstawioną na rys. 10.5. Przykładami niezmienników miejsc dla tej sieci³ są następujące równania:

$$\begin{aligned} M(p1) + M(p2) &= a, \\ M(p3) + M(p4) + M(p5) &= 2b. \end{aligned} \tag{11.3}$$

Nie są to jedyne niezmienniki miejsc, jakie można podać dla rozważanej sieci. W sposób bezpośredni nie można jednak wyrazić wszystkich zależności, na przykład łączna liczba znaczników w miejscach $p4$, $p5$ i $p6$ jest równa dwa, ale nie można tego przedstawić za pomocą sumy $M(p4) + M(p5) + M(p6)$, gdyż różne są typy miejsca $p6$ i miejsc $p4$ i $p5$. Zapisanie tej zależności jest możliwe dopiero po zdefiniowaniu odpowiednich funkcji, na których podstawie znakowania tych miejsc można przedstawić w postaci wielozbiorów nad pewnym ustalonym wspólnym zbiorem X .

Pojęcie wielozbioru jako rozszerzenie klasycznego pojęcia zbioru było wprowadzone w podrozdz. 10.1. Poniżej, na potrzeby definicji niezmienników miejsc sieci kolorowanych skorzystamy z definicji *wielozbioru uogólnionego*. Jedyna różnica między wielozbiorami a wielozbiorami uogólnionymi dotyczy współczynników elementów, które w przypadku wielozbiorów uogólnionych mogą być ujemne.

Definicja 11.5. *Wielozbiorem uogólnionym A^* nad niepustym zbiorem A nazywamy dowolną funkcję*

$$A^*: A \rightarrow \mathbb{Z}. \tag{11.4}$$

³ Niezmienniki sieci uogólnionych można również zapisywać w postaci równań, na przykład niezmiennik $I_1 = (1, 1, 1, 0, 0)$ (sieć na rys. 5.21) można zapisać jako: $M(p1) + M(p2) + M(p3) = 3$.

Niech A będzie dowolnym zbiorem. Symbolem 2^{A^*} będziemy oznaczać rodzinę wszystkich wielozbiorów uogólnionych, zbudowanych nad zbiorem A . Jak wielozbior, tak i wielozbior uogólnione będziemy zapisywać algebraicznie jako odpowiednie sumy (zob. podrozdz. 10.1). Operacje na wielozbiorach uogólnionych definiujemy tak jak dla wielozbiorów, przy czym odejmowanie wielozbiorów uogólnionych jest zawsze wykonalne.

Definicja 11.6. Niech dane będą zbiory A i B . Funkcję $f: 2^{A^*} \rightarrow 2^{B^*}$ nazywamy *liniową*, jeżeli dla dowolnych $A_1^*, A_2^* \in 2^{A^*}$ spełniony jest warunek:

$$f(A_1^* + A_2^*) = f(A_1^*) + f(A_2^*). \quad (11.5)$$

■

Jeżeli funkcja $f: 2^{A^*} \rightarrow 2^{B^*}$ jest funkcją liniową, to można pokazać, że dla dowolnego $n \in \mathbb{Z}$ i $A^* \in 2^{A^*}$ spełnia ona następujące zależności:

$$\begin{aligned} f(n \cdot A^*) &= n \cdot f(A^*), \\ f(-A^*) &= -f(A^*), \\ f(\emptyset) &= \emptyset. \end{aligned} \quad (11.6)$$

Definiowanie funkcji liniowej określonej na rodzinie wszystkich wielozbiorów uogólnionych zbudowanych nad zbiorem A można sprowadzić do określenia funkcji $g: A \rightarrow 2^{B^*}$. Funkcję $f: 2^{A^*} \rightarrow 2^{B^*}$ definiujemy wówczas następująco:

$$f(A^*) = \sum_{x \in A} A^*(x)g(x), \quad (11.7)$$

przy czym $A^* = \sum_{x \in A} A^*(x)x$.

Definicja 11.7. Niech dane będą zbiory A i B oraz funkcja liniowa $f: 2^{A^*} \rightarrow 2^{B^*}$. Funkcję f nazywamy *nieujemną*, jeżeli spełniony jest warunek:

$$\forall x \in A: f(x) \in 2^{B^*}. \quad (11.8)$$

■

Nieujemna funkcja liniowa f przekształca wielozbior na wielozbior.

Niech $\mathcal{N} = (\Sigma, P, T, A, \gamma, C, G, E, M_0)$ będzie siecią kolorowaną i $X \in \Sigma$ będzie ustalonym typem (kolorem) sieci \mathcal{N} . Stosując zdefiniowane powyżej pojęcia, można dla dowolnego typu $Y \in \Sigma$ zdefiniować funkcję $g: Y \rightarrow 2^{X^*}$. Funkcja ta wyznacza

jednoznacznie funkcję liniową $f: 2^{Y^*} \rightarrow 2^{X^*}$, określoną zgodnie z wzorem (11.7). W ten sposób można określić zbiór funkcji, przy których użyciu dowolne znakowanie każdego miejsca sieci \mathcal{N} można przedstawić jako wielozbiór uogólniony nad zbiorem X . W konsekwencji umożliwia to formułowanie niezmienników miejsc sieci z wykorzystaniem dowolnych jej miejsc.

Definicja 11.8. Niech $X \in \Sigma$ będzie ustalonym typem sieci kolorowanej \mathcal{N} . Zbiorem wag miejsc sieci \mathcal{N} o wartościach w zbiorze X nazywamy zbiór funkcji:

$$F_P = \left\{ f_p: 2^{C(p)^*} \rightarrow 2^{X^*} : p \in P \right\}. \quad (11.9)$$

Zbiór wag miejsc F_P nazywamy *zachowawczym*, jeżeli:

$$\forall (t, b) \in \mathcal{B}: \sum_{p \in P} f_p(E(p, t)_b) = \sum_{p \in P} f_p(E(t, p)_b). \quad (11.10)$$

Zbiór wag miejsc F_P wyznacza *niezmiennik miejsc* sieci \mathcal{N} , jeżeli:

$$\forall M \in \mathcal{R}(M_0): \sum_{p \in P} f_p(M(p)) = \sum_{p \in P} f_p(M_0(p)). \quad (11.11)$$

■

Twierdzenie 11.4. Zbiór wag miejsc F_P jest zachowawczy wtedy i tylko wtedy, gdy F_P wyznacza niezmiennik miejsc sieci \mathcal{N} . ■

Dowód. Zachowawcość zbioru wag miejsc F_P oznacza, że ważona suma znaczników usuwanych z miejsc sieci przez dowolne przejście $t \in T$ (przy dowolnym wiążaniu $b \in \mathcal{B}(t)$) jest taka sama jak ważona suma znaczników dodawanych do miejsc sieci przez to przejście. Wynika stąd, że ważona suma znaczników w miejscach sieci pozostaje niezmieniona. Z drugiej strony, jeżeli nie ulega zmianie ważona suma znaczników w miejscach sieci, to zbiór wag F_P musi być zachowawczy. Formalny zapis tego dowodu można znaleźć w pracy [32]. □

Rozważmy ponownie sieć pokazaną na rys. 10.5. W celu uproszczenia zapisu przyjmijmy oznaczenia $A_1 = \text{Proces}$ i $A_2 = \text{Zasób}$. Dla zbiorów A_1 i A_2 zdefiniujemy funkcje:

$$\begin{aligned} g_1: A_1 &\rightarrow 2^{A_2^*}, \quad g_1(a) = g_1(b) = z, \\ g_2: A_2 &\rightarrow 2^{A_2^*}, \quad g_2(z) = z, \\ g_3: A_1 &\rightarrow 2^{A_2^*}, \quad g_3(a) = g_3(b) = \emptyset, \\ g_4: A_2 &\rightarrow 2^{A_2^*}, \quad g_4(z) = \emptyset. \end{aligned} \quad (11.12)$$

Funkcje g_1, \dots, g_4 wyznaczają jednoznacznie funkcje f_1, \dots, f_4 określone zgodnie z wzorem (11.7). Niech zbiór wag miejsc F_P o wartościach w zbiorze A_2 będzie określony tak jak poniżej:

$$\begin{aligned} F_P &= \{f_{p1}, f_{p2}, \dots, f_{p7}\}, \text{ przy czym:} \\ f_{p1} &= f_{p2} = f_{p3} = f_3, \\ f_{p4} &= f_{p5} = f_1, \\ f_{p6} &= f_2, \\ f_{p7} &= f_4. \end{aligned} \tag{11.13}$$

Zbiór wag miejsc F_P jest zachowawczy i wyznacza niezmiennik miejsc:

$$f_1(M(p4)) + f_1(M(p5)) + f_2(M(p6)) = 2z, \tag{11.14}$$

Funkcja f_2 jest tożsamościowa, więc niezmiennik (11.14) można zapisać jako:

$$f_1(M(p4)) + f_1(M(p5)) + M(p6) = 2z, \tag{11.15}$$

Twierdzenie 11.5. Niech p' będzie miejscem sieci \mathcal{N} i niech F_P będzie zachowawczym zbiorem wag miejsc, takim że wszystkie funkcje należące do zbioru F_P są nieujemne oraz $f_{p'} = f_{id}$. Suma

$$\sum_{p \in P} f_p(M_0(p)) \tag{11.16}$$

jest górnym wielozbiorowym ograniczeniem miejsca p' . Jeżeli wielozbior (11.16) jest skończony, to jego rozmiar jest górnym liczbowym ograniczeniem miejsca p' . ■

Dowód. Zachowawczy przepływ miejsc F_P wyznacza niezmiennik miejsc sieci \mathcal{N} , dla którego równanie (11.11) można zapisać w postaci:

$$M(p') + \sum_{p \in P - \{p'\}} f_p(M(p)) = \sum_{p \in P} f_p(M_0(p)). \tag{11.17}$$

Ponieważ wszystkie funkcje należące do zbioru F_P są nieujemne, suma występująca po lewej stronie równania (11.17) jest wielozborem. A zatem

$$M(p') \leq \sum_{p \in P} f_p(M_0(p)). \tag{11.18}$$

□

Ograniczenia miejsca p' , które możemy otrzymać w wyniku zastosowania twierdzenia 11.5, nie są optymalne, ale pozwalają stwierdzić, że dane miejsce jest ograniczone. Analiza innych własności sieci prowadzona z użyciem niezmienników miejsc zależy od postaci tych niezmienników. Często ma ona charakter matematycznego dowodu, w którym niezmienniki miejsc odgrywają zasadniczą rolę.

Niezmienniki sieci kolorowanych można wyznaczać metodami analogicznymi do metod dla sieci uogólnionych. Sieć kolorowaną można zapisać w postaci macierzy incydencji, przy czym współczynniki takiej macierzy nie są liczbami, lecz funkcjami. Wyznaczanie niezmienników sprowadza się zatem do przekształcania⁴ takiej macierzy; jest to na ogół zadanie złożone obliczeniowo ze względu na elementy tej macierzy. Do wyznaczania niezmienników sieci kolorowanych używa się najczęściej odpowiedniego oprogramowania, które automatyzuje ten proces. W praktyce często potrzebne niezmienniki miejsc można podać bez uciekania się do macierzy incydencji, tak jak dla sieci przedstawionej na rys. 10.5.

Niezmienniki miejsc dla kolorowanych sieci hierarchicznych definiuje się analogicznie. Należy wziąć pod uwagę fakt, że przy określaniu zbioru wag miejsca należące do fuzji lub tworzące parę (*gniazdo, port*) są traktowane jak pojedyncze miejsce (zob. pracę [32]). Ponadto niezmienniki miejsc sieci hierarchicznej można wyznaczać na podstawie niezmienników podanych dla pojedynczych stron tej sieci lub fragmentów hierarchicznego modelu. Pozwala to na uzyskanie niezmienników miejsc dla bardzo dużych modeli bez konieczności uwzględniania całości modelu przy obliczeniach. Oprócz tego daje to możliwość wyznaczenia niezmienników sieci, dla których liczba instancji stron nie musi być dokładnie określona.

Rozważmy przykład modelu autostrady, przedstawiony w p. 10.6.2. Jeżeli sieć zawierałaby tylko jednąinstancję strony *Nitka*, to przykładem niezmiennika miejsc dla tej strony i jej bezpośrednich podstron byłoby równanie:

$$M(Otoczenie) + \sum_{i=1}^6 M(Odcinek i_1) = 10so + 10sc. \quad (11.19)$$

Jeżeli weźmiemy pod uwagę fakt, że wszystkie instancje strony *Nitka* współdzielą miejsce *Otoczenie* (fuzja globalna), to dla rozważanego wcześniej modelu z dwiema instancjami tej strony otrzymujemy niezmiennik miejsc:

⁴ Szczegółowy opis zasad przekształcania macierzy incydencji dla sieci kolorowanych można znaleźć w pracy [32].

$$\begin{aligned}
 M(Otoczenie) + \sum_{i=1}^6 M(Odcinek i_1) \\
 + \sum_{i=1}^6 M(Odcinek i_2) = 10so + 10sc.
 \end{aligned} \tag{11.20}$$

Na podstawie przedstawionego niezmiennika można wnioskować, że wszystkie występujące w nim miejsca są ograniczone (wszystkie funkcje f_p są tożsamościami). Górnym ograniczeniem wielozbiorowym tych miejsc jest wielozbiór $10so + 10sc$, liczbowy zaś 20.

Podobnie jak w sieciach uogólnionych, niezmienniki przejść są stosowane do wyznaczania ciągów przejść, które prowadzą ponownie do tego samego znakowania. *Niezmiennik przejścia* sieci kolorowanej jest zatem wielozbiorem $I \in 2^{\mathcal{B}^*}$, którego wykonanie nie zmienia znakowania sieci⁵.

Definicja 11.9. Skończony wielozbiór $I \in \mathcal{B}$ nazywamy *niezmiennikiem przejścia* sieci \mathcal{N} jeżeli

$$\forall p \in P: \sum_{(t,b) \in I} E(p,t)_b = \sum_{(t,b) \in I} E(t,p)_b. \tag{11.21}$$

Rozważmy sieć przedstawioną na rys. 10.5. Przykładami niezmienników przejść dla tej sieci są następujące wielozbiory⁶:

$$\begin{aligned}
 I_1 &= t1 + t2, \\
 I_2 &= t3 + t4 + t5.
 \end{aligned} \tag{11.22}$$

Niezmienniki te odpowiadają cyklom wykonywanym przez poszczególne procesy. W celu sprawdzenia, że określony wielozbiór nad zbiorem \mathcal{B} jest niezmiennikiem przejść, nie trzeba stosować funkcji wag. Wartościowania wyrażeń luków otaczających dowolne miejsce są wielozbiorami nad tym samym zbiorem.

Definicja 11.10. Niech I będzie niezmiennikiem przejść sieci \mathcal{N} . Niezmiennik I nazywamy *realizowanym*, jeżeli istnieje znakowanie $M \in \mathcal{R}(M_0)$ i taki ciąg przejść $\alpha \in \mathcal{L}(M)$, że każdy element z wielozbioru I występuje w tym ciągu dokładnie tyle razy, ile wynosi wartość jego współczynnika w wielozbiorze I . ■

⁵ Niezmienniki przejść sieci uogólnionych również mogą być przedstawiane jako wielozbiory, ale przejść, na przykład niezmiennik $I = (2, 1, 1, 1)$ dla sieci na rys. 5.21 można zapisać jako wielozbiór $I = 2t1 + t2 + t3 + t4$.

⁶ Ponieważ w tym przypadku wszystkie wiązania są funkcjami, których dziedziny są zbiorami pustymi, więc pominięto je przy zapisie niezmienników.

Oba podane niezmienniki przejść są realizowalne. Podobnie jak niezmienniki miejsc, mogą one być wyznaczane na podstawie macierzy incydencji sieci kolorowanej, przy czym ze względu na złożoność obliczeń stosuje się w tym celu raczej odpowiednie oprogramowanie (zob. pracę [32]). Niezmienników przejść można używać do analizy żywotności sieci lub własności związanych z jej sprawiedliwością. Istnienie realizowanego niezmiennika przejść oznacza istnienie cyklu w grafie osiągalności sieci.

Jeżeli wszystkie łuki sieci kolorowanej $\mathcal{N} = (\Sigma, P, T, A, \gamma, C, G, E, M_0)$ mają przypisane jednolite wyrażenia, to do analizy sieci kolorowanej można zastosować sieci uogólnione (przy założeniu, że znakowania początkowe miejsc sieci \mathcal{N} są skończonymi wielozbiorami). Sieć uogólnioną $\mathcal{N}' = (P', T', A', W', M'_0)$, odpowiadającą sieci \mathcal{N} , budujemy w następujący sposób:

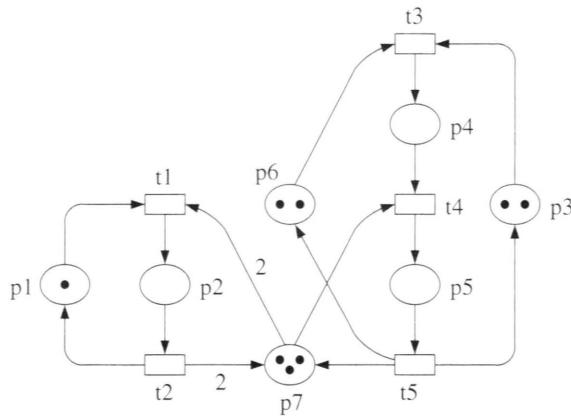
- (1) Dla sieci \mathcal{N}' przyjmujemy takie same zbiory miejsc i przejść, tzn. $P' = P$ i $T' = T$.
- (2) Jeżeli w sieci \mathcal{N} węzły p i t są połączone co najmniej jednym łukiem prowadzącym od p do t , to węzły te w sieci \mathcal{N}' są połączone łukiem (p, t) , przy czym wagę łuku ustalamy jako rozmiar wielozbioru znaczników, który jest usuwany z miejsca p w wyniku wykonania przejścia t . Analogicznie postępujemy, jeżeli węzły p i t są połączone co najmniej jednym łukiem prowadzącym od t do p .
- (3) Znakowanie początkowe M'_0 tak definiujemy, by liczba znaczników w miejscu p sieci \mathcal{N}' była równa rozmiarowi wielozbioru będącego znakowaniem początkowym w miejscu p sieci \mathcal{N} , tzn. $M'_0(p) = |M_0(p)|$.

Niezmienniki sieci \mathcal{N}' mogą być traktowane jako wskazówki do szukania niezmienników sieci kolorowanej \mathcal{N} , przy czym nie każdy niezmiennik sieci \mathcal{N}' oznacza istnienie odpowiadającego mu niezmiennika sieci \mathcal{N} . Na rysunku 11.5 przedstawiono sieć uogólnioną \mathcal{N}' odpowiadającą sieci kolorowanej na rys. 10.5.

Przykładami niezmienników miejsc sieci \mathcal{N}' są wektory $I_1 = (1, 1, 0, 0, 0, 0, 0)$, $I_2 = (0, 0, 1, 1, 1, 0, 0)$, $I_3 = (0, 0, 0, 1, 1, 1, 0)$ oraz $I_4 = (0, 2, 0, 0, 1, 0, 1)$. Niezmienniki I_1 i I_2 odpowiadają niezmiennikom miejsc sieci kolorowanej \mathcal{N} , przedstawionym we wzorze (11.3), a niezmiennik I_3 – niezmiennikowi opisanemu wzorem (11.15).

Rozważmy ponownie funkcje g_1, \dots, g_4 , których definicje zawiera wzór (11.12). Definicję funkcji g_1 zmodyfikujemy następująco:

$$g_1 : A_1 \rightarrow 2^{A_2^*}, \quad g_1(a) = 2z, \quad g_1(b) = z. \quad (11.23)$$



Rys. 11.5. Model współzawodnictwa procesów – sieć uogólniona

Funkcje g_1, \dots, g_4 wyznaczają jednoznacznie funkcje f_1, \dots, f_4 , określone zgodnie z wzorem (11.7). Niech zbiór wag miejsc F_P o wartościach w zbiorze A_2 będzie tak określony:

$$\begin{aligned} F_P &= \{f_{p1}, f_{p2}, \dots, f_{p7}\}, \text{ przy czym:} \\ f_{p1} &= f_{p3} = f_{p4} = f_3, \\ f_{p2} &= f_{p5} = f_1, \\ f_{p6} &= f_4, \\ f_{p7} &= f_2. \end{aligned} \tag{11.24}$$

Zbiór wag miejsc F_P jest zachowawczy i wyznacza niezmiennik miejsc

$$f_1(M(p2)) + f_1(M(p5)) + M(p7) = 3z, \tag{11.25}$$

który odpowiada niezmiennikowi I_4 .

Przykładami niezmienników przejść sieci \mathcal{N}' są wektory $I_1 = (1, 1, 0, 0, 0)$ oraz $I_2 = (0, 0, 1, 1, 1)$. Odpowiadają one niezmiennikom przejść sieci kolorowanej \mathcal{N} przedstawionym we wzorze (11.22).

11.4. Podsumowanie

W tym rozdziale omówiliśmy metody analizy kolorowanych sieci Petriego. Konieczność uwzględnienia różnych typów znaczników powoduje, że metod przedstawionych w rozdz. 5 nie można bezpośrednio stosować do CP-sieci. Do analizowania sieci kolorowanych używa się przede wszystkim grafów osiągalności i grafów relacyjnych. Grafy osiągalności konstruuje się tak jak dla sieci uogólnionych. Węzły grafu reprezentują znakowania sieci, a łuki prowadzące od jednego znakowania do drugiego mają przypisane etykiety zawierające informację o przejściu (i wiązaniu), którego wykonanie powoduje zmianę znakowania z pierwszego na drugie. Grafy relacyjne są używane do zredukowania rozmiaru grafu przedstawiającego zbiór osiągalnych znakowań. Węzły i łuki takiego grafu reprezentują klasy abstrakcji relacji równoważności określonych na zbiorach $\mathcal{R}(M_0)$ i \mathcal{B} .

Dla sieci kolorowanych można ponadto, podobnie jak dla sieci uogólnionych, wyznaczać niezmienniki miejsc i przejść. W przypadku CP-sieci zapisuje się je w postaci równań, które wyrażają pewne stałe zależności między znakowaniami miejsc sieci lub ciągami wykonywanych przejść. Ze względu na fakt, że macierz incydencji sieci kolorowanej nie jest macierzą liczbową, wyznaczanie takich niezmienników bez korzystania z odpowiedniego oprogramowania jest na ogół bardzo trudne.

Czasowe sieci kolorowane

Czasowe sieci kolorowane stanowią rozszerzenie sieci kolorowanych o model czasu. W przeciwieństwie do prostych i przedziałowych sieci czasowych, omawianych w rozdz. 8, w czasowych CP-sieciach ograniczenia czasowe połączono nie z przejściami, ale ze znacznikami sieci. W porównaniu z sieciami nieczasowymi znacznik sieci, oprócz wartości odpowiedniego typu, ma przypisaną również tzw. *pieczętkę czasową*, której wartość oznacza najwcześniejszy czas w systemie, kiedy dany znacznik staje się dostępny dla przejść sieci, tzn. może zostać pobrany przy wykonaniu przejścia. Aby móc odmierzać upływający czas, wprowadza się pojęcie *zegara globalnego*, który może przyjmować wartości dyskretne lub ciągłe. Wartość zegara reprezentuje rzeczywisty czas systemu i decyduje o tym, które znaczniki są *dostępne*, a które nie. Czas wykonania danego przejścia jest reprezentowany przez pieczętki czasowe znaczników dodawanych do jego miejsc wyjściowych. Jeżeli czas wykonania przejścia t wynosi 3 jednostki czasu, to czas ten można modelować, nadając pieczętkom czasowym wszystkich znaczników dodawanych do miejsc wyjściowych przejścia t wartości o 3 większe niż czas w systemie, w którym przejście to jest wykonywane (dopiero po upływie 3 jednostek czasu znaczniki te będą mogły być pobrane przez kolejne przejścia w sieci).

W tym rozdziale omówimy czasowe sieci kolorowane, które opisano w monografii [32]. Jest to jedno z kilku zaproponowanych w literaturze rozszerzeń czasowych sieci kolorowanych. Opis innego modelu czasu dla tej klasy sieci można znaleźć na przykład w pracy [1].

12.1. Wielozbiory czasowe

W podrozdziale 10.1 zajmowaliśmy się definicją wielozbioru i podstawowymi opercjami wykonywanymi na wielozbiorach. Rozważając czasowe sieci kolorowane, musimy do opisu wielozbioru dołączyć informacje o pieczętkach czasowych jego elementów.

Niech dany będzie zbiór *pieczętek czasowych* $R \subseteq \mathbb{R}$, zawierający wartość 0 i zamknięty ze względu na operację dodawania.

Definicja 12.1. *Wielozbiorem czasowym* A_t^* nad niepustym zbiorem A nazywamy dowolną funkcję

$$A_t^*: A \times R \rightarrow \mathbb{Z}_+. \quad (12.1)$$

Rodzinę wszystkich wielozbiorów czasowych zbudowanych nad zbiorem A będziemy oznaczać symbolem $2^{A_t^*}$. Każdy wielozbiór czasowy wyznacza jednoznacznie odpowiadający mu wielozbiór nieczasowy, otrzymywany poprzez pominięcie pieczętek czasowych jego elementów.

Rozważmy zbiór $A = \{a, b, c, d, e\}$ i przyjmijmy ponadto, że $R = \mathbb{Z}_+$. Przykładami wielozbiorów czasowych nad zbiorem A są wielozbiorы:

$$A_{t1}^* = \{(a, 0), (a, 12), (a, 15), (c, 7), (e, 3), (e, 6)\},$$

$$A_{t2}^* = \{(a, 0), (a, 14), (e, 7)\}.$$

Wielozbiory czasowe można również zapisywać algebraicznie:

$$A_{t1}^* = 3a[0, 12, 15] + c[7] + 2e[3, 6],$$

$$A_{t2}^* = 2a[0, 14] + e[7].$$

Wartości w nawiasach kwadratowych tworzą *listy pieczętek czasowych*.

Dowolny wielozbiór czasowy nad zbiorem A można więc zapisać jako

$$A_t^* = \sum_{x \in A} A_t^*(x) \ x \ A_t^*[x], \quad (12.2)$$

przy czym $A_t^*(x)$ oznacza *współczynnik* (liczbę wystąpień) elementu x w wielozbiorze A_t^* oraz $A_t^*[x]$ jest listą pieczętek czasowych o długości $A_t^*(x)$.

Dla wielozbiorów czasowych takie operacje jak dodawanie wielozbiorów czy iloczyn wielozbioru przez stałą, a także relację równości i rozmiar wielozbioru definiuje się analogicznie jak dla wielozbiorów nieczasowych. Różnice występują przy definiowaniu relacji mniejszości i operacji odejmowania wielozbiorów czasowych.

Niech dane będą dwie listy pieczętek czasowych $a = [a_1, a_2, \dots, a_m]$ oraz $b = [b_1, b_2, \dots, b_n]$ uporządkowane rosnąco. Mówimy, że $a \leq b$, jeżeli $m \leq n$ oraz $a_i \geq b_i$ dla $i = 1, 2, \dots, m$. Oznacza to, że żądamy, by mniejsza lista miała większe wartości pieczętek czasowych¹. Jeżeli $a \leq b$, to różnica $b - a$ jest listą o długości

¹ Jeżeli odniesiemy ten fakt do znaczników, to mniejszy wielozbiór musi zawierać wcześniejsze znaczniki.

$n - m$, uzyskaną z listy b w sposób następujący: z listy b usuwamy największą wartość, która jest mniejsza lub równa a_1 . Następnie z tak zmienionej listy b usuwamy największą² wartość, która jest mniejsza lub równa a_2 itd.

Definicja 12.2. Niech A_{t1}^* i A_{t2}^* będą dowolnymi wielozbiorami czasowymi nad zbiorem A . Relację *mniejszości* wielozbiorów czasowych definiujemy następująco:

$$A_{t2}^* \leq A_{t1}^* \Leftrightarrow \forall x \in A: A_{t2}^*[x] \leq A_{t1}^*[x] \quad (12.3)$$

Jeżeli $A_{t2}^* \leq A_{t1}^*$, to różnicę wielozbiorów czasowych A_{t1}^* i A_{t2}^* nazywamy wielozbior

$$A_{t1}^* - A_{t2}^* = \sum_{x \in A} (A_{t1}^*(x) - A_{t2}^*(x)) \cdot x \cdot (A_{t1}^*[x] - A_{t2}^*[x]). \quad (12.4)$$

■

Dla wielozbiorów czasowych:

$$A_{t1}^* = 3a[0, 12, 15] + c[7] + 2e[3, 6],$$

$$A_{t2}^* = 2a[0, 14] + e[7],$$

prawdziwe są zależności: $A_{t2}^* \leq A_{t1}^*$ i $A_{t1}^* - A_{t2}^* = a[15] + c[7] + e[3]$.

12.2. Dynamika sieci

Zaprezentowana w tym podrozdziale definicja czasowych sieci kolorowanych jest oparta na definicji 10.3 kolorowanych sieci niehierarchicznych. Zostało to podyktowane wyłącznie prostszą formą zapisu w porównaniu z sieciami hierarchicznymi. Hierarchiczne czasowe sieci kolorowane definiuje się podobnie jak hierarchiczne sieci nieczasowe.

Zbiór typów (kolorów) czasowej sieci kolorowanej jest podzielony na dwa podzbiory zawierające odpowiednio, typy czasowe i nieczasowe. Zdefiniowanie typu jako typu czasowego oznacza, że znaczniki tego typu będą miały przypisane pieczętki czasowe i ich dostępność dla przejść sieci będzie zależona od wartości tych pieczętek. Zastosowanie w czasowej sieci kolorowanej typów nieczasowych pozwala na korzystanie ze znaczników, które są zawsze dostępne niezależnie od wartości zegara.

² Przy odejmowaniu wielozbiorów czasowych w pierwszej kolejności będą usuwane znaczniki najwcześniejsze. Wymóg ten jest powiązany z koniecznością zapewnienia dowolnej kolejności wykonywania współbieżących przejść (zob. pracę [32]).

Definicja 12.3. Strukturę $\mathcal{N} = (\Sigma, P, T, A, \gamma, C, G, E, M_0, R, r_0)$ nazywamy niehierarchiczną *czasową siecią kolorowaną*, jeżeli są spełnione warunki:

- (1) Σ jest niepustym, skończonym zbiorem typów (kolorów), z których każdy jest zbiorem niepustym. Zbiór Σ może zawierać zarówno typy czasowe, jak i nieczasowe.
- (2) P jest niepustym, skończonym zbiorem miejsc.
- (3) T jest niepustym, skończonym zbiorem przejść.
- (4) A jest niepustym, skończonym zbiorem łuków, spełniającym warunek: $P \cap T = P \cap A = T \cap A = \emptyset$.
- (5) $\gamma: A \rightarrow (P \times T) \cup (T \times P)$ jest funkcją zaczepienia przypisującą każdemu łukowi uporządkowaną parę węzłów.
- (6) $C: P \rightarrow \Sigma$ jest funkcją typów (kolorów) określającą, jakiego typu znaczniki każde z miejsc może zawierać.
- (7) G jest funkcją zatrzeżeń (dozorów) przypisującą każdemu przejściu wyrażenie, takie że: $\forall t \in T: \mathcal{T}(G(t)) \subseteq \text{Bool} \wedge \mathcal{T}(\mathcal{V}(G(t))) \subseteq \Sigma$, tzn. wyrażenie mogące zawierać zmienne typów należących do Σ i którego dowolne wartościowanie daje w wyniku wartość logiczną.
- (8) E jest funkcją wag łuków przypisującą każdemu z łuków wyrażenie, takie że: $\forall a \in A: \mathcal{T}(\mathcal{V}(E(a))) \subseteq \Sigma$ i albo $\mathcal{T}(E(a)) \subseteq 2^{C(P(a))^*}$, jeżeli $C(P(a))$ jest typem nieczasowym, albo $\mathcal{T}(E(a)) \subseteq 2^{C(P(a))_t^*}$, jeżeli $C(P(a))$ jest typem czasowym. Waga łuku jest zatem wyrażeniem mogącym zawierać zmienne typów należących do Σ i którego dowolne wartościowanie daje w wyniku wielozbiór nad typem przypisanym do miejsca $P(a)$.
- (9) M_0 jest znakowaniem początkowym, takim, że dla dowolnego miejsca $p \in P$ $M_0(p) \in 2^{C(p)^*}$, jeżeli $C(p)$ jest typem nieczasowym, lub $M_0(p) \in 2^{C(p)_t^*}$, jeżeli $C(p)$ jest typem czasowym.
- (10) $R \subseteq \mathbb{R}$ jest zbiorem punktów czasowych (pieczętek czasowych) zawierającym wartość 0 i zamkniętym ze względu na operację dodawania.
- (11) $r_0 \in R$ jest wyróżnionym czasem początkowym.

Dla sieci czasowych pozostaje w mocy definicja wiążania sformułowana dla sieci kolorowanych (zob. definicję 10.4).

Niech \mathcal{N} będzie czasową siecią kolorowaną.

Definicja 12.4. Znakowaniem sieci \mathcal{N} nazywamy dowolną funkcję M określoną na zbiorze miejsc sieci, taką że dla dowolnego miejsca $p \in P$ znakowanie $M(p) \in 2^{C(p)^*}$, jeżeli $C(p)$ jest typem nieczasowym, lub $M(p) \in 2^{C(p)_t^*}$, jeżeli $C(p)$ jest typem czasowym.

Stanem sieci \mathcal{N} nazywamy parę (M, r) , przy czym M jest znakowaniem, $r \in R$ jest punktem czasowym. *Stanem początkowym* nazywamy parę (M_0, r_0) . ■

Niech (M, r) będzie stanem sieci. Podobnie jak dla nieczasowych sieci kolorowanych, wynik wartościowania wyrażenia $E(x, y)$ przy wiązaniu b będziemy oznaczać symbolem $E(x, y)_b$. Symbole $E(p, t)_b$ i $E(t, p)_b$ oznaczają odpowiednio wielozbiory znaczników, które są usuwane i dodawane do miejsca p , jeżeli przejście t jest wykonywane przy wiązaniu b . W zależności od typu przypisanego do miejsca p , mogą to być wielozbiory czasowe lub nieczasowe.

Definicja 12.5. Przejście t jest *aktywne* w stanie (M, r) przy wiązaniu $b \in \mathcal{B}(t)$, jeżeli spełniony jest warunek:

$$\forall p \in In(t): E(p, t)_b \leq M(p), \quad (12.5)$$

tzn., gdy:

- każde z miejsc wejściowych przejścia t zawiera przy wiązaniu b odpowiednią liczbę i odpowiednie wartości znaczników;
- jeżeli typ miejsca jest czasowy, to pieczętki czasowe tych znaczników umożliwiają ich usunięcie.

Przejście t jest *aktywne* w stanie (M, r) , jeżeli jest aktywne w stanie (M, r) przy pewnym swoim wiązaniu b . ■

Jeżeli przejście t jest aktywne w stanie (M, r) przy wiązaniu b , to w wyniku jego wykonania otrzymujemy nowy stan (M', r) , przy czym znakowanie M' jest określone zgodnie z wzorem (10.13). Należy przy tym zwrócić uwagę na fakt, że w przypadku czasowych sieci kolorowanych wyrażenia $E(a)_b$ mogą być wielozbiarami czasowymi.

Definicja 12.6. Niech (M, r) będzie stanem sieci \mathcal{N} . Mówimy, że zmiana stanu z (M, r) na (M, r') następuje w wyniku upływu czasu τ , jeżeli może wystąpić taki upływ czasu τ , tzn. żadne przejście sieci \mathcal{N} nie jest aktywne w żadnym ze stanów (M, r'') , przy czym $r \leq r'' < r'$ i $r', r'' \in R$. ■

Jeżeli zmiana stanu następuje w wyniku wykonania przejścia, to zachowują ważność oznaczenia (10.14) i (10.15). Jeżeli zmiana stanu następuje w wyniku upływu czasu τ , to fakt ten zapisujemy jako:

$$(M, r) \xrightarrow{\tau} (M, r'). \quad (12.6)$$

Dowolny stan sieci \mathcal{N} , który możemy otrzymać ze stanu (M, r) w wyniku wykonania skończonego ciągu przejść, nazywamy *stanem osiągalnym* ze stanu (M, r) . Zbiór wszystkich stanów osiągalnych ze stanu (M, r) oznaczamy symbolem $\mathcal{R}(M, r)$. Zbiór stanów osiągalnych ze stanu początkowego oznaczamy symbolem $\mathcal{R}(M_0, r_0)$. Rozważając czasowe sieci kolorowane, możemy mówić również o znakowaniach osiągalnych i używać wówczas oznaczenia $\mathcal{R}(M)$ (lub $\mathcal{R}(M_0)$). Z każdym stanem (M', r') osiągalnym ze stanu (M, r) jest związany ciąg przejść, którego wykonanie prowadzi od stanu (M, r) do (M', r') , przy czym zmiany stanów spowodowane wykonywaniem przejść mogą być przeplatane ze zmianami stanów spowodowanych upływem czasu. Zbiór wszystkich ciągów przejść, które można wykonać, rozpoczynając od stanu (M, r) , będziemy oznaczać symbolem $\mathcal{L}(M, r)$.

Dla czasowych sieci kolorowanych pojęcia takie jak *stan odtwarzalny*, *sieć odtwarzalna*, *sieć odwracalna* i *stan własny* definiuje się analogicznie jak dla sieci czasowych (zob. definicję 9.2).

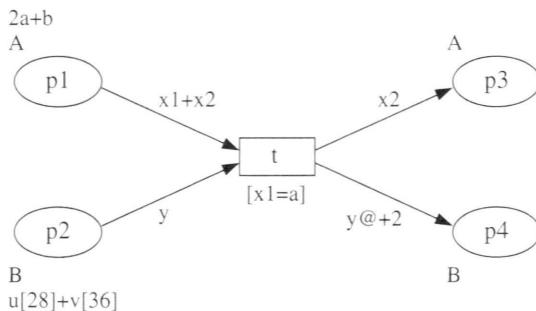
Każda czasowa sieć kolorowana wyznacza jednoznacznie sieć nieczasową, która powstaje w wyniku pominięcia wszystkich wyrażeń dotyczących modelu czasu i przyjęcia, że wszystkie typy są nieczasowe. Zbiór osiągalnych znakowań czasowej sieci kolorowanej jest podzbiorem zbioru osiągalnych znakowań odpowiadającej jej sieci nieczasowej. Podobną własność ma zbiór wszystkich ciągów przejść, które można wykonać, rozpoczynając od znakowania M_0 . Oznacza to, że sieci te charakteryzują się podobnym zachowaniem, przy czym dla sieci czasowej pewne ciągi przejść nie są wykonywane ze względu na uwarunkowania czasowe. Analizując więc własności sieci niezależne od czasu, można zastosować metody przedstawione w rozdz. 11. Dla czasowych sieci kolorowanych zachowują ważność definicje własności omawianych w podrozdz. 10.3 i 10.4.

Definicja 12.7. Niech (M, r) będzie stanem sieci \mathcal{N} . Stan sieci nazywamy *martwym*, jeżeli w stanie tym mogą występować zmiany stanu wyłącznie w wyniku upływu czasu. ■

Stan (M, r) czasowej sieci kolorowanej jest zatem stanem martwym, jeżeli M jest martwym znakowaniem sieci nieczasowej odpowiadającej sieci \mathcal{N} .

Rozważmy czasową sieć kolorowaną \mathcal{N} przedstawioną na rys. 12.1. Zbiór Σ typów tej sieci zawiera jeden typ nieczasowy (A) i jeden typ czasowy (B). Oznacza to, że znaczniki w miejscu $p1$ są zawsze dostępne, podczas gdy znaczniki w miejscu $p2$ mają przypisane pieczętki czasowe i ich dostępność dla przejścia t jest uzależniona od wartości zegara. Przejście t nie jest aktywne przy żadnym ze swoich wiązań w stanie $(M_0, 20)$, przy czym $M_0 = (2a + b, u[28] + v[36], \emptyset, \emptyset)$. Rozważmy wyrażenia przypisane poszczególnym łukom sieci \mathcal{N} .

time: 20



color A = with a | b;
 color B = with u | v timed;
 var x1, x2 : A;
 var y : B;

Rys. 12.1. Przykład czasowej sieci kolorowanej

- $E(p1, t) = x1 + x2$; przy dowolnym wiązaniu z miejsca $p1$ są pobierane dwa znaczniki; mogą to być dwa znaczniki o wartości a lub jeden znacznik o wartości a i jeden o wartości b .
- $E(p2, t) = y$; przy dowolnym wiązaniu z miejsca $p2$ jest pobierany jeden znacznik, którego pieczętka czasowa jest mniejsza niż aktualna wartość zegara, tzn. dowolne wartościowanie tego wyrażenia w stanie (M, r) daje w wyniku wielozbiór $u[r]$ lub $v[r]$.
- $E(t, p3) = x2$; przy dowolnym wiązaniu do miejsca $p3$ jest dodawany znacznik, którego wartość jest równa wartości przypisanej zmiennej $x2$ (może to być znacznik a lub b).
- $E(t, p4) = y@ + 2$; znak @ oddziela część wyrażenia, która służy do wyznaczenia elementów wielozbioru, od części, która służy do określenia pieczętek czasowych tych znaczników (symbol @+ może być traktowany jak operator). Przy dowolnym wiązaniu do miejsca $p4$ jest dodawany znacznik, którego wartość jest równa wartości przypisanej zmiennej y . Dowolne wartościowanie tego wyrażenia w stanie (M, r) daje w wyniku wielozbiór $u[r + 2]$ lub $v[r + 2]$.

Rozważmy teraz przejście t przy wiązaniu $b = (a/x1, b/x2, u/y)$. Przy wiązaniu b w stanie $(M_0, 20)$ wynikiem wartościowania wyrażeń przypisanych łukom sieci są następujące wielozbiory: $E(p1, t)_b = a + b$, $E(p2, t)_b = u[20]$, $E(t, p3)_b = b$

i $E(t, p4) = u[22]$. Przejście t nie jest aktywne przy wiązaniu b w stanie $(M_0, 20)$, gdyż $E(p2, t)_b \notin M(p2)$. W stanie $(M_0, 20)$ może wystąpić upływ czasu $\tau = 8$, który doprowadzi do stanu $(M_0, 28)$. W stanie tym aktywne jest przejście t przy wiązaniu b . W wyniku wykonania tego przejścia otrzymujemy nowy stan $(M_1, 28)$, przy czym $M_1 = (a, v[36], b, u[30])$. Stan $(M_1, 28)$ jest stanem martwym.

Jeżeli przejście czasowej sieci kolorowanej ma kilka miejsc wyjściowych z przypisanymi typami czasowymi, to znaczniki dodawane do tych miejsc mogą mieć różne pieczętki czasowe. Wyrażenie, które określa wartość pieczętki czasowej znacznika dodawanego do miejsca wyjściowego nie musi mieć wartości stałej. Może to być na przykład funkcja, której wartość należy do zbioru R ; wyniki takiej funkcji mogą zależeć od wartości dodawanego znacznika.

Rozważmy skończony ciąg przejść $\alpha = (t_1, b_1), (t_2, b_2), \dots, (t_n, b_n)$ prowadzących od stanu (M_1, r_1) , taki że:

$$(M_1, r_1) \xrightarrow{\tau_1} (M_1, r'_1) \xrightarrow{(t_1, b_1)} (M_2, r_2) \xrightarrow{\tau_2} \dots \\ \dots \xrightarrow{(t_n, b_n)} (M_{n+1}, r_{n+1}). \quad (12.7)$$

Bez straty ogólności rozważać można przyjąć, że między wykonaniem dwóch kolejnych przejść występuje co najwyżej jedna zmiana stanu sieci w wyniku upływu czasu (bierzemy pod uwagę maksymalny możliwy upływ czasu). Jeżeli wykonanie dwóch kolejnych przejść nie jest rozdzielone zmianą stanu w wyniku upływu czasu, to przyjmujemy, że odpowiednie τ_i jest równe 0.

Definicja 12.8. Czasem wykonania ciągu przejść α nazywamy sumę upływów czasu τ_i , którą oznaczamy symbolem $\delta(\alpha)$, tzn:

$$\delta(\alpha) = \sum_{i=1}^n \tau_i = r_{n+1} - r_1. \quad (12.8)$$

Czas wykonania ciągu przejść α jest określony jednoznacznie. Czas przejścia między stanami (M, r) i $(M', r') \in \mathcal{R}(M, r)$ jest równy różnicy $r' - r$. Częściej poszukuje się czasu przejścia od zadanego stanu (M, r) do stanu, który zawiera określone znakowanie M' . Może wtedy istnieć więcej niż jeden poszukiwany stan końcowy; można więc mówić o minimalnych i maksymalnych czasach przejścia.

Graf osiągalności dla czasowej sieci kolorowanej buduje się analogicznie jak dla nieczasowych sieci kolorowanych. Węzłami tego grafu są nie znakowania, ale stany sieci. Łuki grafu reprezentują zmiany stanów będące wynikiem wykonywanych przejść. Aby uzyskać czas przejścia między dwoma stanami połączonymi łukiem,

wystarczy odjąć od siebie punkty czasowe tych stanów. Przykład takiego grafu osiągalności omówimy w następnym podrozdziale.

Zbudowanie pełnego grafu osiągalności jest na ogół niemożliwe ze względu na nieskończoną przestrzeń stanów. Dla sieci czasowych można budować grafy relacyjne tak jak dla sieci nieczasowych.

□ 12.3. Modelowanie z zastosowaniem czasowych sieci kolorowanych

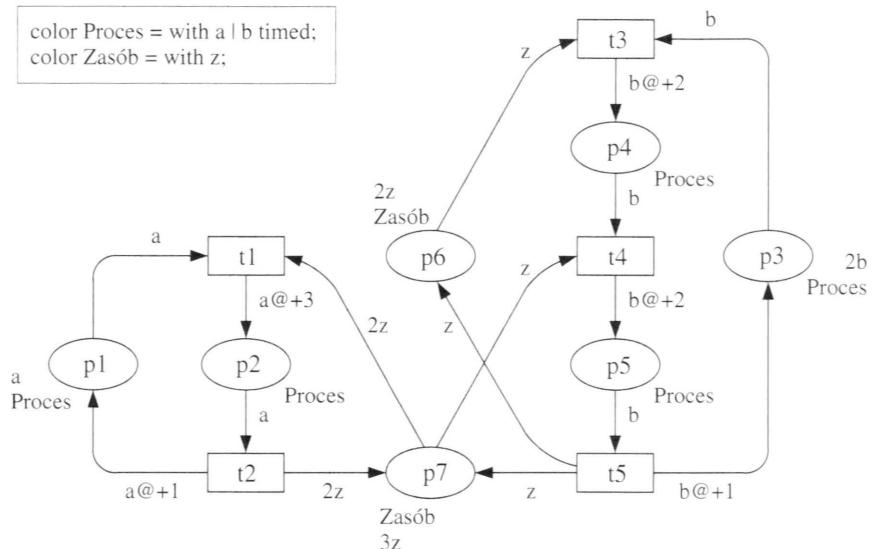
W tym podrozdziale omówimy przykłady wykorzystania czasowych sieci kolorowanych do modelowania systemów współbieżnych z czasem. Pierwszym przykładem będzie modyfikacja modelu współzawodnictwa procesów. W tej wersji modelu systemu wzbogacono o zależności czasowe. Drugim przykładem będzie model metra.

12.3.1. Model współzawodnictwa procesów

Rozważmy ponownie system przedstawiony na rys. 10.5, w którym o dwa źródła współdzielonych zasobów rywalizują ze sobą trzy współbieżnie działające procesy. W sieci przedstawionej na rys. 12.2 zdefiniowano typ czasowy *Proces*. Łuki wyjściowe przejść, które prowadzą do miejsc mogących przechowywać znaczniki typu *Proces*, mają więc przypisane wyrażenia zawierające część *czasową*.

Stanem początkowym rozważanej czasowej sieci kolorowanej \mathcal{N} jest para (M_0, r_0) , przy czym $M_0 = (a[0], \emptyset, 2b[0, 0], \emptyset, \emptyset, 2z, 3z)$ i $r_0 = 0$. W stanie początkowym aktywne są przejścia $t1$ i $t3$, przy czym przejście $t3$ może być wykonane współbieżnie ze sobą. Ponieważ wszystkie przejścia tej sieci mogą być wykonywane tylko przy trywialnych (pustych) wiązaniach, będziemy mówić o aktywnych przejściach, pomijając informację o wiązaniach. Wykonanie przejścia $t1$ prowadzi do stanu $(M_1, 0)$, przy czym $M_1 = (\emptyset, a[3], 2b[0, 0], \emptyset, \emptyset, 2z, z)$. W nowym stanie aktywne jest tylko przejście $t3$, a jego dwukrotne wykonanie prowadzi do stanu $(M_2, 0)$, przy czym $M_2 = (\emptyset, a[3], \emptyset, 2b[2, 2], \emptyset, \emptyset, z)$. W stanie $(M_2, 0)$ nie jest aktywne żadne przejście, ale możliwy jest upływ czasu $\tau = 2$ prowadzący do stanu $(M_2, 2)$. W stanie tym aktywne jest tylko przejście $t4$ (jednokrotnie, bo miejsce $p7$ zawiera tylko jeden znacznik). Ze względu na pieczętkę czasową znacznika w miejscu $p2$, nie jest aktywne przejście $t2$.

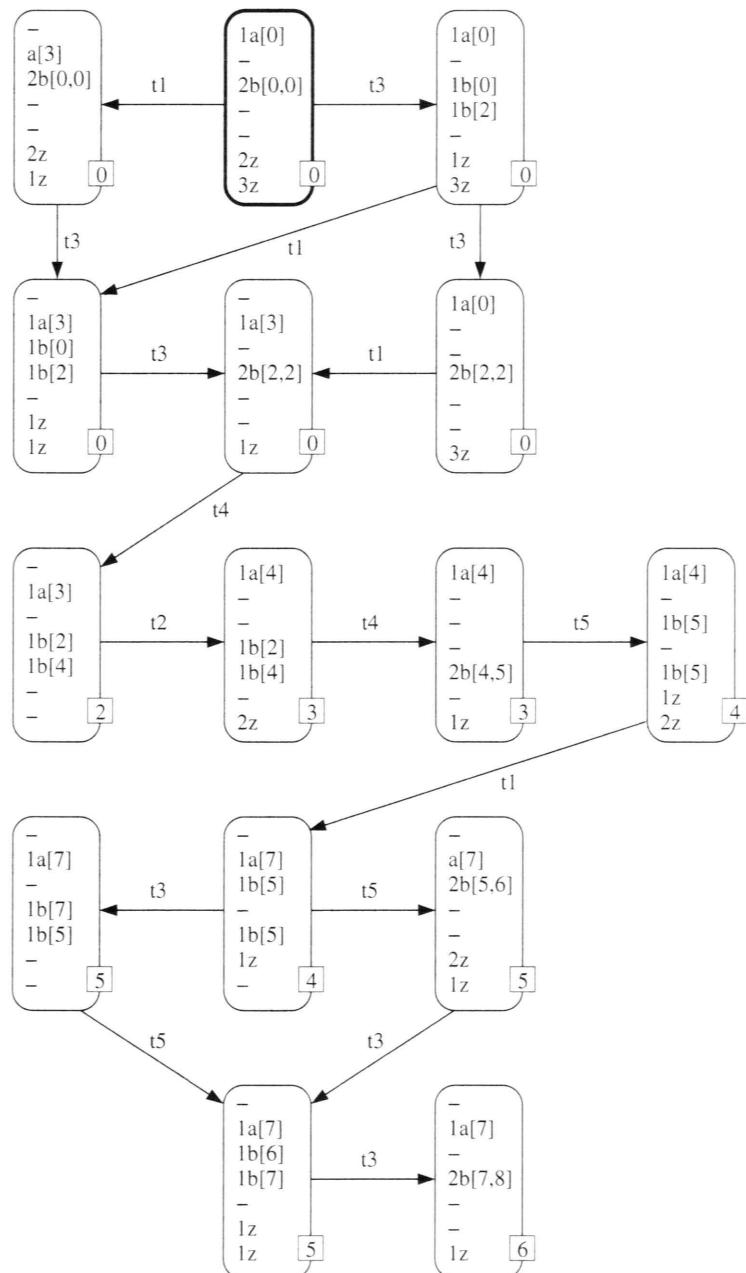
Opisany powyżej fragment pracy sieci \mathcal{N} pozwala zauważyc różnice występujące między kolorowanymi sieciami czasowymi a nieczasowymi. Jak już wiemy, każdej



Rys. 12.2. Model współzawodnictwa procesów – czasowa sieć kolorowana

sieci czasowej możemy przypisać odpowiadającą jej sieć nieczasową. Wszystkie znakowania osiągane przez sieć czasową są również znakowaniami osiąganymi przez odpowiadającą jej sieć nieczasową. Jeżeli sieć nieczasowa jest ograniczona, to również takie same ograniczenia dotyczą sieci czasowej (choć niekoniecznie muszą to być ograniczenia optymalne). Również wszystkie ciągi przejść, które można wykonać dla sieci czasowej, można wykonać dla sieci nieczasowej. Dla rozważanej sieci prawdziwe pozostają ograniczenia przedstawione w podrozdz. 11.1. Podobnie jak sieć na rys. 10.5, sieć \mathcal{N} jest żywa, ale w przeciwieństwie do sieci nieczasowej, jest siecią sprawiedliwą.

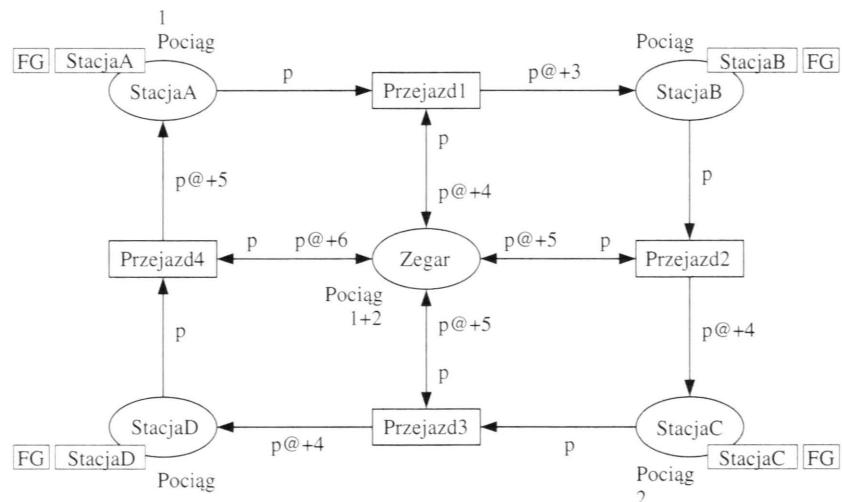
Jeżeli czasowa sieć kolorowana jest żywa, to zazwyczaj graf osiągalności sieci jest nieskończony. Na rysunku 12.3 przedstawiono fragment grafu osiągalności dla rozważanej sieci. Etykiety umieszczone przy węzłach grafu osiągalności reprezentują punkty czasowe. Łatwo zauważyc, że graf zawiera wszystkie osiągalne znakowania sieci \mathcal{N} oraz żaden ze stanów sieci nie ma szansy się powtórzyć (stan początkowy nie jest odtwarzalny).



Rys. 12.3. Fragment grafu osiągalności dla czasowej sieci kolorowanej

12.3.2. Model metra

Rozważmy uproszczony model metra składający się z czterech stacji, przez które przejeżdżają cyklicznie dwa pociągi. Model powinien przedstawić następujące aspekty systemu: aktualne położenie pociągów, liczbę pasażerów podróżujących w danej chwili poszczególnymi pociągami i liczbę pasażerów oczekujących na kolejnych stacjach. Ponadto ruch pociągów powinien być tak zorganizowany, aby nigdy nie znalazły się one jednocześnie na tej samej stacji.

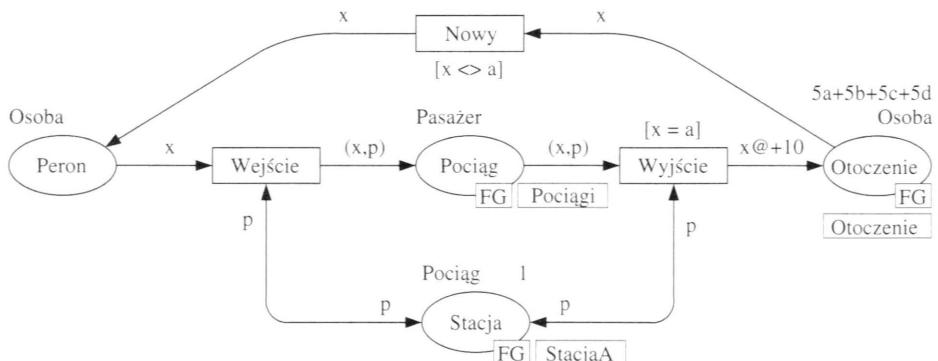


Rys. 12.4. Strona *Trasa* modelu metra

Rozważany model ma postać hierarchicznej czasowej sieci kolorowanej, przy czym do łączenia stron modelu użyto wyłącznie globalnych fuzji miejsc. Na rysunku 12.4 przedstawiono stronę *Trasa* modelu metra. Strona ta reprezentuje między innymi strukturę linii metra i czas przejazdu między poszczególnymi stacjami (stacje metra oznaczono literami od A do D). Pociągi (w modelu oznaczone numerami 1 i 2) rozpoczynają jazdę odpowiednio od stacji A i C. Dodatkowe miejsce *Zegar* służy do zagwarantowania, że postój pociągu po przyjeździe na daną stację będzie trwał jedną jednostkę czasu. W modelu użyto następujących typów i zmiennych:

```
color Osoba = with a | b | c | d timed;
color Pociąg = int with 1..2 timed;
color Pasażer = product Osoba * Pociąg;
var x : Osoba;
var p : Pociąg;
```

Potraktowanie strony *Trasa* jak całości modelu pozwala śledzić wyłącznie ruch pociągów. Czas trwania jednego cyklu pracy pociągu (od chwili wyruszenia ze stacji początkowej do ponownego wyruszenia w kolejnym cyklu) dla każdego z pociągów wynosi 20 jednostek czasu (inne strony w modelu nie mają wpływu na te zależności czasowe). Ponieważ pociągi rozpoczynają jazdę na różnych stacjach, a czasy przejazdu między stacjami dla obu pociągów są takie same, więc nigdy nie spotkają się na tej samej stacji.



Rys. 12.5. Strona *StacjaA* modelu metra

Na rysunku 12.5 przedstawiono stronę *StacjaA* modelu metra. Miejsce *Otoczenie* służy do ograniczenia liczby pasażerów poruszających się w rozważanym systemie. Wartość znacznika tego typu oznacza jednocześnie stację docelową podróżującego. Przejście *Nowy* umieszcza na peronie stacji nowego podróżnego (zastrzeżenie tego przejścia wyklucza możliwość umieszczenia pasażera *a* na peronie stacji A). Jeżeli na stacji znajduje się pociąg, to osoby które znajdują się na peronie, wsiadają do pociągu, a przebywające w pociągu osoby, które dotarły do stacji docelowej, wysiadają. Pasażerowie znajdujący się w pociągach są reprezentowani przez znaczniki umieszczone w miejscu *Pociąg*. Znaczniki te są parami; drugi element pary oznacza numer pociągu, którym dana osoba podróżuje.

Kompletny model zawiera jeszcze trzy podobne strony: *StacjaB*, *StacjaC* i *StacjaD*. Różnice między tymi stronami dotyczą fuzji, do której należy miejsce *Stacja* (w zależności od fuzji znakowanie początkowe tego miejsca jest zbiorem pustym lub nie), oraz zastrzeżeń przejść *Nowy* i *Wyjście*.

Rozważana sieć jest siecią ograniczoną. Dla równoważnej jej sieci nieczasowej można wskazać poniższe niezmienne miejsca, na których podstawić można podać

ograniczenia liczbowe i wielozbiorowe miejsc sieci:

$$M(Zegar) = 1 + 2;$$

$$M(StacjaA) + M(StacjaB) + M(StacjaC) + M(StacjaD) = 1 + 2; \quad (12.9)$$

$$M(Otoczenie) + f(M(Pociąg)) + M(Peron_A) +$$

$$M(Peron_B) + M(Peron_C) + M(Peron_D) = 5a + 5b + 5c + 5d.$$

Funkcja f rzutuje wielozbiory nad typem *Pasażer* na wielozbiory nad typem *Osoba*, pomijając numery pociągów.

Sieć jest żywa i sprawiedliwa. W przeciwnieństwie do zbioru osiągalnych znakowań, zbiór wszystkich osiągalnych stanów sieci jest nieskończony. Ponadto, odpowiadająca rozważanej sieci, sieć nieczasowa jest odwracalna.

12.4. Podsumowanie

W tym rozdziale omówiliśmy podstawowe pojęcia dotyczące czasowych sieci kolorowanych. W porównaniu z CP-sieciami przedstawionymi w rozdz. 10, znaczniki występujące w modelu mogą mieć przypisane pieczętki czasowe. Wartość takiej pieczętki oznacza punkt czasowy, od którego począwszy znacznik będzie dostępny dla przejść w sieci. Do odmierzania upływającego czasu służy globalny zegar. Czas wykonania danego przejścia jest reprezentowany przez pieczętki czasowe znaczników dodawanych do jego miejsc wyjściowych. Wartości pieczętek dla takich znaczników są ustalane jako suma wartości zegara w chwili wykonywania danego przejścia i wymaganego opóźnienia.

Do analizowania czasowych sieci kolorowanych można korzystać z metod przedstawionych w rozdz. 11, przy czym w przypadku sieci czasowych węzły grafów osiągalności reprezentują stany sieci, a nie znakowania. W efekcie na ogół grafy takie są nieskończone.



RTCP-sieci

Występujący w nazwie sieci skrót RTCP pochodzi od słów użytych w ich pełnej nazwie w jęz. ang.: *Real-Time Coloured Petri*. RTCP-sieci powstały jako wynik adaptacji czasowych kolorowanych sieci Petriego do modelowania systemów czasu rzeczywistego (zob. prace: [77] i [81]). Stanowią one podklasę sieci kolorowanych, która charakteryzuje się m.in. zupełnie odmiennym modelem czasu, występowaniem priorytetów przejść i zmodyfikowanym podejściem do tworzenia modeli hierarchicznych. Wprowadzone zmiany miały na celu ułatwić i przyspieszyć tworzenie modeli oraz umożliwić bezpośrednie modelowanie pojęć typowych dla programowania współbieżnego, takich jak przeterminowanie, priorytety zadań itp.

W tym rozdziale omówimy definicje niehierarchicznych RTCP-sieci oraz pojęć charakterystycznych dla tej podklasy sieci. Większość pojęć i oznaczeń typowych dla kolorowanych sieci Petriego zachowuje ważność również dla RTCP-sieci. Pojawiły się nieliczne nowe symbole i pojęcia, a niektóre z uprzednio wprowadzonych – zmieniły znaczenie.

13.1. Niehierarchiczne RTCP-sieci

Przedstawiona w tym podrozdziale definicja RTCP-sieci dotyczy sieci niehierarchicznych. Jak się wkrótce przekonamy, nie ma potrzeby formalnego wprowadzania hierarchicznych RTCP-sieci, mimo że przedstawione przykłady mają w większości taką formę. Dla RTCP-sieci obowiązują definicje i oznaczenia wprowadzone na początku podrozdz. 10.2, na przykład: $\mathcal{T}(v)$, $\mathcal{V}(x)$, $Bool$ itd.

Podstawowe różnice między sieciami kolorowanymi a RTCP-sieciami polegają na wprowadzeniu ograniczeń dotyczących struktury RTCP-sieci, nowego modelu czasu oraz priorytetów przejść. Ponieważ w RTCP-sieciach nie są dopuszczalne łuki wielokrotne, więc zbiór łuków zdefiniowano jako relację (podobnie jak w sieciach niskiego poziomu). Każdy łuk ma przypisane dwa wyrażenia: *wagę łuku* i *wyrażenie czasowe*, powiązane z modelem czasu. Z definicji funkcja wag łuków E_M

przypisuje wszystkim łukom sieci wyrażenia jednolite z wielokrotnością 1; oznacza to, że dowolne dopuszczalne wartościowanie takiego wyrażenia daje w wyniku pojedynczy znacznik należący do odpowiedniego typu. *Funkcja wag czasowych* E_S przypisuje wszystkim łukom sieci wyrażenia czasowe, których dowolne wartościowanie daje w wyniku nieujemną liczbę wymierną.

W rozdziale 12 zajmowaliśmy się modelem czasu stosowanym w sieciach kolorowanych. Uzależnienie pieczętek czasowych przypisywanych do znaczników od wartości globalnego zegara powoduje, że żywe czasowe sieci kolorowane mają na ogół nieskończone zbiory osiągalnych stanów. Wprowadzony w RTCP-sieciach nowy model czasu łączy pieczętki czasowe z miejscami sieci. Dodatnia wartość pieczętki czasowej określa, jak długo znaczniki w danym miejscu pozostaną jeszcze niedostępne. Z każdym taktiem zegara wartość pieczętki czasowej maleje, a kiedy osiągnie zero, wtedy znaczniki stają się dostępne dla przejść sieci. Kolejne tatory zegara mogą spowodować, że pieczętka czasowa będzie przyjmować wartości ujemne. Określa się to jako *starzenie się znaczników*; na przykład wartość -3 oznacza wiek równy 3 jednostkom czasu. Dla dowolnego przejścia można określić, czy może usunąć znacznik z danego swojego miejsca wejściowego bezpośrednio po tym, jak stanie się on dostępny, czy też musi czekać, aż osiągnie odpowiedni wiek (decydują o tym *wyrażenia czasowe łuków*). Podobnie jak znakowanie początkowe, określony jest wstępny rozkład pieczętek czasowych w sieci.

Definicja 13.1. Strukturę $\mathcal{N} = (\Sigma, P, T, A, C, G, I, E_M, E_S, M_0, S_0)$ nazywamy *RTCP-siecią*, jeżeli są spełnione warunki:

- (1) Σ jest niepustym, skończonym zbiorem typów (*kolorów*), z których każdy jest zbiorem niepustym.
- (2) P jest niepustym, skończonym zbiorem *miejsc*.
- (3) T jest niepustym, skończonym zbiorem przejść, takim że $P \cap T = \emptyset$.
- (4) $A \subseteq (P \times T) \cup (T \times P)$ jest zbiorem *łuków* sieci.
- (5) $C: P \rightarrow \Sigma$ jest funkcją typów (*kolorów*), która określa, jakiego typu znaczniki może zawierać każde miejsce.
- (6) G jest funkcją zastrzeżeń (*dozorów*), która każdemu przejściu przypisuje takie wyrażenie, że: $\forall t \in T: \mathcal{T}(G(t)) \subseteq \text{Bool} \wedge \mathcal{T}(\mathcal{V}(G(t))) \subseteq \Sigma$, tzn. wyrażenie mogące zawierać zmienne typów należących do Σ i którego dowolne wartościowanie daje w wyniku wartość logiczną.
- (7) $I: T \rightarrow \mathbb{Z}_+$ jest funkcją priorytetów, która każdemu przejściu przypisuje jego priorytet.
- (8) E_M jest funkcją wag łuków, która każdemu łukowi przypisuje takie wyrażenie, że: $\forall a \in A: \mathcal{T}(E_M(a)) \subseteq C(P(a)) \wedge \mathcal{T}(\mathcal{V}(E_M(a))) \subseteq \Sigma$, tzn. wyrażenie mogące

zawierać zmienne typów należących do Σ i którego dowolne wartościowanie daje w wyniku pojedynczy znacznik należący do typu miejsca $P(a)$.

- (9) E_S jest funkcją wag czasowych, która każdemu łukowi przypisuje takie wyrażenie czasowe, że: $\forall a \in A: \mathcal{T}(E_S(a)) \subseteq \mathbb{Q}_+ \wedge \mathcal{T}(\mathcal{V}(E_S(a))) \subseteq \Sigma$, tzn. wyrażenie mogące zawierać zmienne typów należących do Σ i którego dowolne wartościowanie daje w wyniku nieujemną liczbę wymierną.
- (10) M_0 jest znakowaniem początkowym, takim że $\forall p \in P: M_0(p) \in 2^{C(p)^*}$, tzn. M_0 jest funkcją, która każdemu miejscu przyporządkowuje wielozbiór nad typem przypisanym do tego miejsca.
- (11) $S_0: P \rightarrow \mathbb{Q}$ jest początkowym rozkładem pieczętek czasowych.

Jeżeli pominiemy nowy model czasu i funkcję priorytetów, to każda RTCP-sieć będzie siecią kolorowaną. Dla RTCP-sieci zachowują ważność definicje 10.4 i 10.5, przy czym odnoszą się one teraz do innej klasy sieci. Należy ponadto zwrócić uwagę, że zbiór $\mathcal{V}(t)$ zawiera również zmienne występujące w wyrażeniach czasowych łuków ze zbioru $A(t)$.

Przyjmiemy założenie, że zbiór P jest zbiorem uporządkowanym, $P = \{p_1, p_2, \dots, p_n\}$. W związku z tym znakowanie sieci może być reprezentowane w postaci wektora. Symbole $E_M(a)_b$ i $E_S(a)_b$ (przy czym $a \in A$) będą oznaczać odpowiednio wartości wyrażeń $E_M(a)$ i $E_S(a)$ przy wiązaniu b . Zakładamy ponadto, że dla dowolnego wiązania $b \in \mathcal{B}(T(a))$ wartości $E_M(a)_b$ i $E_S(a)_b$ są określone jednoznacznie.

Niech dana będzie RTCP-sieć $\mathcal{N} = (\Sigma, P, T, A, C, G, I, E_M, E_S, M_0, S_0)$.

Definicja 13.2. Rozkładem pieczętek czasowych sieci \mathcal{N} nazywamy dowolną funkcję S określoną na zbiorze miejsc sieci, taką że

$$\forall p \in P: S(p) \in \mathbb{Q}. \quad (13.1)$$

■

Biorąc pod uwagę założenie o uporządkowaniu zbioru P , również funkcję S można przedstawić w postaci wektora o $|P|$ zmiennych. Zamiast określenia rozkład będziemy również używać określenia wektor pieczętek czasowych.

Definicja 13.3. Stanem sieci \mathcal{N} nazywamy parę (M, S) , przy czym M jest znakowaniem, S zaś wektorem pieczętek czasowych. Parę (M_0, P_0) nazywamy stanem początkowym. ■

Definicja 13.4. Przejście $t \in T$ jest *aktywne* w stanie (M, S) przy wiązaniu b , jeżeli spełnione są warunki:

$$\begin{aligned} \forall p \in In(t): E_M(p, t)_b &\in M(p) \wedge E_S(p, t)_b \leq -S(p), \\ \forall p \in Out(t): S(p) &\leq 0 \end{aligned} \quad (13.2)$$

oraz dla dowolnego przejścia $t' \neq t$, które przy pewnym wiązaniu $b' \in \mathcal{B}(t')$ spełnia powyższe warunki, $I(t') \leq I(t)$ lub $In(t) \cap In(t') = Out(t) \cap Out(t') = \emptyset$. ■

Innymi słowy, przejście t jest aktywne w stanie (M, S) przy wiązaniu b , jeżeli wszystkie miejsca wejściowe zawierają odpowiednie znaczniki, mają odpowiednie wartości pieczętek czasowych, wszystkie miejsca wyjściowe są dostępne oraz żadne inne przejście o wyższym priorytecie nie rywalizuje z przejściem t o to samo miejsce wejściowe lub wyjściowe.

Przejście $t \in T$ jest *aktywne* w stanie (M, S) , jeżeli jest aktywne w stanie (M, S) przy pewnym swoim wiązaniu b .

Jeżeli przejście t jest aktywne w stanie (M_1, S_1) przy wiązaniu b , to może zostać wykonane, zmieniając stan (M_1, S_1) na nowy stan (M_2, S_2) , taki że:

$$M_2(p) = \begin{cases} M_1(p) - \{E_M(p, t)_b\}, & \text{gdy } p \in In(t) - Out(t), \\ M_1(p) - \{E_M(p, t)_b\} \cup \{E_M(t, p)_b\}, & \text{gdy } p \in In(t) \cap Out(t), \\ M_1(p) \cup \{E_M(t, p)_b\}, & \text{gdy } p \in Out(t) - In(t), \\ M_1(p) & \text{w innym przypadku.} \end{cases} \quad (13.3)$$

$$S_2(p) = \begin{cases} E_S(t, p)_b, & \text{gdy } p \in Out(t), \\ 0, & \text{gdy } p \in In(t) - Out(t), \\ S_1(p) & \text{w innym przypadku.} \end{cases} \quad (13.4)$$

Jeżeli przejście t jest aktywne w stanie (M_1, S_1) przy wiązaniu b i w wyniku jego wykonania otrzymujemy stan (M_2, S_2) , to fakt ten zapisujemy jako:

$$(M_1, S_1) \xrightarrow{(t,b)} (M_2, S_2). \quad (13.5)$$

O stanie (M_2, S_2) mówimy, że jest *bezpośrednio osiągalny* ze stanu (M_1, S_1) .

Podobnie jak o stanach, mówimy o znakowaniu M_2 , że jest *bezpośrednio osiągalne* ze znakowania M_1 ; fakt ten zapisujemy jako:

$$M_1 \xrightarrow{(t,b)} M_2. \quad (13.6)$$

Jeżeli przy rozważaniu zmiany znakowania interesować nas będzie tylko nazwa przejścia, to będziemy stosować zapis uproszczony:

$$(M_1, S_1) \xrightarrow{t} (M_2, S_2), \quad (13.7)$$

$$M_1 \xrightarrow{t} M_2.$$

Jeżeli w pewnym stanie (M, S) aktywne są dwa przejścia, ale wykonanie dowolnego z nich powoduje utratę aktywności przez drugie, to mówimy, że przejścia te pozostają ze sobą w *konflikcie*.

Do mierzenia czasu w RTCP-sieciach stosowany jest zegar globalny, podobnie jak w sieciach kolorowanych. Każdy takt zegara powoduje zmniejszenie wartości pieczętek czasowych o taką wartość, o jaką zmieniła się przy tym takcie wartość zegara. Wynika stąd następujący wniosek: każde wykonanie przejścia sieci zmienia (może zmienić) zarówno wektor M , jak i S , natomiast każdy takt zegara zmienia wektor S , ale nie zmienia wektora M .

Niech e oznacza wektor o $|P|$ współrzędnych równych 1.

Definicja 13.5. Niech (M, S) będzie stanem sieci \mathcal{N} . Mówimy, że następuje *zmiana stanu* z (M, S) na (M, S') w wyniku upływu czasu $\tau \in \mathbb{Q}_+$, jeżeli jest możliwy upływ czasu τ , tzn. żadne przejście $t \in T$ nie jest aktywne w żadnym ze stanów (M, S'') , przy czym

$$S'' = S - \tau' \cdot e, \text{ dla } 0 \leq \tau' < \tau \text{ i } \tau' \in \mathbb{Q}_+. \quad (13.8)$$

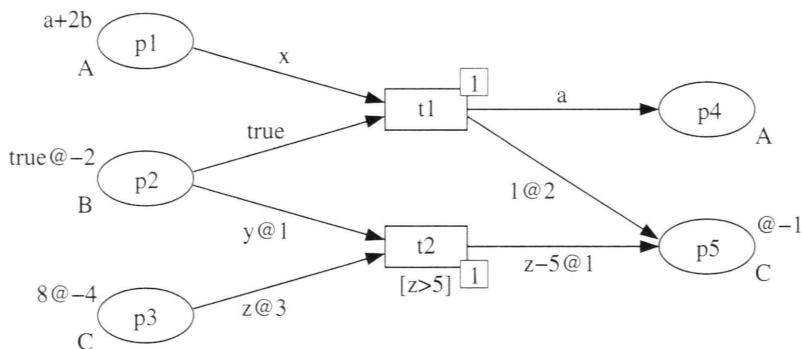
■

Jeżeli zmiana stanu RTCP-sieci następuje w wyniku upływu czasu τ , to fakt ten zapisujemy jako:

$$(M, S) \xrightarrow{\tau} (M', S'). \quad (13.9)$$

Definicja 13.6. Stan (M, S) nazywamy *martwym*, jeżeli możliwe są wyłącznie zmiany stanu wynikające z upływu czasu. ■

Zakładamy, że przejście, które staje się aktywne (czy to w wyniku wykonania innego przejścia, czy też upływu czasu), jest natychmiast wykonywane, chyba że jest w konflikcie z innym przejściem i traci aktywność w wyniku wykonania tego przejścia. Oznacza to, że niedopuszczalna jest sytuacja, w której aktywne przejście nie jest wykonywane, a znaczniki w miejscach wejściowych tego przejścia się starzeją.



Rys. 13.1. Przykład RTCP-sieci (wersja 1)

Rozważmy przykład sieci N_1 przedstawionej na rys. 13.1. W modelu występują następujące typy i zmienne:

```

color A = with a | b | c;
color B = bool with (false, true);
color C = int with 0..9;
var x : A;
var y : B;
var z : C;

```

Na rysunku 13.1 zastosowano pewne uproszczenia. Nie umieszczone na nim: zastrzeżenia przejścia równego stałej *true*, równych 0 wyrażeń czasowych łyków, znakowań początkowych miejsc będących zbiorami pustymi oraz równych 0 pieczętek czasowych.

W stanie początkowym (M_0, S_0):

$$\begin{aligned} M_0 &= (a + 2b, \text{true}, 8, -, -), \\ S_0 &= (0, -2, -4, 0, -1), \end{aligned} \quad (13.10)$$

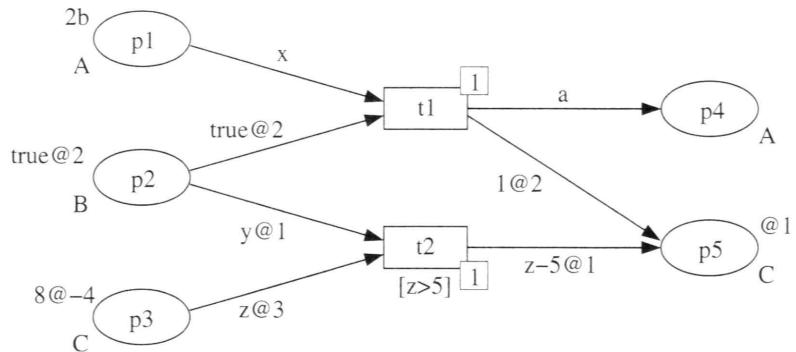
aktywne są oba przejścia sieci. Przejście $t1$ jest aktywne przy dwóch różnych wiązaniach: $b_1 = (a/x)$ i $b_2 = (b/x)$, a przejście $t2$, przy wiązaniu $b = (\text{true}/y, 8/z)$. Ponieważ oba przejścia mają takie same priorytety i rywalizują o dostęp do miejsca $p2$, więc pozostają ze sobą w konflikcie, a zatem tylko jedno z nich może zostać wykonane. Wynikiem wykonania przejścia $t1$ przy wiązaniu b_1 jest stan (M_1, S_1):

$$\begin{aligned} M_1 &= (2b, -, 8, a, 1), \\ S_1 &= (0, 0, -4, 0, 2), \end{aligned} \quad (13.11)$$

a wynikiem wykonania przejścia t_2 , przy wiązaniu b jest stan (M_2, S_2) :

$$\begin{aligned} M_2 &= (a + 2b, -, -, -, 3), \\ S_2 &= (0, 0, 0, 0, 1). \end{aligned} \quad (13.12)$$

Jeżeli przejścia t_1 i t_2 miałyby różne priorytety, to między przejściami nie byłoby konfliktu, gdyż aktywne byłoby tylko przejście o wyższym priorytecie.



Rys. 13.2. Przykład RTCP-sieci (wersja 2)

Rozważmy przykład sieci N_2 przedstawionej na rys. 13.2. W porównaniu z siecią N_1 na rys. 13.1 zmianie uległo wyrażenie czasowe łuku (p_2, t_1) oraz zmienił się stan początkowy:

$$\begin{aligned} M_0 &= (2b, true, 8, -, -), \\ S_0 &= (0, 2, -4, 0, 1). \end{aligned} \quad (13.13)$$

Ze względu na pieczętki czasowe miejsc, żadne z przejść sieci nie jest aktywne w stanie początkowym. Aktywność każdego przejścia można rozważyć tylko w kontekście jednego wiązania. Dla przejścia t_1 jest to wiązanie $b_1 = (b/x)$, a dla przejścia t_2 , wiązanie $b_2 = (true/y, 8/z)$. Miejsca wejściowe tych przejść zawierają odpowiednie znaczniki, ale niedostępne jest miejsce wyjściowe p_5 . Ponadto w przypadku przejścia t_1 znaczniak w miejscu p_2 powinien mieć wiek równy 2, a w przypadku przejścia t_2 wymagany jest wiek znacznika równy 1. W stanie (M_0, S_0) możliwy jest upływ czasu $\tau = 3$ prowadzący do stanu (M_0, S'_0) :

$$S'_0 = (-3, -1, -7, -3, -2). \quad (13.14)$$

W stanie (M_0, S'_0) aktywne jest tylko przejście t_2 , a jego wykonanie prowadzi do stanu (M_1, S_1) :

$$\begin{aligned} M_1 &= (2b, -, -, -, 3), \\ S_1 &= (-3, 0, 0, -3, 1). \end{aligned} \quad (13.15)$$

Jeżeli zmieni się znakowanie początkowe miejsca p_3 , na przykład $M_0(p_3) = 4$, to dla sieci \mathcal{N}_2 w stanie początkowym (M_0, S_0) możliwy jest upływ czasu $\tau = 4$, po którym aktywne będzie przejście t_1 .

Dowolny stan sieci \mathcal{N} , który możemy otrzymać ze stanu (M, S) po wykonaniu skończonego ciągu przejść, nazywamy *stanem osiągalnym* ze stanu (M, S) . Zbiór wszystkich stanów osiągalnych ze stanu (M, S) oznaczamy symbolem $\mathcal{R}(M, S)$. Zbiór stanów osiągalnych ze stanu początkowego oznaczamy symbolem $\mathcal{R}(M_0, S_0)$. Rozważając RTCP-sieci, możemy mówić również o znakowaniach osiągalnych i używać oznaczenia $\mathcal{R}(M)$ (lub $\mathcal{R}(M_0)$). Z każdym stanem (M', S') osiągalnym ze stanu (M, S) jest związany ciąg przejść, których wykonanie prowadzi od stanu (M, S) do (M', S') . Zbiór wszystkich ciągów przejść, które można wykonać, rozpoznając od stanu (M, S) , będziemy oznaczać symbolem $\mathcal{L}(M, S)$. Zbiór $\mathcal{L}(M, S)$ zawiera zarówno skończone, jak i nieskończone ciągi przejść.

Dla RTCP-sieci pojęcia takie jak *stan odtwarzalny*, *sieć odtwarzalna*, *sieć odwracalna* i *stan własny* definiuje się analogicznie jak dla sieci czasowych (zob. definicję 9.2).

Rozważmy skończony ciąg przejść $\alpha = (t_1, b_1), (t_2, b_2), \dots, (t_n, b_n)$ prowadzących od stanu (M_1, S_1) , taki że:

$$(M_1, S_1) \xrightarrow{\tau_1} (M_1, S'_1) \xrightarrow{(t_1, b_1)} (M_2, S_2) \xrightarrow{\tau_2} \cdots \xrightarrow{(t_n, b_n)} (M_{n+1}, S_{n+1}). \quad (13.16)$$

Bez straty ogólności rozważać można przyjąć, że między wykonaniem dwóch kolejnych przejść występuje co najwyżej jedna zmiana stanu sieci w wyniku upływu czasu (bierzemy pod uwagę maksymalny możliwy upływ czasu). Jeżeli wykonanie dwóch kolejnych przejść nie jest rozdzielone zmianą stanu w wyniku upływu czasu, to przyjmujemy, że odpowiednie τ_i jest równe zero.

Definicja 13.7. Czasem wykonania ciągu przejść α nazywamy sumę upływów czasu τ_i i oznaczamy go symbolem $\delta(\alpha)$, tzn.

$$\delta(\alpha) = \sum_{i=1}^n \tau_i. \quad (13.17)$$

Definicja 13.8. Niech (M, S) oraz (M', S') będą takimi stanami, że $(M', S') \in \mathcal{R}(M, S)$. Czasem przejścia od stanu (M, S) do (M', S') nazywamy czas wykonania dowolnego ciągu przejść α prowadzącego od stanu (M, S) do (M', S') i oznaczamy go symbolem $\delta((M, S), (M', S'))$. ■

Dla RTCP-sieci czas przejścia między stanami nie jest określony jednoznacznie. Jeżeli istnieje wiele ciągów przejść prowadzących od stanu (M, S) do (M', S') , to otrzymujemy również wiele potencjalnie różnych czasów przejść. Najczęściej szuka się minimalnego i maksymalnego czasu przejścia między stanami.

RTCP-sieci mogą być traktowane jako podklasa sieci kolorowanych. Przedstawione w podrozdz. 10.3 i 10.4 definicje własności sieci kolorowanych obowiązują również w odniesieniu do RTCP-sieci.

13.2. Hierarchiczne RTCP-sieci

Jak już wspomniałem, hierarchicznych RTCP-sieci nie definiuje się w sposób formalny. Hierarchizacja modelu należy raczej do praktyki inżynierskiej, gdyż ułatwia opis dużych systemów. Koncepcję hierarchicznych RTCP-sieci oparto na hierarchicznych sieciach kolorowanych. Do łączenia stron modelu używa się tych samych mechanizmów, tj. podstawianych przejść i fuzji miejsc. Wprowadzono jednak pewne ograniczenia w zakresie ich stosowania.

W RTCP-sieciach mogą występować tylko trzy typy portów: wejściowe, wyjściowe i wejściowo-wyjściowe. Każdy port sieci musi mieć przypisane dokładnie jedno gniazdo tego samego typu i na odwrót, każdemu gniazdu musi być przypisany dokładnie jeden port. Oczywiście gniazdo i port muszą mieć takie same typy (kolory) i znakowania początkowe. Nazwa gniazda zasłania nazwę portu, dlatego w opisie stanów sieci używa się wyłącznie nazw miejsc gniazdowych.

Stosowanie fuzji miejsc ograniczono do fuzji globalnych, przy czym nazwa fuzji zasłania nazwy miejsc do niej należących. W RTCP-sieciach nie używa się instancji stron w takim znaczeniu jak w sieciach kolorowanych. Jeżeli jakaś strona występuje wielokrotnie w modelu, to każda jej kopia ma własny numer węzła w grafie hierarchii stron. Wszystkie przedstawione ograniczenia dotyczące konstruowania modeli hierarchicznych pozwalają łatwo przekształcić hierarchiczną RTCP-sieć na równoważną jej sieć niehierarchiczną.

Omawiane poniżej hierarchiczne RTCP-sieci są przedstawione w tzw. postaci kanonicznej. Modele w postaci hierarchicznej składają się z czterech typów stron o ścisłe określonej strukturze i roli, którą odgrywają. Struktura modelowanego systemu

jest reprezentowana przez *podstawowe strony miejsc i podstawowe strony przejść*. Pierwsze z nich służą do reprezentowania aktywnych obiektów¹ i powiązanych z nimi aktywności systemu. Strony te są przeznaczone do reprezentowania składników systemu i tworzą najwyższą warstwę modelu. Podstawowe strony przejść są natomiast przeznaczone do reprezentowania aktywności systemu i tworzą drugą warstwę. Najniższą warstwę tworzą strony będące grafami synchronizacji, które są stosowane do przedstawiania części funkcjonalnej modelowanego systemu.

Tworzenie hierarchicznych RTCP-sieci jest ponadto ułatwione dzięki zastosowaniu *szablonów stron*, które grają podobną rolę jak na przykład wzorce klas lub funkcji w języku C++. Szablon strony może zawierać dowolną liczbę parametrów, z których każdy zastępuje pewien ciąg znaków. Wartości parametrów są określane podczas konstruowania grafu hierarchii stron. Jeżeli w sieci występują strony o taki samej strukturze, ale różniące się opisem elementów sieci (np. znakowaniem początkowym miejsc, wyrażeniami czasowymi łuków itp.), to można stworzyć jeden szablon i wielokrotnie z niego korzystać, budując model hierarchiczny.

Postać kanoniczna RTCP-sieci nakłada dodatkowe ograniczenia na ich strukturę. Zbiór miejsc jest podzielony na dwa rozłączne podzbiory *miejsc podstawowych* P_M i *miejsc pomocniczych* $P_A = P - P_M$. Miejsca podstawowe reprezentują wyróżnione części (np. podsystemy, obiekty) modelowanego systemu. Miejsca pomocnicze są stosowane w niższych warstwach modelu przy szczegółowym opisie aktywności przejść sieci. Analogicznie podzielono zbiór przejść na dwa rozłączne podzbiory *przejść podstawowych* T_M i *przejść pomocniczych* $T_A = T - T_M$. Przejścia podstawowe reprezentują aktywności modelowanego systemu. Przejścia pomocnicze są stosowane w niższych warstwach. Miejsca podstawowe mogą być połączone tylko z przejściami podstawowymi.

Zakładamy ponadto, że wyrażenia czasowe łuków, które prowadzą do lub od miejsc pomocniczych, są równe 0. Dla miejsc pomocniczych przyjmuje się, że ich początkowe pieczętki czasowe są niedodatnie. Podział na podstawowe i pomocnicze węzły sieci jest decyzją subiektywną konstruktora modelu. Podział ten wprowadzono między innymi po to, by analizę własności czasowych sieci ograniczyć do wyróżnionego zbioru miejsc.

Stosowanie RTCP-sieci do modelowania systemów czasu rzeczywistego nie wymaga zachowania jej postaci kanonicznej. Opracowano ją między innymi w celu automatycznego generowania sieci.

¹ Obiektami nazywamy dowolne wyróżnione fragmenty systemu.

13.2.1. Podstawowe strony miejsc

Omówimy teraz przykład tworzenia modelu systemu w postaci hierarchicznej RTCP-sieci dla abstrakcyjnego układu pomiarowego składającego się z dwóch procesorów i trzech sensorów. Sensory cyklicznie wykonują pomiary, które są danymi wejściowymi (częścią danych) dla obliczeń wykonywanych przez pierwszy procesor. Jeżeli obliczenia trwają zbyt długo, to wyniki pomiarów z kolejnego cyklu są przetwarzane przez drugi procesor. Wyniki obliczeń ukazują się na wyświetlaczu LCD.

Konstruowanie modelu rozpoczyna się od wyodrębnienia obiektów, z których jest zbudowany rozważany system, oraz podziału ich na elementy *aktywne* i *pasywne*. Podział ten rozróżnia elementy, z którymi będą powiązane pewne aktywności modelowanego systemu. W rozważanym przykładzie występuje pięć obiektów aktywnych (dwa procesory i trzy sensory) oraz jeden obiekt pasywny (wyświetlacz LCD). Dla każdego obiektu tworzy się listę atrybutów (dziedzinami atrybutów mogą być tylko proste typy danych), które go charakteryzują, i definiuje się odpowiednie typy danych. Wszystkie obiekty są reprezentowane przez podstawowe miejsca, przy czym obiekty aktywne są umieszczone na własnych stronach podstawowych. Typ miejsca reprezentującego dany obiekt jest definiowany jako iloczyn kartezjański typów jego atrybutów.

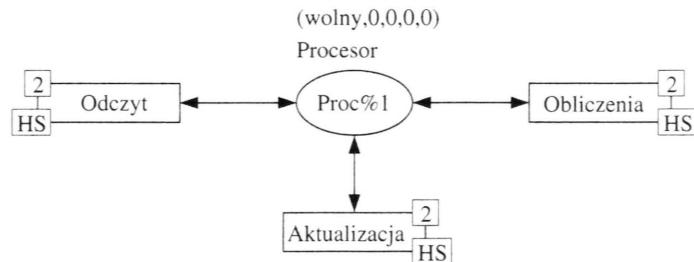
Podstawowe strony miejsc są konstruowane w postaci sieci składających się z dokładnie jednego miejsca, które reprezentuje modelowany obiekt, oraz kilku przejść – po jednym dla każdej aktywności przypisanej do tego obiektu. Każde przejście jest połączone z miejscem dwoma łukami tworzącymi ciasną pętlę.

W naszym przykładzie obiekt *Procesor* jest opisany przez 5 atrybutów:

- stan procesora (dopuszczalne stany: *wolny* – procesor jest gotowy do pobrania nowych danych, *dane* – stan po odczytaniu danych z sensorów, *wynik* – stan po wykonaniu obliczeń, których wynik powinien zostać wyświetlony);
- wynik obliczeń (zero, jeśli ich jeszcze nie wykonano);
- trzy wartości używane w obliczeniach (zera, jeśli jeszcze nie odczytano danych).

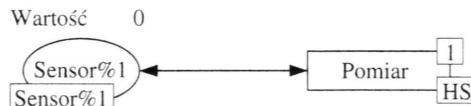
Dla powyższych atrybutów zdefiniowano typy:

```
color Stan      = with wolny | dane | wynik;
color Wynik     = int with 0..2;
color Wartość   = int with 0..9;
color Procesor = product Stan * Wynik * Wartość * Wartość * Wartość;
color Czas       = int with 1..4;
var n : Czas;
```



Rys. 13.3. Procesor – szablon stron

Na rysunku 13.3 przedstawiono podstawową stronę miejsca *Procesor*. W rzeczywistości jest to szablon stron z jednym parametrem (%1), który zastępuje numer procesora. Szablonu tego użyto do wygenerowania dwóch stron reprezentujących procesory. Każdy procesor może wykonywać trzy czynności: *Odczyt*, *Obliczenia*, *Aktualizacja*, które są reprezentowane przez podstawiane przejścia (należące do zbioru T_M) widoczne na rys. 13.3. Wszystkim przejściom przypisano taki sam priorytet równy 2. Wyrażenia dla łuków pominięto, wystarczy przedstawić je w niższej warstwie modelu.



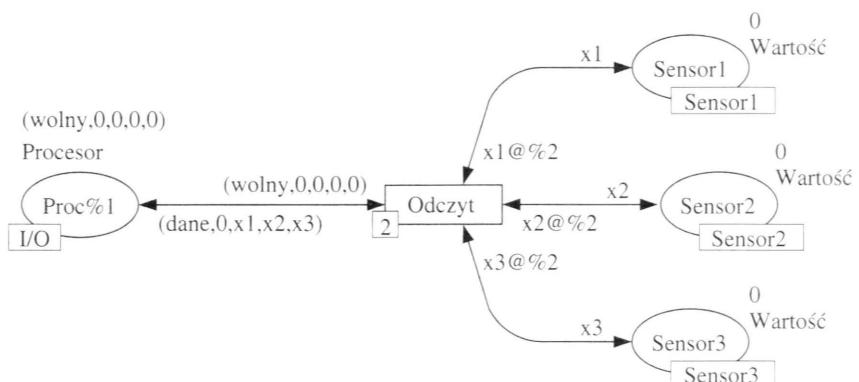
Rys. 13.4. Sensor – szablon stron

Na rysunku 13.4 przedstawiono szablon, którego użyto do generowania stron reprezentujących sensory. Jedyny parametr szablonu reprezentuje numer sensora (w efekcie końcowym otrzymamy różne nazwy miejsc i fuzji miejsc). Ponieważ w RTCP-sieciach występują tylko fuzje globalne, więc w opisie fuzji miejsc po-minięto etykietę reprezentującą jej typ. Każdy z sensorów może wykonać tylko jedną czynność – *Pomiar*, reprezentowaną przez odpowiednie przejście.

Obiekty pasywne (np. wyświetlacz LCD) są reprezentowane przez pojedyncze miejsca podstawowe lub fuzje miejsc, ale nie są umieszczane na własnych stronach podstawowych.

13.2.2. Podstawowe strony przejść

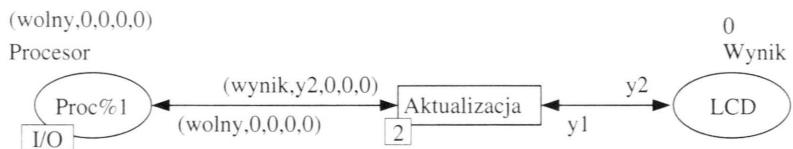
Przejścia umieszczane na podstawowych stronach miejsc są zazwyczaj przejściami podstawianymi. Dla każdego z podstawianych przejść tworzy się *podstawową stronę przejścia*. Strona taka zawiera dokładnie jedno podstawowe przejście reprezentujące daną aktywność oraz kilka podstawowych miejsc reprezentujących obiekty, których wartości atrybutów są potrzebne do wykonania tego przejścia. Jedno z tych miejsc jest miejscem portowym, skojarzonym z gniazdem znajdującym się na odpowiedniej nadstronie. Pozostałe miejsca są niezależnymi miejscami, jeżeli występują tylko na tej stronie, albo miejscami należącymi do fuzji miejsc, jeżeli występują na kilku stronach. Każde z tych miejsc jest połączone z przejściem dwoma łukami tworzącymi ciasną pętlę. Szablon, na którego podstawie można wygenerować stronę dla aktywności *Odczyt*, przedstawiono² na rys. 13.5.



Rys. 13.5. *Odczyt* – szablon stron

Szablon przedstawiony na rys. 13.5 zawiera dwa parametry. Pierwszy z nich zastępuje numer procesora, drugi – wyrażenie czasowe dla trzech łuków wejściowych przejścia *Odczyt*. Przyjmijmy, że wartość drugiego parametru (%2) jest równa 0 dla pierwszego procesora i 3 dla drugiego. Oznacza to, że pierwszy procesor może pobrać znaczniki z miejsc *Sensor1..3*, jeżeli tylko są one dostępne (tzn. wartości pieczętek czasowych są niedodatnie), a drugi musi czekać, aż znaczniki te osiągną wiek równy 3. Jeżeli zatem w ciągu 3 jednostek czasu od dokonania pomiarów pierwszy z procesorów nie pobierze tych znaczników, to będzie mógł to zrobić drugi z nich.

² Na kolejnych rysunkach występują zmienne, których deklaracji nie zamieszczono. Z kontekstu wynika, jakiego typu są te zmienne.



Rys. 13.6. Aktualizacja – podstawowa strona przejęcia

Na rysunku 13.6 przedstawiono podstawową stronę przejęcia *Aktualizacja*. Strona ta zawiera miejsce *LCD* reprezentujące wyświetlacz.

13.2.3. Warstwa funkcjonalna modelu

Zastosowanie wyłącznie podstawowej strony przejęcia może okazać się niewystarczające do opisania aktywności reprezentowanej przez dane podstawowe przejęcie. Przejście znajdujące się na podstawowej stronie przejęcia może mieć dołączoną kolejną podstronę. Z przejęciem takim możemy skojarzyć podsieć opisującą szczegóły jego wykonania.

Jednym z zastosowań RTCP-sieci jest modelowanie systemów wbudowanych, których działanie może być oparte na systemie regułowym. Zbiór reguł decyzyjnych może być przedstawiony za pomocą niehierarchicznej sieci kolorowanej. Przyjmijmy, że reguły opisujące zasady wykonania przejęcia *Obliczenia* mają taką postać, jaką przedstawiono w tab. 13.1.

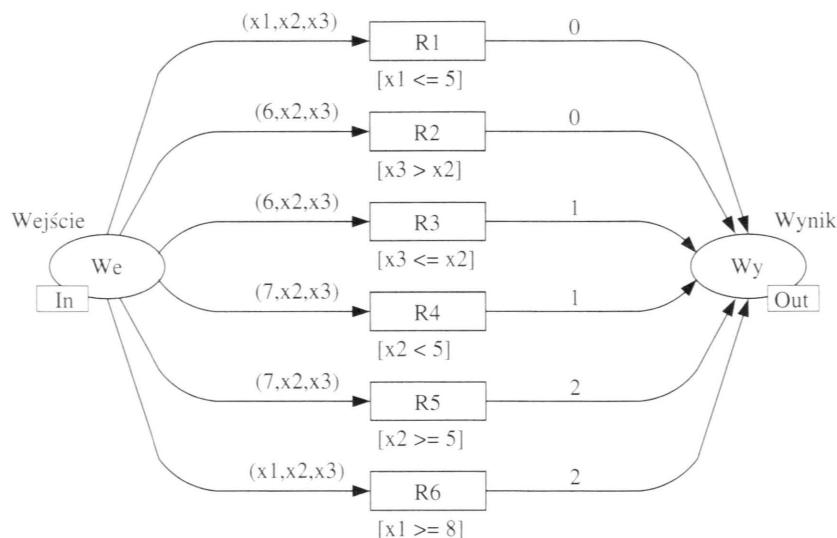
Tabela 13.1. Tablica decyzyjna dla przejęcia *Obliczenia*

	Sensor1	Sensor2	Sensor3	Wynik
R1	Sensor1 <= 5	Sensor2	Sensor3	0
R2	Sensor1 = 6	Sensor2	Sensor3 > Sensor2	0
R3	Sensor1 = 6	Sensor2	Sensor3 <= Sensor2	1
R4	Sensor1 = 7	Sensor2 < 5	Sensor3	1
R5	Sensor1 = 7	Sensor2 >= 5	Sensor3	2
R6	Sensor1 >= 8	Sensor2	Sensor3	2

Aby skonstruować tablicę decyzyjną, należy w pierwszej kolejności zdefiniować zbiór atrybutów, których wartości mogą być niezbędne do podjęcia decyzji lub które są powiązane z podejmowanymi decyzjami. Zbiór ten jest podzielony na dwa (niekoniecznie rozłączne) podzbiory: *atrybuty warunkowe* i *atrybuty decyzyjne*. Reguły decyzyjne mają postać implikacji. Atrybuty warunkowe występują

w poprzednikach implikacji, atrybuty decyzyjne – w następcach. Kolumny tablicy decyzyjnej są etykietowane nazwami atrybutów, wiersze zaś, numerami reguł. Każdy element tablicy może zawierać wyrażenie, którego dowolne wartościowanie musi być wartością logiczną dla atrybutów warunkowych, a dla atrybutów decyzyjnych, pojedynczą wartością z dziedziny danego atrybutu. Wiersze tablicy decyzyjnej nazywamy *regułami decyzyjnymi*³.

Nasza tablica decyzyjna zawiera trzy atrybuty warunkowe, jeden atrybut decyzyjny i sześć reguł decyzyjnych. Rozważmy na przykład regułę R_2 . Jej znaczenie jest następujące: Jeżeli wartość atrybutu Sensor1 jest równa 6, wartość atrybutu Sensor2 jest dowolna i wartość atrybutu Sensor3 jest większa niż wartość atrybutu Sensor2, to atrybut decyzyjny Wynik przyjmuje wartość 0.

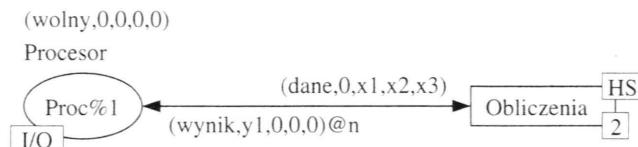


Rys. 13.7. Obliczenia – sieć decyzyjna

Sieć niehierarchiczną, która reprezentuje tablicę decyzyjną, nazywamy *siecią decyzyjną* lub krótko *D-siecią*. Przykład D-sieci, która reprezentuje rozważaną tablicę decyzyjną, przedstawiono na rys. 13.7. D-siec zawiera dwa miejsca – *warunkowe* i *decyzyjne* (w tym przypadku nazwane *We* i *Wy*), które są odpowiednio powiązane z atrybutami warunkowymi i decyzyjnymi. Znacznik umieszczony w miejscu warunkowym jest krotką reprezentującą wartości atrybutów warunkowych. Podobnie

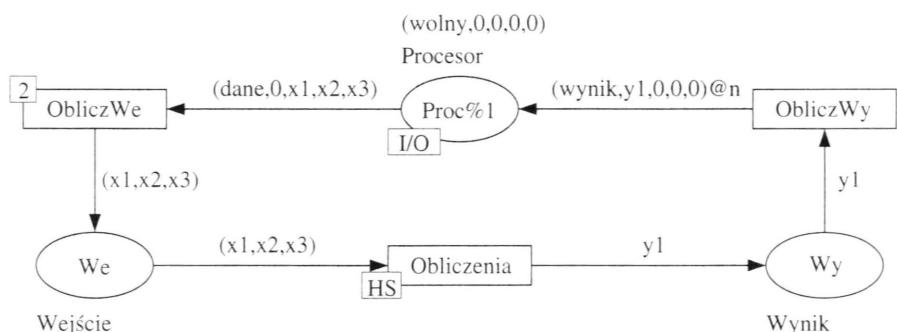
³ W rzeczywistości są to tzw. uogólnione reguły decyzyjne (zob. pracę [82]), gdyż atrybuty mogą przyjmować wartości nieatomowe.

znacznik umieszczony w miejscu decyzyjnym jest krotką reprezentującą wartości atrybutów decyzyjnych. Każda reguła decyzyjna jest reprezentowana przez jedno przejście i otaczające je łuki. Przejście takie ma jeden łuk wejściowy, prowadzący od miejsca warunkowego, i jeden łuk wyjściowy, prowadzący do miejsca decyzyjnego. Wszystkie węzły sieci decyzyjnej należą do zbiorów węzłów pomocniczych.



Rys. 13.8. *Obliczenia* – podstawowa strona przejścia

Na rysunku 13.8 przedstawiono podstawową stronę przejścia *Obliczenia*. Bezpośrednie połączenie rozważanej D-sieci z tą stroną nie jest możliwe. Konieczne jest w tym celu przygotowanie pośredniczącej podstrony, która *zbierze* odpowiednie informacje dla sieci decyzyjnej oraz *rozdysytrubuje* wyniki. Sieć taką będziemy nazywać *podsięcią łączącą* lub krótko *łącznikiem*.



Rys. 13.9. *Obliczenia* – łącznik

Łącznik dla przejścia *Obliczenia* przedstawiono na rys. 13.9. Ma on bardzo prostą strukturę, ale zazwyczaj może być zdecydowanie bardziej złożony. Łączniki są tym typem stron sieci, których konstruktorom pozostawiono największą swobodę. Oprócz funkcji *doklejania* D-sieci do modelu, mogą one reprezentować dowolne algorytmy. Jednakże łącznik jest siecią, która musi spełniać pewne ograniczenia. Strona ta musi zawierać miejsca portowe dla wszystkich gniazd z odpowiadającej jej nadstrony (w rozważanym przypadku występuje tylko jedno miejsce *Proc%1*) oraz dwa przejścia (w tym przypadku *ObliczWe* i *ObliczWy*). Wszystkie te ele-

menty sieci należą do jej węzłów podstawowych, pozostałe węzły występujące w łączniku są zaliczane do węzłów pomocniczych. Dwa podstawowe przejścia stanowią swoisty *podziały* przejścia z nadstrony. Jedno z nich jest jedynym przejściem wyjściowym wszystkich portów, a drugie – jedynym przejściem wejściowym. Przejścia te nazywamy odpowiednio *przejściem wejściowym* i *wyjściowym* łącznika.

Dowolna aktywność łącznika musi być zapoczątkowana wykonaniem przejścia wejściowego i musi się kończyć wykonaniem przejścia wyjściowego. Podczas pracy RTCP-sieci przejście do łącznika przypomina wywołanie funkcji w językach programowania. Struktura łącznika ma zagwarantować przekazanie zawsze dokładnie jednej wartości (jednego zestawu wartości) będącej *wynikiem* wywołania takiej funkcji. Innymi słowy, dopóki nie zostanie wykonane przejście wejściowe, dopóty żadne inne przejście łącznika nie jest aktywne. Po wykonaniu przejścia wejściowego następuje ciąg wykonań przejścia łącznika (innych niż przejście wejściowe i wyjściowe), a po nim, wykonanie przejścia wyjściowego.

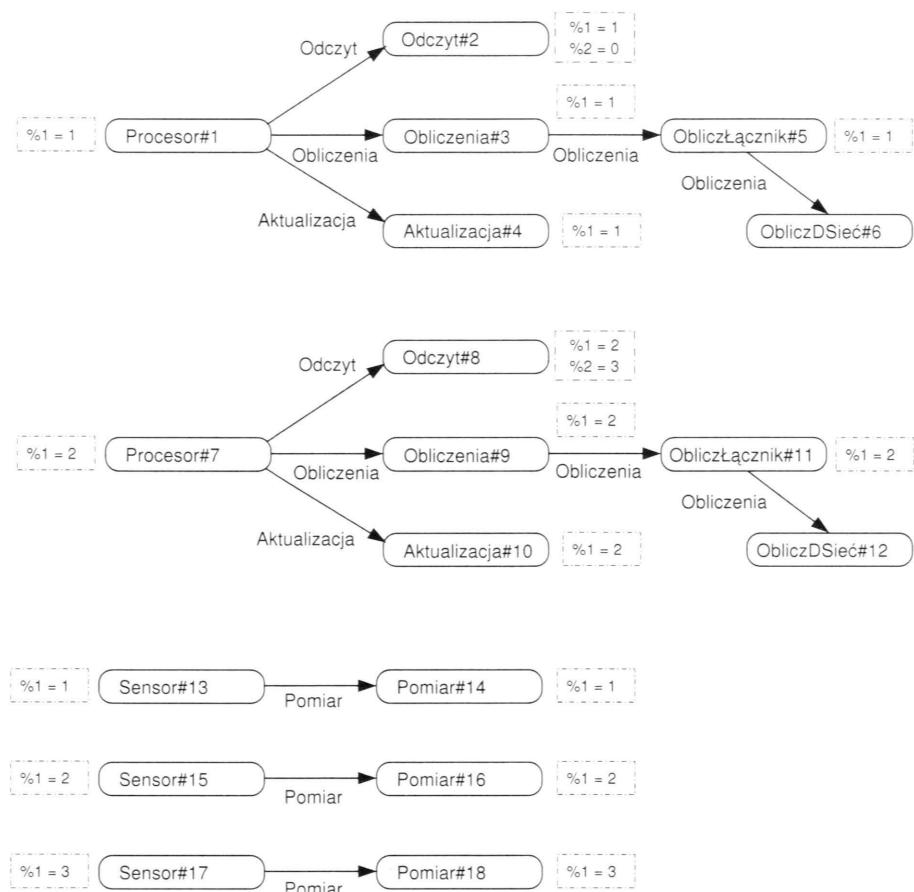
Przedstawiony na rys. 13.9 łącznik służy jedynie do doklejenia D-sieci do modelu. W ogólnym przypadku łącznik może zawierać dowolne podsieci, ale z tym zastrzeżeniem, by po przekształceniu go w sieć niehierarchiczną wykazywał opisane powyżej własności. Biorąc pod uwagę fakt, że wszystkie węzły umieszczone między przejściami wejściowym i wyjściowym są węzłami pomocniczymi, czas upływający między wykonaniem przejścia wejściowego i wyjściowego wynosi zero. Dla analizy czasowej RTCP-sieci nie ma zatem znaczenia fakt, czy strona podstawowa przejścia ma podstronę będącą łącznikiem czy też jej nie ma⁴.

13.2.4. Graf hierarchii stron

Skonstruowanie wszystkich stron i szablonów kończy pierwszy etap projektowania RTCP-sieci. Drugim etapem jest zbudowanie hierarchicznego modelu na podstawie przygotowanych komponentów. Podobnie jak w hierarchicznych sieciach kolorowych, zależności wynikające ze stosowania podstawianych przejść są przedstawiane za pomocą grafu hierarchii stron. Każda strona modelu jest reprezentowana za pomocą jednego węzła grafu hierarchii. Łuki grafu, które są etykietowane nazwami podstawianych przejść, wskazują zależności nadstrona-podstrona. Graf hierarchii stron dla rozważanego przykładu przedstawiono na rys. 13.10. Jeżeli strona sieci

⁴ Możemy tu zauważać pewną analogię do krokowego śledzenia wykonania programu. Napotykając wywołanie funkcji, niekoniecznie musimy śledzić krokowo instrukcje znajdujące się w jej wnętrzu.

jest generowana na podstawie szablonu, to obok reprezentującego ją węzła umieszcza się informacje⁵ o wartościach parametrów.



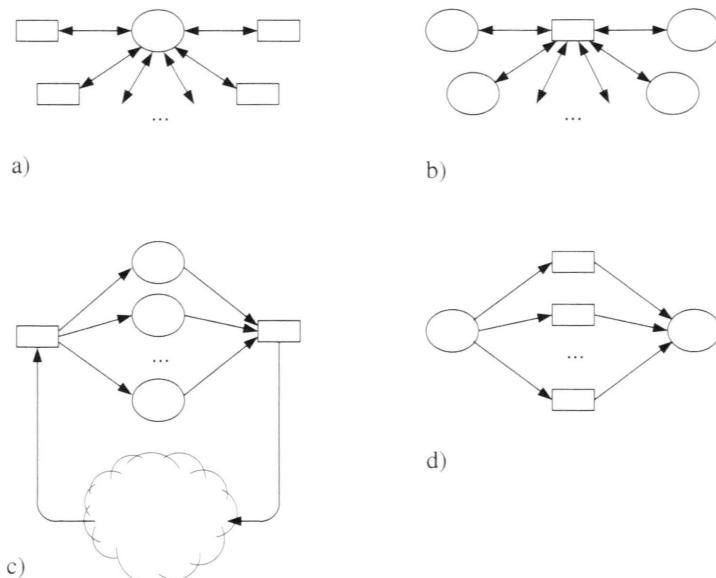
Rys. 13.10. Graf hierarchii stron dla systemu pomiarowego

Zmiana wartości parametrów w grafie hierarchii stron może mieć znaczący wpływ na własności sieci. Korzystanie z szablonów stron pozwala na bardzo łatwe eksperymentowanie z różnymi wersjami modelu, które powstają wyłącznie w wy-

⁵ W opisie przykładu nie zamieszczono strony podstawowej dla przejścia *Pomiar*, gdyż nie było to konieczne przy opisie hierarchicznym RTCP-sieci. Na potrzeby symulacji można zbudować stronę o strukturze takiej jak na rys. 13.8 (przy czym przejście nie będzie przejściem podstawianym), a nowe wartości sensorów – wybierać losowo.

niku zmian wartości parametrów. Ponadto, łatwo jest również reorganizacja hierarchicznej struktury modelu.

Na rysunku 13.11 przedstawiono uproszczoną strukturę czterech typów stron, które występują w RTCP-sieciach w postaci kanonicznej.

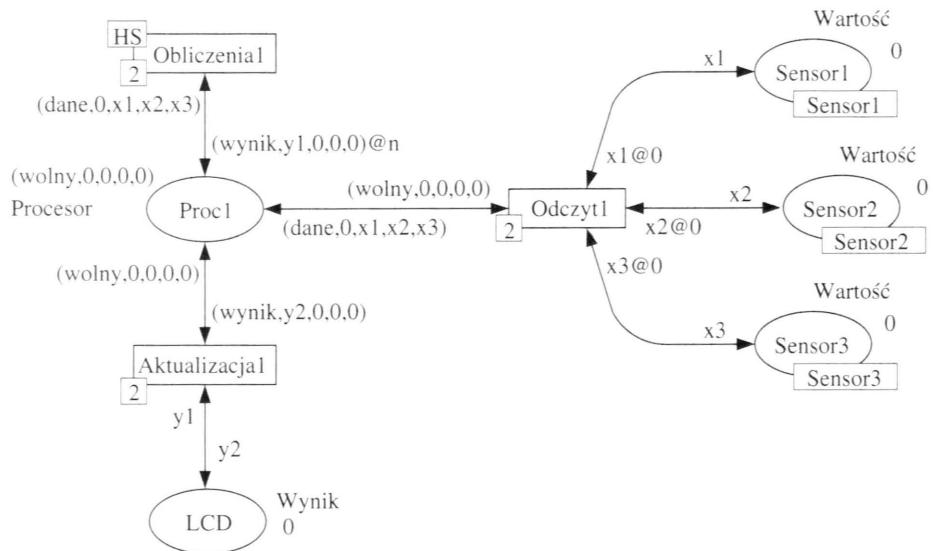


Rys. 13.11. Uproszczona struktura stron RTCP-sieci w postaci kanonicznej: a) podstawowa strona miejsca, b) podstawowa strona przejścia, c) łącznik, d) D-sieć

RTCP-sieci w postaci kanonicznej można łatwo nadać formę sieci niehierarchicznej. W pierwszej kolejności *Scalane* są ze sobą strony podstawowe. Pierwsze dwie warstwy modelu łączy się w następujący sposób:

- (1) Występujące na wszystkich stronach sieci parametry zastępujemy ich wartościami określonymi w grafie hierarchii stron. Każdą stronę wygenerowaną z szablonu opracowujemy indywidualnie.
- (2) Każdą podstawową stronę miejsca scalamy ze stronami podstawowymi występującymi na niej przejść. Przy scaleniu miejsce gniazdowe łączy się z odpowiednim portem, a przejścia występujące na podstawowej stronie miejsca są scalane z odpowiedającymi im przejściami z podstawowych stron przejść.
- (3) Sieci otrzymane w wyniku wykonania kroku (2) łączymy ze sobą, scalając miejsca należące do tych samych fuzji miejsc.

- (4) Gdy jest to konieczne, wówczas modyfikujemy nazwy węzłów, dodając im także indeksy, aby można było uniknąć niejednoznacznych nazw.



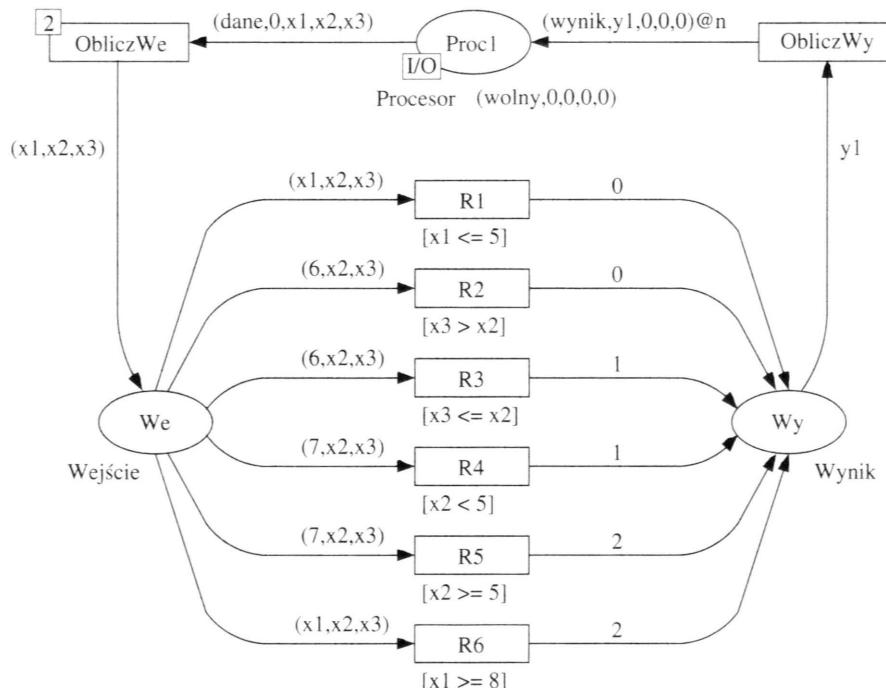
Rys. 13.12. Strona *Procesor* po scaleniu z podstronami

Na rysunku 13.12 przedstawiono stronę *Procesor* (dla pierwszego procesora) po scaleniu z podstawowymi podstronami przejść. W sieci niehierarchicznej znajdują się dwie takie podsieci dla obu procesorów oraz trzy podsieci dla sensorów. Podsieci te scalono przez połączenie ze sobą miejsc tworzących fuzje.

Jeżeli sieć nie zawiera stron łączących, to opisany powyżej etap kończy transformację modelu do postaci niehierarchicznej. Jeżeli w modelu występują łączniki (i ewentualnie sieci decyzyjne), to po pierwszym etapie transformacji otrzymujemy sieć hierarchiczną, która składa się z jednej strony podstawowej i zbioru podstron będących łącznikami lub D-sieciami.

Scalanie strony będącej łącznikiem z jej nadstroną polega na usunięciu z nadstroń podstawianego przejścia i połączeniu ze sobą par (gniazdo, port). Analogicznie przebiega scalanie D-sieci z łącznikiem. W wyniku scalania łącznika z nadsiecią jedno z podstawowych przejść zostaje usunięte. W rzeczywistości sytuację tę można traktować jako *podział* tego przejścia na dwa przejścia: *wejściowe* i *wyjściowe*. Ponadto w trakcie symulacji pracy sieci czas upływający między wykonywaniem tych dwóch przejść jest równy 0. W ostatnim kroku, jeżeli jest to konieczne, mo-

dyfikujemy nazwy węzłów sieci, dodając im indeksy, aby można było uniknąć niejednoznacznych nazw. Otrzymaną w ten sposób sieć będziemy nazywać *równoważną postacią niehierarchiczną* RTCP-sieci. Na rysunku 13.13 przedstawiono fragment modelu będący wynikiem scalenia D-sieci (rys. 13.7) z siecią łączącą (rys. 13.9).



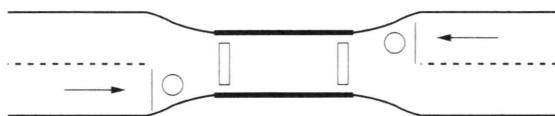
Rys. 13.13. Efekt scalenia D-sieci z siecią łączącą

13.3. Modelowanie z zastosowaniem RTCP-sieci

W tym podrozdziale omówimy dwa przykłady wykorzystania RTCP-sieci do modelowania systemów współbieżnych z czasem. Najpierw zajmiemy się modelowaniem sterowania ruchem wahadłowym pojazdów na moście. Drugi przykład to zmodyfikowany model systemu sterowania rogatkami na przejeździe kolejowym (zob. p. 7.2.1).

13.3.1. Model systemu sterowania ruchem wahadłowym

Rozważmy system sterowania ruchem pojazdów na moście, na którym obowiązuje ruch wahadłowy. Z obu stron mostu zainstalowano światła drogowe. Sygnał zielony zostaje zapalony na określony czas, po czym zmienia się na czerwony. Z drugiej strony mostu zmiana sygnału z czerwonego na zielony odbywa się z opóźnieniem spowodowanym oczekiwaniem, aż wszystkie pojazdy, które jeszcze znajdują się na moście, go opuszczą.



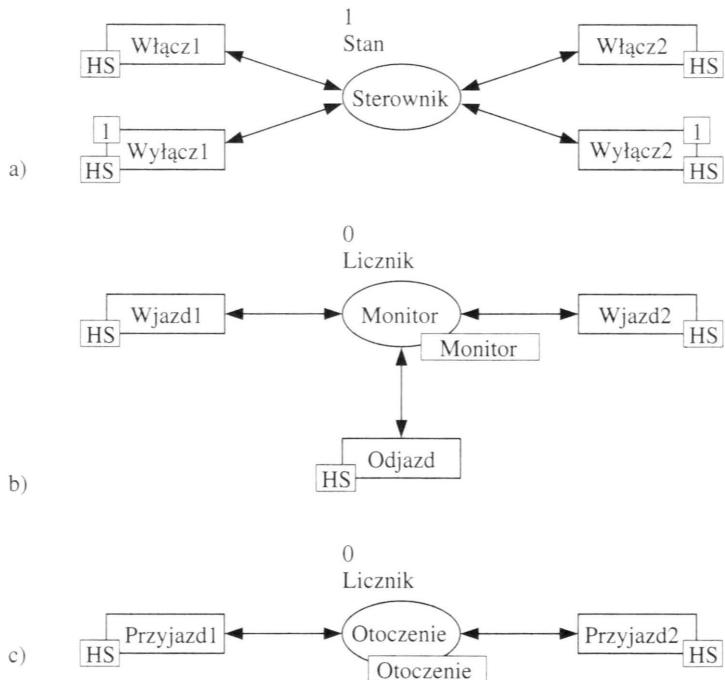
Rys. 13.14. Schemat przejazdu przez most

Schemat przejazdu przez most przedstawiono na rys. 13.14. Sygnalizatory umieszczone przy drogach dojazdowych do mostu są reprezentowane przez okręgi. Zakładamy ponadto, że przy moście umieszczono czujniki (reprezentowane przez dwa prostokąty), które zliczają pojazdy przejeżdżające przez most. Pozwala to zbadać, czy w danej chwili most jest pusty.

W omawianym modelu wyróżniono trzy obiekty aktywne: *Sterownik*, *Monitor* i *Otoczenie* oraz pięć obiektów pasywnych: *Droga1*, *Droga2*, *Światła1*, *Światła2* i *Zegar*.

Na rysunku 13.15 przedstawiono podstawowe strony miejsc dla rozważanego modelu. Obiekt *Sterownik* jest scharakteryzowany przez jeden atrybut określający jego stan. Wartość tego atrybutu zmienia się cyklicznie w zakresie od 1 do 4. Stan sterownika decyduje o kolejnej wykonywanej przez niego operacji. Cyklicznie wykonywane są operacje: *Włącz1*, *Wyłącz1*, *Włącz2* i *Wyłącz2*, które włączają i wyłączają sygnał zielony naprzemiennie dla obu dróg dojazdowych.

Obiekt *Monitor* jest używany do testowania obecności pojazdów na moście. Jedynek atrybut przyjmujący wartości całkowite oznacza liczbę pojazdów znajdujących się na moście. Stan monitora zmienia się w wyniku wykonywania operacji: *Wjazd1*, *Wjazd2* i *Odjazd*. W celu ograniczenia liczby pojazdów znajdujących się w systemie wprowadzono obiekt *Otoczenie*. Skojarzone z nim operacje (*Przyjazd1* i *Przyjazd2*) służą do umieszczania na drogach dojazdowych nowych pojazdów.



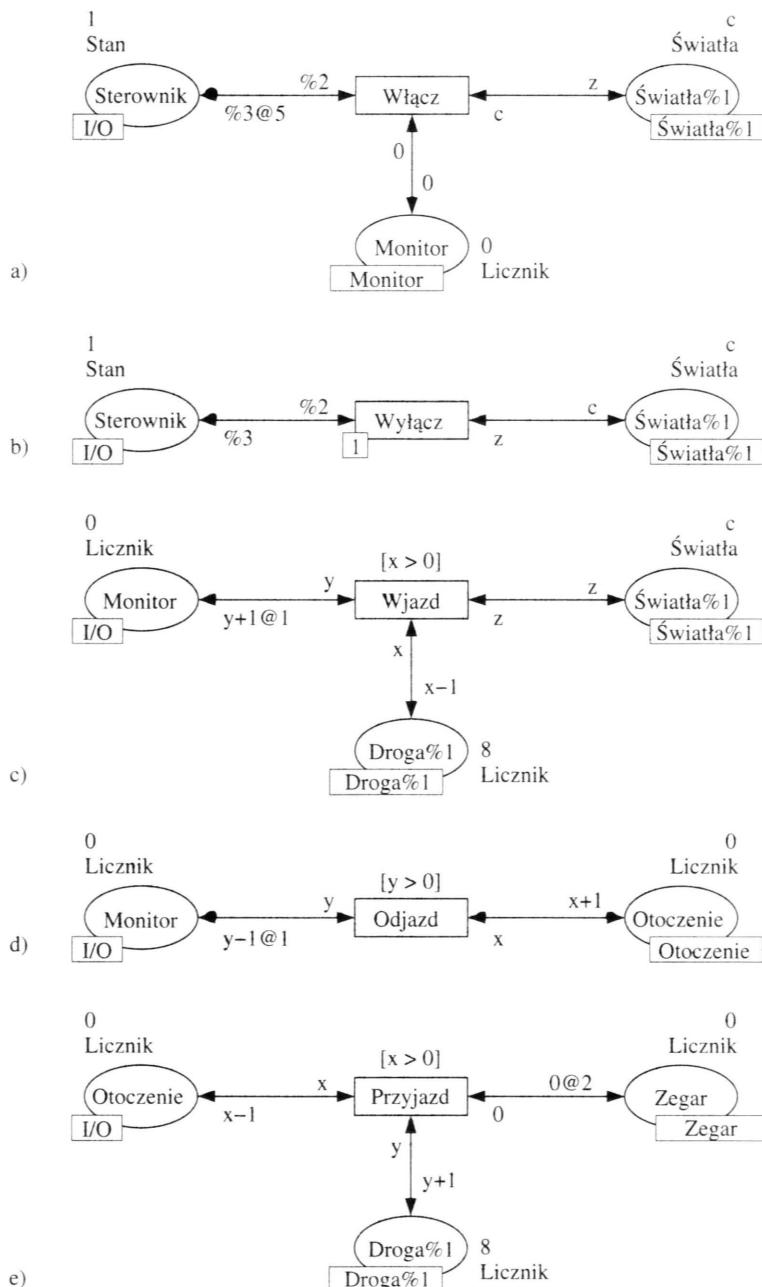
Rys. 13.15. Podstawowe strony miejsc dla modelu sterowania ruchem

W modelu występują następujące typy danych i zmienne:

```
color Stan    = int with 1..4;
color Licznik = int with 0..20;
color Światła = with c | z;
var x, y : Licznik;
```

Na rysunku 13.16 przedstawiono szablony podstawowych stron przejść dla rozważanego modelu. Oto ich krótki opis:

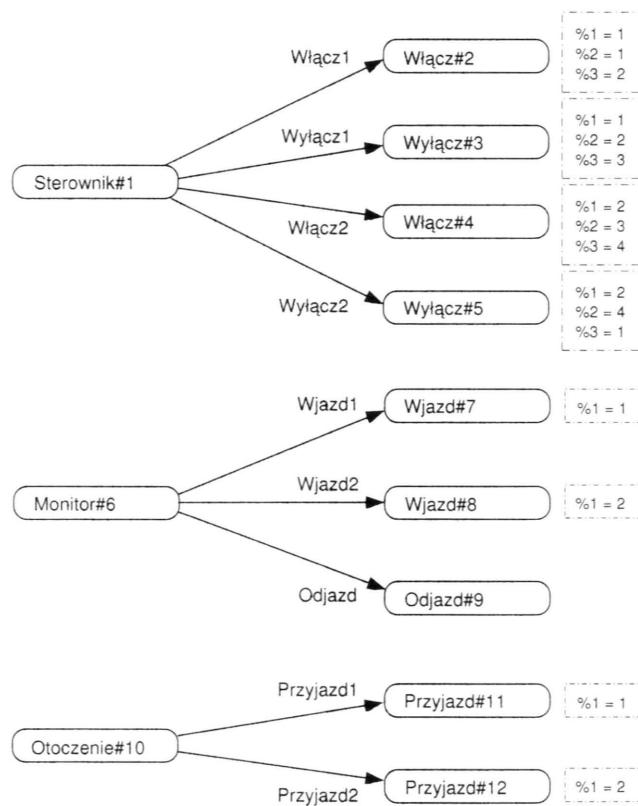
- Włączenie sygnału zielonego jest możliwe, o ile most jest pusty, czyli wartość znacznika w miejscu *Monitor* jest równa 0. Parametr %1 oznacza numer sygnalizatora, a pozostałe parametry – stan sterownika, przy którym wykonywane jest przejście, i stan przyjmowany przez sterownika po wykonaniu przejścia. Sygnał zielony jest zapalany na 5 jednostek czasu.
- Przejście *Wyłącz* ma przypisany najwyższy priorytet w modelu. To gwarantuje, że po 5 jednostkach czasu sygnał zielony będzie wyłączony, ale nie oznacza,



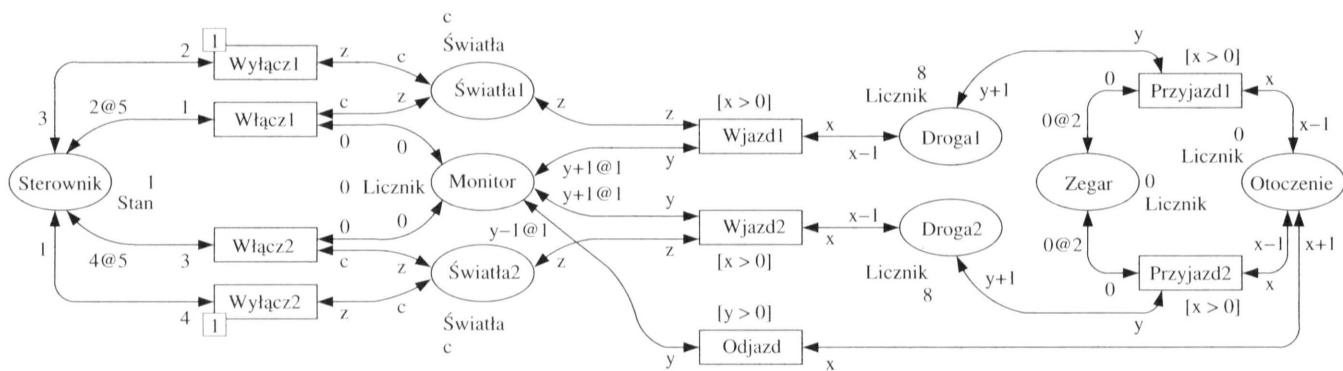
Rys. 13.16. Podstawowe strony przejść dla modelu sterowania ruchem

że natychmiast zostanie zapalony sygnał zielony po drugiej stronie. Parametry występujące w tym szablonie mają takie samo znaczenie jak przedstawione na rys. 13.16 a).

- c) Wykonanie przejścia *Wjazd* oznacza wjazd pojazdu na most. Fakt ten jest odnotowywany przez zwiększenie wartości znacznika w miejscu *Monitor*. Wykonanie przejścia jest możliwe, o ile na danej drodze znajdują się pojazdy i mają zielone światło.
- d) Strona przedstawiona na rys. 13.16 d) nie jest szablonem. Wykonanie przejścia *Odjazd* oznacza zjazd pojazdu z mostu. Fakt ten jest odnotowywany przez zmniejszenie wartości znacznika w miejscu *Monitor*.
- e) Wykonanie przejścia *Przyjazd* oznacza pojawienie się nowego pojazdu na drodze dojazdowej do mostu. Jest to możliwe, o ile znacznik w miejscu *Otoczenie* ma wartość większą niż zero. Dodatkowe miejsce *Zegar* służy do sterowania częstością pojawiania się nowych pojazdów.



Rys. 13.17. Graf hierarchii stron dla modelu sterowania ruchem wahadłowym



Rys. 13.18. Model sterowania ruchem wahadłowym – sieć niehierarchiczna

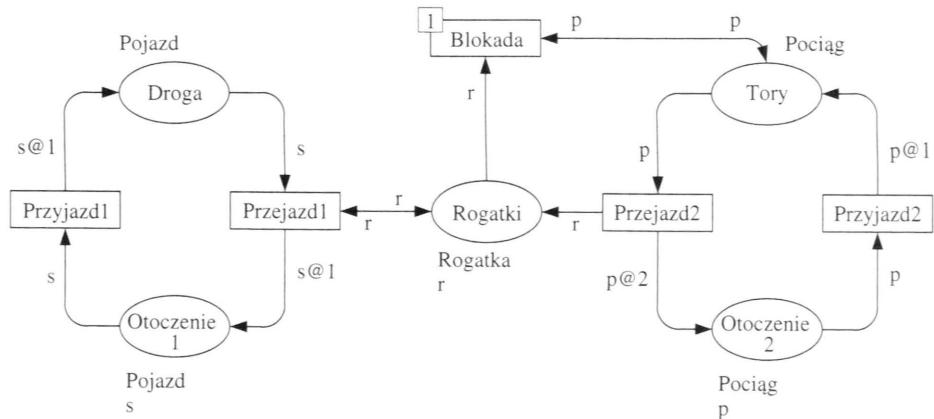
Na rysunku 13.17 przedstawiono graf hierarchii stron dla modelu sterowania ruchem wahadłowym. Rozważana sieć jest bezpieczna (wszystkie miejsca są 1-ograniczone), żywa, ale nie jest sprawiedliwa. Bezpieczeństwo wynika bezpośrednio ze struktury sieci. Każde miejsce sieci zawiera dokładnie jeden znacznik w stanie początkowym. Ponadto, jeżeli dwa węzły są połączone, to tworzą ciasną pętlę, a zatem wykonywanie dowolnego przejścia powoduje jedynie zmianę znaczników w jego miejscach wejściowych i wyjściowych, ale nie zmienia ich liczby.

Analizując znakowania sieci, dość łatwo można pokazać, że jest ona żywa. Brak sprawiedliwości wynika z faktu, że pojazdy z otoczenia mogą być przesyłane tylko na jedną z dróg dojazdowych; można zatem wskazać nieskończony ciąg przejść, w którym pewne przejścia nie występują.

Na rysunku 13.18 przedstawiono rozważany model w formie niehierarchicznej. Jest to postać, którą można otrzymać po scaleniu wszystkich stron sieci hierarchicznej.

13.3.2. Model przejazdu kolejowego

Rozważmy ponownie system sterowania rogatkami na przejeździe kolejowym, który był omawiany w p. 7.2.1. Schemat przejazdu przedstawiono na rys. 7.2, a model w postaci sieci priorytetowej na rys. 7.3. Zastosowanie RTCP-sieci pozwala dodatkowo na modelowanie zależności czasowych w takim systemie.



Rys. 13.19. Model sterowania rogatkami na przejeździe kolejowym

Modelowanie z zastosowaniem RTCP-sieci nie wymaga zachowania postaci kanonicznej, z której świadomie zrezygnowano w prezentowanym przykładzie. Model

przejazdu kolejowego w postaci RTCP-sieci przedstawiono na rys. 13.19. Struktura sieci jest izomorficzna ze strukturą sieci priorytetowej pokazanej na rys. 7.3.

Wszystkie typy występujące w tym modelu są typami wyliczeniowymi, zawierającymi pojedyncze elementy. Znaczenie poszczególnych elementów sieci jest analogiczne do opisanego w p. 7.2.1. Przedstawiona sieć jest ograniczona, żywa i sprawiedliwa (sieć priorytetowa nie była sprawiedliwa). Więcej informacji na temat analizy RTCP-sieci znajduje się w rozdz. 14.

13.4. Podsumowanie

W tym rozdziale omówiliśmy podstawowe pojęcia dotyczące RTCP-sieci stanowiących podklaś sieci kolorowanych, którą zdefiniowano na potrzeby modelowania systemów czasu rzeczywistego. Dołączenie funkcji priorytetów do definicji RTCP-sieci oraz wprowadzenie nowego modelu czasu pozwala łatwo modelować typowe pojęcia występujące w programowaniu współbieżnym, na przykład priorytety zadań, zjawisko przeterminowania itp.

W RTCP-sieciach zmieniono również podejście do konstruowania modeli hierarchicznych. Wprawdzie korzysta się w nich z mechanizmów zaczerpniętych z sieci kolorowanych (takich jak podstawiane przejścia i fuzje miejsc), ale wprowadzono pewne ograniczenia w zakresie ich stosowania. Jednocześnie wykorzystanie postaci kanonicznej pozwala na częściowo automatyczne generowanie modeli.

Analiza RTCP-sieci

Większość metod stosowanych do analizy sieci kolorowanych może posłużyć również do analizy RTCP-sieci. Jeżeli pominiemy aspekty czasowe RTCP-sieci, to otrzymamy w efekcie sieć kolorowaną (ewentualnie z priorytetami), której własności (np. dotyczące ograniczoności miejsc) można badać, budując graf osiągalności tak jak w podrozdz. 11.1 lub wyznaczając niezmienikie sieci.

Bardzo ważną cechą RTCP-sieci jest możliwość zbudowania dla wielu z nich skończonego grafu osiągalności lub grafu pokrycia, który pozwala analizować zależności czasowe, na przykład wyznaczyć minimalny i maksymalny czas wykonania ciągu przejścia czy też minimalny i maksymalny czas przejścia między dwoma zadanymi stanami sieci. Własność ta jest jedną z głównych cech odróżniających RTCP-sieci od czasowych sieci kolorowanych. W tym rozdziale omówimy grafy osiągalności dla RTCP-sieci, koncepcję pokrywania się stanów i grafy pokrycia.

14.1. Grafy osiągalności

Węzły grafu osiągalności RTCP-sieci reprezentują stany będące wynikiem wykonania przejścia. Te zmiany stanów, które spowodował upływ czasu, nie są jawnie przedstawiane za pomocą oddzielnych łuków. Etykiety łuków grafu są parami, w których pierwszym elementem jest para (przejście, wiązanie), drugim zaś liczba wymienna oznaczająca upływ czasu między stanem reprezentowanym przez węzeł początkowy łuku a chwilą wykonywania danego przejścia.

Niech dana będzie RTCP-sieć $\mathcal{N} = (\Sigma, P, T, A, C, G, I, E_M, E_S, M_0, S_0)$.

Definicja 14.1. *Grafem osiągalności sieci \mathcal{N} nazywamy graf $\mathcal{G} = (V, A, \gamma)$ etykietowany nad zbiorem $\mathcal{B} \times \mathbb{Q}_+$, spełniający warunki:*

- (1) $V \subseteq \mathcal{R}(M_0, S_0)$ jest zbiorem tych stanów sieci, które są wynikiem wykonania przejścia, tzn. stanów, dla których istnieją pary $(M', S') \in \mathcal{R}(M_0, S_0)$ oraz

$(b, t) \in \mathcal{B}$, takie że $(M', S') \xrightarrow{(t,b)} (M, S)$. Ponadto, zbiór V zawiera stan początkowy (M_0, S_0) .

(2) A jest zbiorem łuków grafu \mathcal{G} postaci $((M, S), ((t, b), \tau), (M', S'))$, przy czym $(M, S), (M', S') \in V$ oraz

- $(M, S) \xrightarrow{(t,b)} (M', S')$, jeżeli $\tau = 0$,
- lub istnieje stan $(M, S'') \in \mathcal{R}(M_0, S_0)$, taki że $(M, S) \xrightarrow{\tau} (M, S'')$ i $(M, S'') \xrightarrow{(t,b)} (M', S')$, jeżeli $\tau > 0$.

(3) $\forall a = ((M, S), ((t, b), \tau), (M', S')) \in A: \gamma(a) = ((M, S), (M', S'))$. ■

Rozważmy model sterowania rogatkami na przejeździe kolejowym, przedstawiony na rys. 13.19. Przyjmijmy, że zbiór miejsc sieci jest uporządkowany następująco: $P = \{Droga, Otoczenie1, Tory, Otoczenie2, Rogatki\}$. Stan początkowy (M_0, S_0) :

$$\begin{aligned} M_0 &= (-, s, -, p, r), \\ S_0 &= (0, 0, 0, 0, 0) \end{aligned} \quad (14.1)$$

stanowi pierwszy węzeł grafu osiągalności. (W zapisie znakowań sieci używa się symbolu – do oznaczenia zbioru pustego). W stanie początkowym aktywne są przejścia: *Przyjazd1* i *Przyjazd2*¹. Wykonanie każdego z tych przejść prowadzi do nowego stanu, który będzie reprezentowany przez odpowiedni węzeł w grafie osiągalności. Wykonanie przejścia *Przyjazd1* prowadzi do stanu (M_1, S_1) :

$$\begin{aligned} M_1 &= (s, -, -, p, r), \\ S_1 &= (1, 0, 0, 0, 0), \end{aligned} \quad (14.2)$$

a wykonanie przejścia *Przyjazd2*, do stanu (M_2, S_2) :

$$\begin{aligned} M_2 &= (-, s, p, -, r), \\ S_2 &= (0, 0, 1, 0, 0). \end{aligned} \quad (14.3)$$

Nowe węzły będą połączone łukami z węzłem początkowym o etykietach odpowiednio (*Przyjazd1, 0*) i (*Przyjazd2, 0*).

W stanie (M_1, S_1) aktywne jest przejście *Przyjazd2*, a jego wykonanie prowadzi do stanu (M_3, S_3) :

$$\begin{aligned} M_3 &= (s, -, p, -, r), \\ S_3 &= (1, 0, 1, 0, 0). \end{aligned} \quad (14.4)$$

¹ Informację o wiązaniach pomijamy, gdyż wszystkie możliwe wiązania mają puste dziedziny.

W nowym stanie nie jest aktywne żadne przejście, ale możliwy jest upływ czasu $\tau = 1$, który prowadzi do stanu (M_3, S'_3) , przy czym:

$$S'_3 = (0, -1, 0, -1, -1). \quad (14.5)$$

Stanu (M_3, S'_3) nie dodaje się do grafu osiągalności. W tym stanie aktywne jest przejście *Blokada*. Stan (M_4, S_4) będący wynikiem wykonania przejścia *Blokada* zostanie dodany do grafu osiągalności i będzie połączony łukiem o etykiecie $(\text{Blokada}, 1)$ z węzłem (M_3, S_3) . Pełny graf osiągalności dla rozważanej sieci przedstawiono na rys. 14.1.

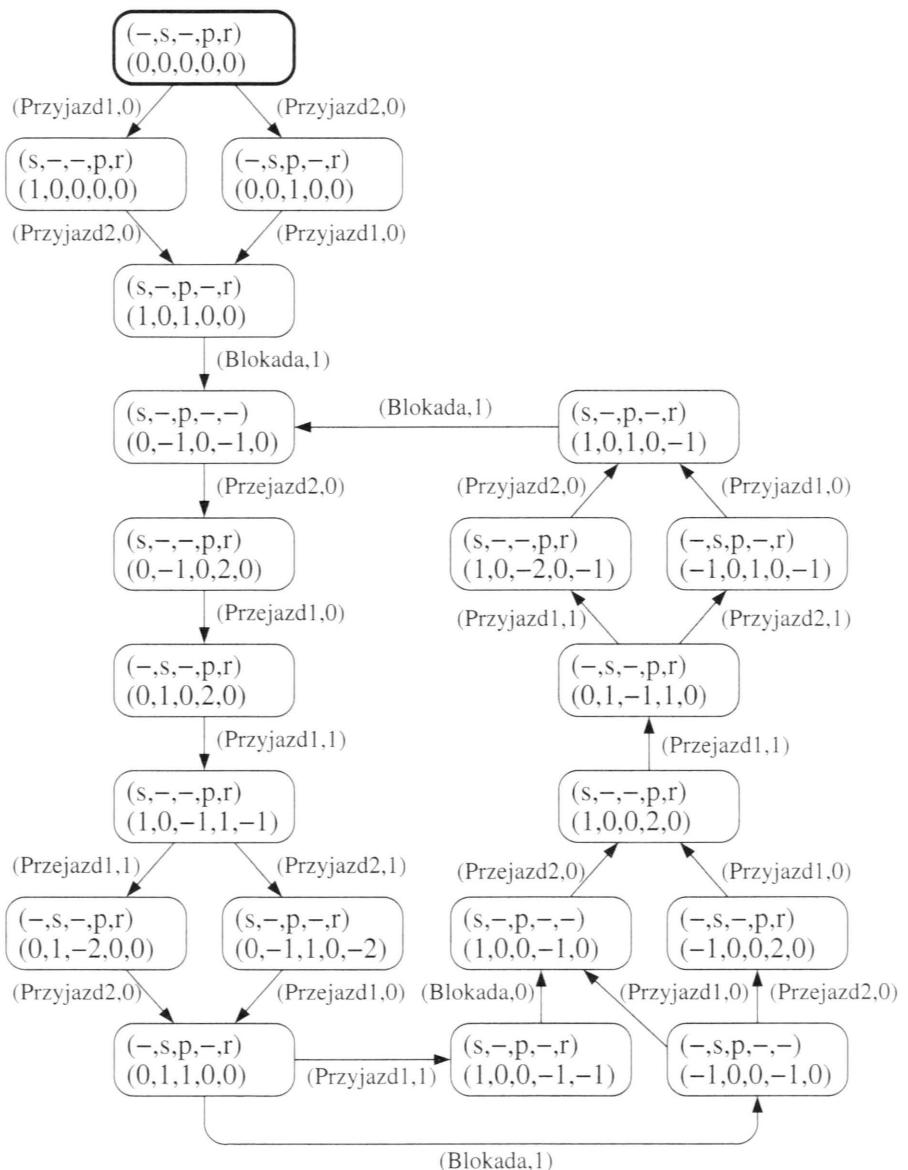
Pomijając własności czasowe, grafy osiągalności dla RTCP-sieci dają takie same możliwości analizowania sieci, jak grafy dla sieci kolorowanych. Można dla nich sformułować twierdzenia analogiczne do twierdzeń 11.1 i 11.2.

Na podstawie grafu osiągalności na rys. 14.1 i wspomnianych powyżej twierdzeń można wykazać, że sieć \mathcal{N} na rys. 13.19 ma następujące własności: jest bezpieczna, jej górnymi wielozbiorowymi ograniczeniami miejsc są odpowiednio zbiory: s, s, p, p i r , dolnym zaś wielozbiorowym ograniczeniem wszystkich miejsc jest zbiór pusty. Sieć \mathcal{N} jest żywa, a biorąc pod uwagę fakt, że występują w niej tylko wiązania o pustych dziedzinach, jest ona silnie żywa. Sieć jest ponadto sprawiedliwa, bo każdy cykl w grafie osiągalności zawiera wszystkie wiązania sieci.

Drogi w grafie osiągalności odpowiadają wykonywanym ciągom przejść. Na podstawie etykiet łuków grafu osiągalności (a dokładniej mówiąc, drugich elementów par stanowiących etykiety) można wyznaczać czas wykonania ciągu przejść czy też czas przejścia od stanu (M, S) do stanu $(M', S') \in \mathcal{R}(M, S)$. Wyznaczanie najkrótszego i najdłuższego czasu przejścia między dwoma stanami sieci sprowadza się więc do zastosowania standardowych algorytmów przeglądania grafów i poszukiwania najkrótszej i najdłuższej ważonej drogi łączącej dwa węzły grafu (jako wagi łuków przyjmujemy drugie elementy par stanowiących etykiety łuków).

Dla rozważanej sieci wyznaczmy minimalny i maksymalny czas, który upływa między kolejnymi przejazdami pociągu przez przejazd kolejowy (wykonanie przejścia *Przejazd2*). Graf osiągalności sieci \mathcal{N} zawiera trzy stany, przy których aktywne jest przejście *Przejazd2*:

- (M_1, S_1) : $M_1 = (s, -, p, -, -)$, $S_1 = (0, -1, 0, -1, 0)$,
- (M_2, S_2) : $M_2 = (-, s, p, -, -)$, $S_2 = (-1, 0, 0, -1, 0)$,
- (M_3, S_3) : $M_3 = (s, -, p, -, -)$, $S_3 = (1, 0, 0, -1, 0)$.



Rys. 14.1. Graf osiągalności dla RTCP-sieci modelującej sterowanie rogatkami

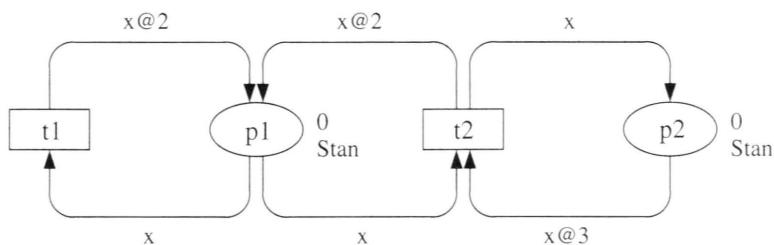
Między kolejnymi przejazdami pociągu przez przejazd kolejowy może upływać czas równy jednej z następujących wartości czasu przejścia:

- od stanu (M_1, S_1) do (M_2, S_2) ,
- od stanu (M_1, S_1) do (M_3, S_3) ,
- od stanu (M_2, S_2) do (M_1, S_1) ,
- od stanu (M_3, S_3) do (M_1, S_1) .

We wszystkich przypadkach minimalny i maksymalny czas przejścia między wskazanymi dwoma stanami sieci wynosi 3 jednostki czasu.

14.2. Grafy pokrycia

Grafy osiągalności dla RTCP-sieci nie zawsze są grafami skończonymi. Rozważmy przykład sieci N przedstawionej na rys. 14.2. Znakowanie początkowe sieci N , $M_0 = (0, 0)$, nie ulega zmianie podczas pracy sieci. Stany sieci zmieniają się ze względu na zmiany pieczętek czasowych poszczególnych miejsc. Sieć N nie jest siecią sprawiedliwą. Przejście $t2$ może zostać zagłodzone przez przejście $t1$. Jeżeli podczas pracy sieci będzie wykonywane tylko przejście $t1$, to pieczętka czasowa miejsca $p2$ będzie nieograniczenie maleć, a zatem dla sieci o tak prostej strukturze otrzymamy nieskończony graf osiągalności.



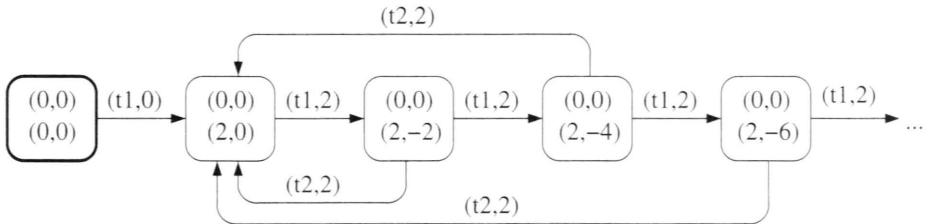
```

color Stan = int with 0..1;
var x : Stan;

```

Rys. 14.2. Przykład niesprawiedliwej RTCP-sieci

Fragment grafu osiągalności dla rozważanej sieci przedstawiono na rys. 14.3. Na podstawie tego fragmentu można łatwo wywnioskować, jaką postać mają pozostałe stany sieci i jaka jest struktura pełnego grafu osiągalności.



Rys. 14.3. Fragment grafu osiągalności dla sieci na rys. 14.2

Rozważmy dwa przykładowe stany sieci \mathcal{N} : (M_0, S_1) i (M_0, S_2) , przy czym $M_0 = (0,0)$, $S_1 = (2,-4)$ i $S_2 = (2,-6)$. Wychodząc z obu tych stanów, można wykonać dokładnie takie same ciągi przejść. Ponadto oba stany reprezentują takie samo znakowanie, a wektory pieczętek czasowych S_1 i S_2 określają takie same poziomy dostępności znaczników. W obu przypadkach należy zatem odczekać 2 jednostki czasu na dostęp do znacznika w miejscu $p1$ i w obu przypadkach dostępny jest znacznik w miejscu $p2$. Znacznik w miejscu $p2$ staje się dostępny, jeżeli jego wiek osiągnie co najmniej 3 jednostki czasu. Dalsze starzenie się znacznika nie wpływa na jego dostępność. O stanach takich mówimy, że się pokrywają. W grafie pokrycia będą one reprezentowane za pomocą pojedynczego węzła.

Niech dana będzie RTCP-siec $\mathcal{N} = (\Sigma, P, T, A, C, G, I, E_M, E_S, M_0, S_0)$.

Definicja 14.2. Niech $p \in P$ będzie miejscem sieci \mathcal{N} i niech $Out_A(p)$ oznacza zbiór łuków wychodzących z miejsca p .

Minimalnym wiekiem dostępu dla miejsca p nazywamy liczbę:

$$\delta_{\min}(p) = \min_{a \in Out_A(p)} \left\{ \min_{b \in \mathcal{B}(T(a))} E_S(a)_b \right\}. \quad (14.6)$$

Maksymalnym wiekiem dostępu dla miejsca p nazywamy liczbę:

$$\delta_{\max}(p) = \max_{a \in Out_A(p)} \left\{ \max_{b \in \mathcal{B}(T(a))} E_S(a)_b \right\}. \quad (14.7)$$

Minimalny wiek dostępu dla miejsca p jest wiekiem, w którym znaczniki w miejscu p stają się dostępne dla co najmniej jednego przejścia wyjściowego tego miejsca (najczęściej wartością tą jest 0). Maksymalny wiek dostępu jest wiekiem, w którym znaczniki w miejscu p stają się dostępne dla wszystkich przejść wyjściowych tego miejsca.

Definicja 14.3. Niech $(M_1, S_1), (M_2, S_2)$ będą stanami sieci \mathcal{N} . Mówimy, że stan (M_1, S_1) pokrywa stan (M_2, S_2) (zapisujemy to jako $(M_1, S_1) \simeq (M_2, S_2)$), jeżeli $M_1 = M_2$ oraz

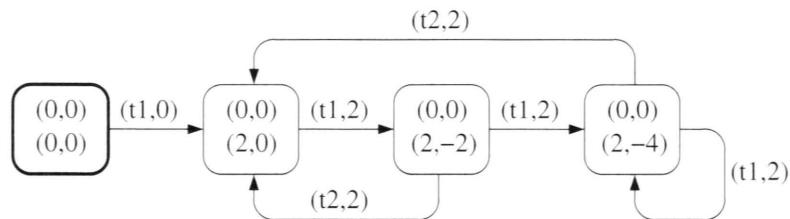
$$\begin{aligned} \forall p \in P: (S_1(p) = S_2(p)) \vee \\ (S_1(p) \leq -\delta_{\max}(p) \wedge S_2(p) \leq -\delta_{\max}(p)). \end{aligned} \quad (14.8)$$

Twierdzenie 14.1. Relacja \simeq pokrywania się stanów jest relacją równoważności. ■

Dowód powyższego twierdzenia można pominąć, ponieważ jest bardzo prosty.

Grafy pokrycia dla RTCP-sieci budujemy podobnie jak graf osiągalności, przy czym po wyznaczeniu nowego węzła, jeżeli w grafie istnieje już węzeł pokrywający ten nowowyznaczony, to do grafu dodawany jest tylko łuk do węzła pokrywającego. W porównaniu z pełnym grafem osiągalności węzły grafu pokrycia reprezentują nie pojedyncze stany, ale klasy abstrakcji relacji pokrywania się stanów.

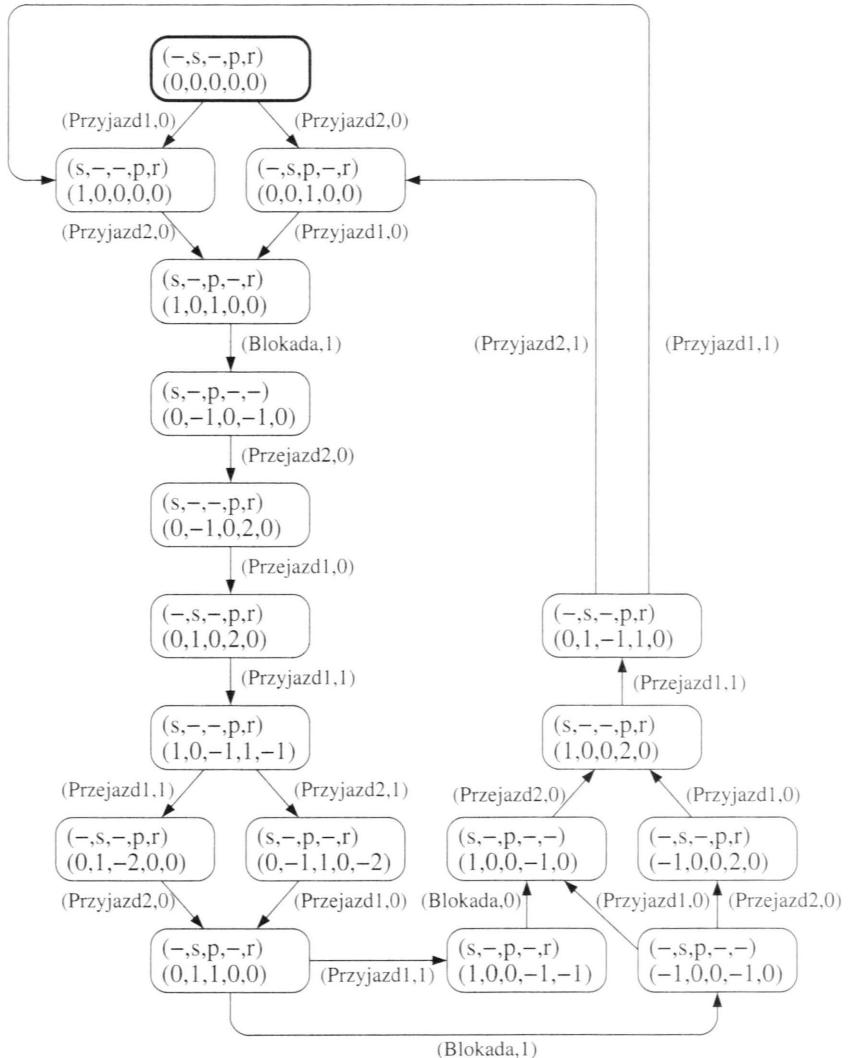
Powróćmy jeszcze do sieci przedstawionej na rys. 14.2. Po wyznaczeniu stanu (M_0, S_2) stwierdzamy, że w grafie istnieje już węzeł pokrywający ten stan (węzeł (M_0, S_1)), a zatem do grafu dołączamy łuk prowadzący od stanu (M_0, S_1) do niego samego. Graf pokrycia dla sieci pokazanej na rys. 14.2 przedstawiono na rys 14.4.



Rys. 14.4. Graf pokrycia dla sieci z rys. 14.2

Graf pokrycia umożliwia analizowanie podobnych własności sieci jak pełny graf osiągalności. Reprezentuje on jawnie wszystkie znakowania sieci, a zatem pozwala dokładnie określić ograniczonosć miejsc. Wszystkie etykiety łuków, które występują w pełnym grafie osiągalności, znajdują się również w grafie pokrycia. Ponadto każdej drodze w grafie osiągalności odpowiada w grafie pokrycia droga wyznaczona przez łuki o identycznych etykietach. Oznacza to, że graf pokrycia pozwala dokładnie określić poziom żywotności sieci. Pewne ograniczenia dotyczą analizy własności czasowych, gdyż nie wszystkie stany są reprezentowane jawnie. Zasadniczą zaletą grafu pokrycia polega na tym, że dla sieci ograniczonych jest on w większości przypadków grafem skończonym.

Jeżeli przyjmiemy, że każde miejsce sieci \mathcal{N} ma skończone górne ograniczenie wielozbiorowe i wszystkie typy $\Sigma_i \in \Sigma$ są zbiorami skończonymi, to dla sieci \mathcal{N} można udowodnić, że jej graf pokrycia jest grafem skończonym, mimo że zbiór stanów osiągalnych może być nieskończony. Dowód tego twierdzenia można znaleźć w pracy [79].



Rys. 14.5. Graf pokrycia dla RTCP-sieci modelującej sterowanie rogatekami

Rozważmy ponownie model sterowania rogatkami na przejeździe kolejowym, przedstawiony na rys. 13.19. Wśród węzłów grafu osiągalności na rys. 14.1 można wskazać trzy pary pokrywających się stanów:

- $((s, -, -, p, r); (1, 0, 0, 0, 0))$ i $((s, -, -, p, r); (1, 0, -2, 0, -1))$,
- $((-, s, p, -, r); (0, 0, 1, 0, 0))$ i $((-, s, p, -, r); (-1, 0, 1, 0, -1))$,
- $((s, -, p, -, r); (1, 0, 1, 0, 0))$ i $((s, -, p, -, r); (1, 0, 1, 0, -1))$.

Graf pokrycia dla rozważanej sieci przedstawiono na rys. 14.5.

Dla RTCP-sieci można stosować analogiczne metody redukcji grafu osiągalności jak dla sieci kolorowanych, na przykład grafy silnie spójnych składowych, grafy relacyjne itp. Redukcja grafu osiągalności prowadzi jednak często do ograniczenia możliwości analizy własności czasowych modelu.

RTCP-sieci można również analizować, stosując niezmienniki sieci. Następujące niezmienniki miejsc pozwalają na wyznaczenie liczbowych i wielozbiorowych ograniczeń dla większości miejsc sieci na rys. 13.19, bez konieczności korzystania z grafu osiągalności:

$$\begin{aligned} M(\text{Droga}) + M(\text{Otoczenie1}) &= s, \\ M(\text{Tory}) + M(\text{Otoczenie2}) &= p. \end{aligned} \tag{14.9}$$

14.3. Podsumowanie

Omówiliśmy w tym rozdziale grafy osiągalności i grafy pokrycia, które są podstawą analizy RTCP-sieci. Pozwalają one badać nie tylko własności dotyczące ograniczoności i żywotności sieci, ale również własności czasowe, które odgrywają ważną rolę w weryfikacji systemów współbieżnych z czasem. Warto podkreślić fakt, że dla sieci, których nieskończona przestrzeń stanów jest wynikiem zależności czasowych, można zbudować skończony graf pokrycia umożliwiający weryfikację większości własności modelu. Ponadto do weryfikacji RTCP-sieci można stosować również inne metody typowe dla sieci kolorowanych, na przykład posługując się niezmiennikami miejsc i przejść.

Zakończenie

Sieci Petriego są działem informatyki rozwijanym od ponad czterdziestu lat. Przedstawione w książce zagadnienia składają się na stosunkowo obszerne, ale zaledwie wprowadzenie w tę tematykę. Słowo *wprowadzenie* wydaje się jak najbardziej na miejscu. Literatura dotycząca sieci Petriego zawiera tysiące pozycji, w których można znaleźć kilkadziesiąt różnych klas sieci, a corocznie na świecie odbywa się kilkanaście konferencji naukowych, poświęconych w całości lub częściowo tym zagadnieniom. Wszystko to powoduje, że napisanie książki, którą można by potraktować jak przegląd całego tego dorobku, wydaje się niemożliwe.

Jednym z założeń, którymi się kierowałem, pisząc tę książkę, było znalezienie pewnej równowagi między przedstawieniem zagadnień teoretycznych a wskazaniem możliwości praktycznego zastosowania poszczególnych klas sieci. Wybór przedstawionych klas był jak najbardziej subiektywny. W książce zaprezentowałem zarówno klasy sieci bardzo często spotykane w literaturze (np. sieci uogólnione, sieci kolorowane), jak również propozycje nowych rozwiązań (RTCP-sieci). Wybór klas sieci był podyktowany między innymi chęcią ukazania zależności między nimi oraz zwrócenia uwagi na to, jak modyfikując definicje, można przejść od jednej klasy do drugiej. Były one wybierane pod kątem przydatności do modelowania systemów współbieżnych z czasem. Dla każdej z opisanych klas sieci zamieściłem przykłady modelowania z ich użyciem, jak również omówiłem metody analizy najważniejszych ich własności. Zaprezentowane przykłady mają charakter akademicki, jednakże powinny okazać się przydatne przy opracowywaniu własnych, bardziej złożonych modeli.

W książce starałem się wprowadzić jednolitą konwencję notacyjną zarówno przy przedstawianiu graficznej notacji sieci, jak i w zapisie zagadnień teoretycznych. Duża liczba terminów i oznaczeń bardzo utrudniała to zadanie. Na ile udało mi się osiągnąć zamierzony cel, ocenią czytelnicy książki. W nielicznych przypadkach ten sam symbol ma różne znaczenia w zależności od klasy sieci, lecz nie powinno to wpływać na zrozumienie treści książki.

Dla wielu klas sieci, jak również dla wielu zagadnień dotyczących omawianych klas zabrakło miejsca w tym opracowaniu. Odpowiednim punktem wyjścia do poszukiwania dalszych informacji na ten temat jest strona internetowa <http://www.informatik.uni-hamburg.de/TGI/PetriNets/> (Petri Nets World). Zawiera ona olbrzymią bazę bibliograficzną, bazę danych o narzędziach komputerowych wspomagających konstruowanie i analizę sieci, liczne informacje o konferencjach naukowych i wiele innych cennych informacji.

A

Polsko-angielski słownik terminów

ciasna pętla	self-loop
dolne ograniczenie	lower integer bound
dolne ograniczenie wielozbiorowe	lower multiset bound
dozór	guard
droga nieskierowana	undirected path, chain
droga skierowana	directed path, path
dziedzina czasu wykonania	firing domain
funkcja przydziału portów	port assignment function
funkcja przydziału stron	page assignment function
funkcja respektuje łuki sieci	function respects flow relation
funkcja respektuje typ węzła	function respects node type
funkcja respektuje wagę łuków	function respects arcs multiplicities
funkcja typów	colour function
funkcja typów fuzji	fusion type function
funkcja typów portów	port type function
funkcja zaczepienia	node function
fuzja	fusion set
fuzja globalna	global fusion set
fuzja instancji strony	instance fusion set
fuzja stronicowa	page fusion set
gniazdo	socket node
górnne ograniczenie	upper integer bound
górnne ograniczenie wielozbiorowe	upper multiset bound
graf	graph
graf acykliczny	acyclic graph
graf dwudzielny	bipartite graph
graf etykietowany	labelled graph
graf klas	class graph

graf osiągalności	reachability graph
graf pokrycia	coverability graph
graf silnie spójny	strong connected graph
graf silnie spójnych składowych	strong connected components graph
graf skierowany	directed graph
graf spójny	connected graph
graf synchronizacji	marked graph, event graph
graf zredukowany ze względu na symetrie	reachability graph with symmetries
klasa stanów	state class
konflikt dynamiczny	dynamic conflict
konflikt statyczny	static conflict
łuk	arc
macierz incydencji	incidence matrix
macierz wejść	input matrix
macierz wyjść	output matrix
maszyna stanowa	state machine, state graph
miejsce	place
miejsce bezpieczne	safe place
miejsce czyste	pure place
miejsce gniazdowe	socket node
miejsce nieograniczone	unbounded place
miejsce ograniczone	bounded place
miejsce portowe	port node
miejsce znakowane	marked place
miejsce żywe	live place
multigraf	multigraph
następnik	successor
niezmiennik miejsc	place invariant, S-invariant
niezmiennik przejść	transition invariant, T-invariant
opóźnienie dynamiczne	dynamic delay
opóźnienie statyczne	static delay
pieczętka czasowa	time stamp
podsieć	subnet
podsieć jednolita	homogeneous subnet
podsieć nieblokująca	non-blocking subnet
podsieć silnie nieblokująca	strongly non-blocking subnet

podstawiane przejście	substitution transition
pojemność miejsca	place capacity
poprzednik	predecessor
port	port node
priorytet przejścia	transition priority
przedziałowe sieci czasowe	time nets, interval time nets
przejście	transition
przejście jednolite	uniform transition
przejście kluczowe	key transition
przejście martwe	dead transition
przejście potencjalnie wykonalne	potentially firable transition
przejście zachowawcze	conservative transition
przejście żywe	live transition
pułapka	trap
relacja przepływu	flow relation
relacyjny graf osiągalności	reachability graph with equivalence classes
rozszerzona sieć prosta	extended simple net
rozszerzona sieć swobodnego wyboru	extended free choice net
scalanie kolejnych miejsc	fusion of series places
scalanie kolejnych przejść	fusion of series transitions
scalanie równoległych miejsc	fusion of parallel places
scalanie równoległych przejść	fusion of parallel transitions
sieć	net
sieć bezpieczna	safe net
sieć czasowa	time net
sieć czysta	pure net
sieć kolorowana	coloured net
sieć miejsc i przejść	place-transition net, PT-net
sieć nieograniczona	unbounded net
sieć odtwarzalna	reproducible net
sieć odwrocalna	reversible net
sieć ograniczona	bounded net
sieć priorytetowa	priority net
sieć prosta	simple net
sieć rozkładalna na maszyny stanowe	state machine decomposable net
sieć skończona	finite net
sieć sprawiedliwa	fair net
sieć strukturalnie żywa	structurally live net
sieć swobodnego wyboru	free choice net

sieć trwała	persistent net
sieć uogólniona	generalized net
sieć zachowawcza	conservative net
sieć znakowana	marked net
sieć żywa	live net
sieć żywa ze względu na miejsca	place-live net
składowa	component
specyfikacja równoważności	equivalence specification
sprawiedliwy ciąg przejść	fair firing sequence
stan	state
stan martwy	dead state
stan początkowy	initial state
stan odtwarzalny	reproducible state
stan odwrocalny	reversible state
stan własny	home state
symetria	symmetry
tranzycja	transition
trwały zbiór przejść	stubborn set
typ miejsca	colour
usuwanie ciasnych pętli miejsc	elimination of self-loop places
usuwanie ciasnych pętli przejść	elimination of self-loop transitions
waga łuku	arc weight
węzeł	node, vertex
węzeł izolowany	isolated node
węzeł końcowy	output node
węzeł pełny	full node
węzeł początkowy	input node
wiązanie	binding
wielozbiór	multiset
wielozbiór czasowy	timed multiset
wielozbiór uogólniony	weighted-set
własność maksymalnie kontrolowanych zatrzasków	max-controlled siphon property
własność minimalnie kontrolowanych zatrzasków	min-controlled siphon property
własność zatrzasku-pułapki	siphon-trap property
zastrzeżenie	guard
zbiór maksymalnie znakowany	max-marked set

zbiór minimalnie znakowany	min-marked set
zatrzask	deadlock, siphon
zatrzask maksymalnie kontrolowany	max-controlled siphon
zatrzask minimalnie kontrolowany	min-controlled siphon
znacznik	token
znakowanie	marking
znakowanie martwe	dead marking
znakowanie odtwarzalne	reproducible marking
znakowanie odwracalne	reversible marking
znakowanie początkowe	initial marking
znakowanie pokrywalne	coverable marking
znakowanie uogólnione	generalized marking
znakowanie własne	home marking
znakowanie żywe	live marking



Angielsko-polski słownik terminów

acyclic graph	graf acykliczny
arc	łuk
arc weight	waga łuku
binding	wiązanie
bipartite graph	graf dwudzielny
bounded net	sieć ograniczona
bounded place	miejsce ograniczone
chain	droga nieskierowana
class graph	graf klas
colour	typ miejsca
colour function	funkcja typów
coloured net	sieć kolorowana
component	składowa
connected graph	graf spójny
conservative net	sieć zachowawcza
conservative transition	przejście zachowawcze
coverability graph	graf pokrycia
coverable marking	znakowanie pokrywalne
deadlock	zatrzask
dead marking	znakowanie martwe
dead state	stan martwy
dead transition	przejście martwe
directed graph	graf skierowany
directed path	droga skierowana
dynamic conflict	konflikt dynamiczny
dynamic delay	opóźnienie dynamiczne
elimination of self-loop places	usuwanie ciasnych pętli miejsc
elimination of self-loop transitions	usuwanie ciasnych pętli przejść

equivalence specification	specyfikacja równoważności
event graph	graf synchronizacji
extended free choice net	roszzerzona sieć swobodnego wyboru
extended simple net	roszzerzona sieć prosta
fair firing sequence	sprawiedliwy ciąg przejść
fair net	sieć sprawiedliwa
finite net	sieć skończona
firing domain	dziedzina czasu wykonania
flow relation	relacja przepływu
free choice net	sieć swobodnego wyboru
full node	węzeł pełny
function respects arcs multiplicities	funkcja respektuje wagę łuków
function respects flow relation	funkcja respektuje łuki sieci
function respects node type	funkcja respektuje typ węzła
fusion of parallel places	scalanie równoległych miejsc
fusion of parallel transitions	scalanie równoległych przejść
fusion of series places	scalanie kolejnych miejsc
fusion of series transitions	scalanie kolejnych przejść
fusion set	fuzja
fusion type function	funkcja typów fuzji
generalized marking	znakowanie uogólnione
generalized net	sieć uogólniona
global fusion set	fuzja globalna
graph	graf
guard	dozór, zastrzeżenie
home marking	znakowanie własne
home state	stan własny
homogeneous subnet	podsieć jednolita
incidence matrix	macierz incydencji
initial marking	znakowanie początkowe
initial state	stan początkowy
input matrix	macierz wejść
input node	węzeł początkowy
instance fusion set	fuzja instancji strony
interval time nets	przedziałowe sieci czasowe
isolated node	węzeł izolowany
key transition	przejście kluczowe

labelled graph	graf etykietowany
live marking	znakowanie żywe
live net	sieć żywa
live place	miejsce żywe
live transition	przejście żywe
lower integer bound	dolne ograniczenie
lower multiset bound	dolne ograniczenie wielozbiorowe
marked graph	graf synchronizacji
marked net	sieć znakowana
marked place	miejsce znakowane
marking	znakowanie
max-controlled siphon	zatrzask maksymalnie kontrolowany
max-controlled siphon property	własność maksymalnie kontrolowanych zatrzasków
max-marked set	zbiór maksymalnie znakowany
min-controlled siphon	zatrzask minimalnie kontrolowany
min-controlled siphon property	własność minimalnie kontrolowanych zatrzasków
min-marked set	zbiór minimalnie znakowany
multigraph	multigraf
multiset	wielozbior
net	sieć
node	węzeł
node function	funkcja zaczepienia
non-blocking subnet	podsieć nieblokująca
output matrix	macierz wyjścia
output node	węzeł końcowy
page assignment function	funkcja przydziału stron
page fusion set	fuzja stronicowa
path	droga skierowana
persistent net	sieć trwała
place	miejsce
place capacity	pojemność miejsca
place invariant	niezmiennik miejsc
place-live net	sieć żywa ze względu na miejsca
place-transition net	sieć miejsc i przejść
port assignment function	funkcja przydziału portów
port node	port, miejsce portowe

port type function	funkcja typów portów
potentially firable transition	przejście potencjalnie wykonalne
predecessor	poprzednik
priority net	sieć priorytetowa
pure net	sieć czysta
pure place	miejsce czyste
reachability graph	graf osiągalności
reachability graph with equivalence classes	relacyjny graf osiągalności
reachability graph with symmetries	
reproducible marking	graf zredukowany
reproducible net	ze względu na symetrie
reproducible state	znakowanie odtwarzalne
reversible marking	sieć odtwarzalna
reversible net	stan odtwarzalny
reversible state	znakowanie odwracalne
safe net	sieć odwracalna
safe place	stan odwracalny
self-loop	
simple net	sieć bezpieczna
S-invariant	miejsce bezpieczne
siphon	ciasna pętla
siphon-trap property	sieć prosta
socket node	niezmiennik miejsc
state	zatrzask
state class	własność zatrzasku-pułapki
state graph	gniazdo, miejsce gniazdowe
state machine	stan
state machine decomposable	klasa stanów
static conflict	maszyna stanowa
static delay	maszyna stanowa
strong connected components graph	sieć rozkładalna na maszyny stanowe
strong connected graph	konflikt statyczny
strongly non-blocking subnet	opóźnienie statyczne
structurally live net	graf silnie spójnych składowych
stubborn set	graf silnie spójny
subnet	podsieć silnie nieblokująca
substitution transition	sieć strukturalnie żywa
	trwały zbiór przejścia
	podsieć
	podstawiane przejście

successor	następnik
symmetry	symetria
time net	sieć czasowa
time stamp	pieczętka czasowa
timed multiset	wielozbiór czasowy
T-invariant	niezmiennik przejść
token	znacznik
transition	przejście, tranzycja
transition invariant	niezmiennik przejść
transition priority	priorytet przejścia
trap	pułapka
unbounded net	sieć nieograniczona
unbounded place	miejsce nieograniczone
undirected path	droga nieskierowana
uniform transition	przejście jednolite
upper integer bound	górnne ograniczenie
upper multiset bound	górnne ograniczenie wielozbiorowe
vertex	węzeł
weighted-set	wielozbiór uogólniony

Wybrane narzędzia do konstruowania i analizy sieci Petriego

Dodatek ten zawiera krótkie informacje o narzędziach komputerowych, z których korzystałem przy pisaniu tej książki. Więcej informacji na temat narzędzi służących do modelowania i analizy sieci Petriego można znaleźć na stronie <http://www.informatik.uni-hamburg.de/TGI/PetriNets/> (Petri Nets World).

APE *Advanced Petri nets Environment* jest kierowanym przeze mnie projektem oprogramowania przeznaczonego do tworzenia i analizowania sieci Petriego niskiego poziomu. Aktualna wersja zawiera edytor graficzny i symulator sieci oraz zaimplementowane algorytmy wyznaczania grafów osiągalności, grafów pokrycia i niezmienników sieci. Oprogramowanie APE wspiera następujące klasy sieci Petriego: sieci miejsc i przejść, sieci priorytetowe, proste sieci czasowe i proste sieci czasowe z priorytetami. Program jest oparty na licencji GPL i dostępny dla użytkowników systemów operacyjnych Linux/UNIX, MacOS X i Windows. Strona domowa projektu: <http://fm.ia.agh.edu.pl>.

Design/CPN jest narzędziem przeznaczonym do tworzenia i analizowania kolorowanych sieci Petriego. Oprócz graficznego edytora i symulatora, zaimplementowano w nim m.in. algorytmy wyznaczania grafów osiągalności, grafów silnie spójnych składowych i grafów relacyjnych. Z programu mogą korzystać nieodpłatnie w wypadku zastosowań niekomercyjnych użytkownicy systemów Linux/UNIX. Obecnie jest opracowywana nowa wersja tego oprogramowania o nazwie **CPN Tools**, której można używać, korzystając z systemów Linux/UNIX i Windows. Jednak jeszcze nie wszystkie metody analizy zaimplementowane w Design/CPN zostały przeniesione do CPN Tools. Strona domowa projektu: <http://www.daimi.au.dk/designCPN>.

INA *Integrated Net Analyzer* jest zestawem narzędzi służących do analizowania sieci miejsc i przejść oraz sieci kolorowanych. Formatem wejściowym danych są pliki tekstowe zawierające opis sieci. W programie zaimplementowano większość metod analizy omówionych w rozdz. 5. Z programu mogą korzystać nieodpłatnie użytkownicy systemów operacyjnych: Linux/UNIX, Sun-OS, Solaris

i Windows. Strona domowa projektu:

<http://www.informatik.hu-berlin.de/~starke/ina.html>.

Tina *TIme petri Net Analyzer* jest narzędziem służącym do tworzenia i analizowania przedziałowych sieci czasowych. Oprócz graficznego edytora i symulatora, zaimplementowano w nim m.in. algorytmy wyznaczania grafów klas dla sieci przedziałowych. Z programu mogą korzystać nieodpłatnie użytkownicy systemów operacyjnych: Linux/UNIX, MacOS X, Sun-OS, Solaris i Windows. Strona domowa projektu: <http://www.laas.fr/tina>.



Lista symboli

2^{A^*}	rodzina wszystkich wielozbiorów zbudowanych nad zbiorem A
2^{A^\ddagger}	rodzina wszystkich wielozbiorów uogólnionych zbudowanych nad zbiorem A
$2^{A_t^*}$	rodzina wszystkich wielozbiorów czasowych zbudowanych nad zbiorem A
γ	funkcja zaczepienia
δ	funkcja (wektor) opóźnień dynamicznych
$\delta((M, \delta), (M', \delta'))$	czas przejścia od stanu (M, δ) do (M', δ') (w zależności od klasy sieci, również: $\delta((M, r), (M', r'))$, $\delta((M, S), (M', S'))$)
$\delta(\alpha)$	czas wykonania ciągu przejść (wiązań) α
δ_{\max}	maksymalne opóźnienie dynamiczne
$\delta_{\max}(p)$	maksymalny wiek dostępu dla miejsca p
δ_{\min}	minimalne opóźnienie dynamiczne
$\delta_{\min}(p)$	minimalny wiek dostępu dla miejsca p
σ	funkcja (wektor) opóźnień statycznych
σ_{\max}	maksymalne opóźnienie statyczne
σ_{\min}	minimalne opóźnienie statyczne
Σ	zbiór typów (kolorów) miejsc
A	zbiór łuków grafu (sieci)
$A(x)$	zbiór łuków otaczających węzeł x
A^*, A_1^*, A_2^*, \dots	wielozbiory nad zbiorem A
$A^*(x)$	współczynnik elementu x w wielozbiorze A^*
$A^\ddagger, A_1^\ddagger, A_2^\ddagger, \dots$	wielozbiory uogólnione nad zbiorem A

$A_t^*, A_{t1}^*, A_{t2}^*, \dots$	wielozbiory czasowe nad zbiorem A
$A_t^*(x)$	współczynnik elementu x w wielozbiorze czasowym A_t^*
$A_t^*[x]$	lista pieczętek czasowych elementów x w wielozbiorze czasowym A_t^*
\mathcal{B}	zbiór par (t, b) , przy czym $t \in T$ i $b \in \mathcal{B}(t)$
$\mathcal{B}(t)$	zbiór wiązań przejścia t
$Bool$	zbiór wartości logicznych
C	funkcja typów (kolorów) miejsc
C_X	wektor charakterystyczny zbioru X
E	funkcja wag łuków (sieci kolorowane)
$E(x, y)$	suma wag łuków o początku x i końcu y (sieci kolorowane)
$E(x, y)_b$	wynik wartościowania wyrażenia $E(x, y)$ przy wiązaniu b
E_M	funkcja wag łuków (RTCP-sieci)
E_S	funkcja wag czasowych (RTCP-sieci)
F_P	zbiór wag miejsc (sieci kolorowane)
$\mathcal{G} = (V, A)$	graf skierowany
$\mathcal{G} = (V, A, \gamma)$	graf skierowany (multigraf)
G	funkcja zastrzeżeń (dozorów) przejść
\mathcal{G}_S	zbiór silnie spójnych składowych grafu \mathcal{G}
I	funkcja priorytetów przejść
I, I_1, I_2, \dots	niezmienniki sieci
$In(x)$	zbiór poprzedników węzła x
K	funkcja pojemności miejsc
L	zbiór etykiet grafu
$\mathcal{L}(M)$	zbiór ciągów przejść, które można wykonać, rozpoczynając od znakowania M
$\mathcal{L}(M, \delta)$	zbiór ciągów przejść, które można wykonać, rozpoczynając od stanu (M, δ) (w zależności od klasy sieci również: $\mathcal{L}(M, r)$, $\mathcal{L}(M, S)$)
(M, δ)	stan sieci (w zależności od klasy sieci również: (M, r) , (M, S))
M, M_1, M_2, \dots	znakowania sieci

M_0	znakowanie początkowe
(M_0, δ_0)	stan początkowy (w zależności od klasy sieci również: $(M_0, r_0), (M_0, S_0)$)
$\mathbb{N} = \{1, 2, \dots\}$	zbiór liczb naturalnych
\mathcal{N}	sieć Petriego
N	macierz incydencji
N^+	macierz wejść
N^-	macierz wyjść
$Out(x)$	zbiór nastęników węzła x
P	zbiór miejsc sieci
p, p_i, p_1, p_2, \dots	miejsc sieci
$P(a)$	węzeł łuku a będący miejscem
\mathbb{Q}	zbiór liczb wymiernych
\mathbb{Q}_+	zbiór liczb wymiernych nieujemnych
\mathbb{R}	zbiór liczb rzeczywistych
\mathbb{R}_+	zbiór liczb rzeczywistych nieujemnych
R	zbiór pieczętek czasowych (czasowe sieci kolorowane)
$\mathcal{R}(M)$	zbiór znakowań osiągalnych ze znakowania M
$\mathcal{R}(M, \delta)$	zbiór stanów osiągalnych ze stanu (M, δ) (w zależności od klasy sieci również: $\mathcal{R}(M, r), \mathcal{R}(M, S)$)
r_0	czas początkowy (czasowe sieci kolorowane)
S	zbiór symetrii sieci
S_0	początkowy rozkład pieczętek czasowych (RTCP-sieci)
T	zbiór przejść sieci
t, t_i, t_1, t_2, \dots	przejścia sieci
$T(a)$	węzeł łuku a będący przejściem
$\mathcal{T}(V)$	typ zbioru zmiennych V
$\mathcal{T}(v)$	typ zmiennej (wyrażenia)
V	zbiór węzłów grafu (sieci)
$\mathcal{V}(t)$	zbiór zmiennych występujących w zastrzeżeniu przejścia t lub w wyrażeniach będących wagami łuków otaczających to przejście

$\mathcal{V}(x)$	zbiór zmiennych występujących w wyrażeniu x
W	funkcja wag łuków (sieci niskiego poziomu)
$ X $	moc zbioru X
\mathbb{Z}	zbiór liczb całkowitych
\mathbb{Z}_+	zbiór liczb całkowitych nieujemnych

Literatura

- [1] Aalst van der W. M. P.: Interval timed coloured Petri nets and their analysis. W: *Proceedings of the 14th International Conference on Application and Theory of Petri Nets*, LNCS, Vol. 691, London, Springer-Verlag 1993, s. 453–472.
- [2] Barkaoui K., Pradat-Peyre J.: On liveness and controlled siphons in Petri nets. W: *Proceedings of the 17th International Conference on Application and Theory of Petri Nets*, LNCS, Vol. 1091, London, Springer-Verlag 1996, s. 57–72.
- [3] Ben-Ari M.: *Podstawy programowania współbieżnego i rozproszonego*. Warszawa, WNT 1996 (tłum. z ang.).
- [4] Berthelot G.: Transformations and decompositions of nets. W: *Advances in Petri nets 1986 Part I*, LNCS, Vol. 254, London, Springer-Verlag 1987, s. 359–376.
- [5] Berthomieu B., Diaz M.: Modeling and verification of time dependent systems using time Petri nets. *IEEE Transactions on Software Engineering*, 1991, Vol. 17, No. 3, s. 259–273.
- [6] Berthomieu B., Ribet P.-O., Vernadat F.: The tool TINA – construction of abstract state spaces for Petri nets and time Petri nets. *International Journal of Production Research*, 2004, Vol. 42, No. 14, s. 2741–2756.
- [7] Berthomieu B., Vernadat F.: State class constructions for branching analysis of time Petri nets. W: *Proceedings of the 9th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'03)*, LNCS, Vol. 2619, London, Springer-Verlag 2003, s. 442–457.
- [8] Bowden F. D. J.: Modelling time in Petri nets. W: *Proceedings of the Second Australia-Japan Workshop on Stochastic Models in Engineering, Technology and Management*, Gold Coast, AU, 1996.
- [9] Buy U. A., Sloan R. H.: Analysis of real-time programs with simple time Petri nets. W: *Proceedings of the International Symposium on Software Testing and Analysis*, New York, NY, ACM Press 1994, s. 228–239.
- [10] Cerone A., Maggiolo-Schettini A.: Time-based expressivity of time Petri nets for system specification. *Theoretical Computer Science*, 1999, Vol. 216, No. 1–2, s. 1–53.
- [11] Cheng A. M. K.: *Real-time systems. Scheduling, analysis, and verification*. New Jersey, NJ, Wiley Interscience 2002.
- [12] Chiola G.: On the structural and behavioural characterization of P/T nets. W: *Proceedings of 5th International Workshop on Petri Nets and Performance Models (PNPM'93)*, Toulouse, FR, 1993, s. 66–75.
- [13] Christensen S., Kristensen L. M.: State space analysis of hierarchical coloured Petri nets. W: *Proceedings of Workshop on Petri Nets in System Engineering, Modelling, Verification and Validation*, Hamburg, 1997, s. 32–43.

- [14] Commoner F.: Deadlocks in Petri nets. Technical Report CA-7206-2311, Applied Data Research Inc., Wakefield, MA, 1972.
- [15] Coolahan J. J. E., Roussopoulos N.: Timing requirements for time-driven systems using augmented Petri nets. *IEEE Transactions on Software Engineering*, 1983, Vol. SE-9, No. 5, s. 603–616.
- [16] Corbett J. C.: Timing analysis of Ada tasking programs. *IEEE Transactions on Software Engineering*, 1996, Vol. 22, No. 7, s. 461–483.
- [17] Cortadella J., Yakovlev A., Rozenberg G., (red.): *Concurrency and hardware design: Advances in Petri nets*. LNCS, Vol. 2549. New York, NY, Springer-Verlag 2002.
- [18] Dadda L.: The synthesis of Petri nets for controlling purposes and the reduction of their complexity. Euromicro 1976.
- [19] David R., Alla H.: *Discrete, continuous and hybrid Petri nets*. Berlin, Springer-Verlag 2005.
- [20] Duri S., Buy U. A., Devarapalli R., Shatz S. M.: Application and experimental evaluation of state space reduction methods for deadlock analysis in Ada. *ACM Transactions on Software Engineering and Methodology*, 1994, Vol. 3, No. 4, s. 340–380.
- [21] Frutos-Escrí de D., Ruiz V. V., Alonso O. M.: Decidability of properties of timed-arc Petri nets. W: *Proceedings of the 21st International Conference on Application and Theory of Petri Nets*, LNCS, Vol. 1825, London, Springer-Verlag 2000, s. 187–206.
- [22] Ghezzi C., Mandrioli D., Morasca S., Pezč M.: A Unified high-level Petri net formalism for time-critical systems. *IEEE Transactions on Software Engineering*, Feb 1991, Vol. 17, No. 2, s. 160–172.
- [23] Hack M. H. T.: Analysis of production schemata by Petri nets. Praca magisterska, Dep. Electrical Engineering, MIT, Cambridge 1972.
- [24] Hack M. H. T.: Petri net languages. *Computation Structures Group Memo* 124, Project MAC, MIT, 1975.
- [25] Hanisch H. M.: Analysis of Place/Transition nets with timed arcs and its application to batch process control. W: *Proceedings of the 14th International Conference on Application and Theory of Petri Nets*, LNCS, Vol. 691, London, Springer-Verlag 1993, s. 282–299.
- [26] Hanisch H. M., Thieme J., Lautenbach K., Simon C.: Timestamp nets in technical applications. W: *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics (SMC'98)*, San Diego, CA 1998, s. 119–124.
- [27] Holliday M. A., Vernon M. K.: A Generalized timed Petri net model for performance analysis. *IEEE Transactions on Software Engineering*, 1987, Vol. SE-13, No. 12, s. 1297–1310.
- [28] Holt A. W., Commoner F.: Events and conditions. Technical report, Applied Data Research Inc., New York, 1970.
- [29] Huber P., Jensen K., Shapiro R. M.: Hierarchies in coloured Petri nets. W: *Proceedings of the 10th International Conference on Application and Theory of Petri Nets*, Bonn, 1989, s. 192–209.
- [30] Iordache M. V., Moody J. O., Antsaklis P. J.: Automated synthesis of liveness enforcing supervisors using Petri nets, Technical Report ISIS-00-004, University of Notre Dame 2000.
- [31] Jensen K.: Coloured Petri nets and the invariant method. *Theoretical Computer Science*, 1981, Vol. 14, s. 317–336.
- [32] Jensen K.: *Coloured Petri nets. Basic concepts, analysis methods and practical use*. Vol. 1–3. Berlin, Springer-Verlag 1992–1997.
- [33] Jensen K.: An introduction to the practical use of coloured Petri nets. Lectures on Petri Nets II – Applications. 1998, LNCS, Vol. 1492, London, Springer-Verlag 1993, s. 237–292.

- [34] Jensen K., Rozenberg G., (red.): *High-level Petri nets – theory and application*. Berlin, Springer-Verlag 1991.
- [35] Jørgensen J. B., Kristensen L. M.: Verification of coloured Petri nets using state spaces with equivalence classes. W: *Proceedings of Workshop on Petri Nets in System Engineering, Modelling, Verification and Validation*, Hamburg, 1997, s. 20–31.
- [36] Jørgensen J. B., Kristensen L. M.: Computer aided verification of Lamport's fast mutual exclusion algorithm using coloured Petri nets occurrence graphs with symmetries. *IEEE Transactions on Parallel and Distributed Systems*, 1999, Vol. 10, No. 7, s. 714–732.
- [37] Karp R. M., Miller R. E.: Parallel program schemata. *Journal of Computer and System Sciences* 3, 1969, No. 2, s. 147–195.
- [38] Kristensen L. M., Christensen S., Jensen K.: The practitioner's guide to coloured Petri nets. *International Journal on Software Tools for Technology Transfer: Special Section on Coloured Petri Nets*, 1998, Vol. 2, No. 2, s. 98–132.
- [39] Kristensen L. M., Valmari A.: Finding stubborn sets of coloured Petri nets without unfolding. W: *Proceedings of the 19th International Conference on Application and Theory of Petri Nets*, LNCS, Vol. 1420, London, Springer-Verlag 1998, s. 104–123.
- [40] Kristensen L. M., Valmari A.: Improved question-guided stubborn set methods for state properties. Technical Report DAIMI PB-543, University of Aarhus, Department of Computer Science 1999.
- [41] Lamport L.: Proving the correctness of multiprocess programs. *IEEE Transaction on Software Engineering*, 1977, Vol. SE-3, No. 2, s. 125–143.
- [42] Leveson N. G., Stolzy L.: Safety analysis using Petri nets. *IEEE Transactions on Software Engineering*, 1987, Vol. SE-13, No. 3, s. 386–397.
- [43] Marsan M. A., Balbo G., Conte G., Donatelli S., Franceschinis G.: *Modelling with generalized stochastic Petri nets*. New York, NY, John Wiley and Sons 1995.
- [44] Memmi G., Roucairol G.: Linear algebra in net theory. W: *Proceedings of the Advanced Course on General Net Theory of Processes and Systems*, LNCS, Vol. 84, London, Springer-Verlag 1979, s. 213–223.
- [45] Memmi G., Vautherin J.: Analysing nets by the invariant method. W: *Advances in Petri nets 1986 Part I*, LNCS, Vol. 254, London, Springer-Verlag 1987, s. 300–336.
- [46] Merlin P. M.: A Study of the recoverability of communication protocols. Technical Report 58. University of California, Department of Information and Computer Science, Irvine, CA, 1974.
- [47] Merlin P. M., Faber D. J.: Recoverability of communication protocols – implications of a theoretical study. *IEEE Transactions on Communications*, 1976, Vol. COM-24, No. 7, s. 381–404.
- [48] Murata T.: Petri Nets: properties, analysis and applications. *Proceedings of the IEEE*, 1989, Vol. 77, No. 4, s. 541–580.
- [49] Nigro L., Pupo F.: Using Design/CPN for the schedulability analysis of actor systems with timing constraints. W: *Proceedings of the Workshop on Practical Use of Coloured Petri Nets and Design/CPN*, Aarhus, DK, 1998, s. 271–285.
- [50] Peterson J. L.: *Petri net theory and the modelling of systems*. New York, NY, Prentice Hall 1981.
- [51] Petri C. A.: Communication with Automata. Technical report New York 1965, Supplement 1 to Technical Report RADC-TR-65-377, (łumaczenie z niemieckiej pracy doktorskiej pt.: Kommunikation mit Automaten. University of Bonn 1962).
- [52] Petri C. A.: Introduction to general net theory. W: *Proceedings of the Advanced Course on General Net Theory of Processes and Systems*, LNCS, Vol. 84, London, Springer-Verlag 1979, s. 1–19.

- [53] Popova-Zeugmann L., Werner M.: Determining worst-case times of unknown paths in time Petri nets. W: *Proceedings of the Workshop on Concurrency, Specification and Programming (CS&P'2004)*, Vol. 1, Caputh, DE 2004, s. 144–154.
- [54] Ramchandani C.: Analysis of asynchronous concurrent systems by Petri nets. Technical Report 120. MIT, Laboratory of Computer Science, Cambridge 1974.
- [55] Ravi K., Krishnaiah Chetty O. V.: Priority nets for scheduling flexible manufacturing systems. *Journal of Manufacturing Systems*, 1993. Vol. 12, No. 4, s. 326–340.
- [56] Reisig W.: *Sieci Petriego*. Warszawa, WNT 1988 (tłum. z ang.).
- [57] Roch S., Starke P.: *INA Integrated Net Analyzer*. Humboldt-Universität zu Berlin, Institut für Informatik 1999.
- [58] Ross K. A., Wright C. R. B.: *Matematyka dyskretna*. Warszawa, PWN 1996.
- [59] Sacha K.: Projektowanie oprogramowania systemów wbudowanych. *Prace Naukowe Elektronika*, T. 115, Warszawa, Politechnika Warszawska 1996.
- [60] Samolej S., Szmuc T.: Time extensions of Petri nets for modelling and verification of hard real-time systems. *Computer Science*, 2002, Vol. 4, s. 55–76.
- [61] Schmidt K.: Symmetries of Petri nets. *Petri Net Newsletter*, 1993, No. 43, s. 9–25.
- [62] Schmidt K.: How to calculate symmetries of Petri nets. *Acta Informatica*, 2000, Vol. 36, No. 7, s. 545–590.
- [63] Schmidt K., Starke P.: An algorithm to compute the symmetries of Petri nets. *Petri Net Newsletter*, 1991, No. 40, s. 25–30.
- [64] Shatz S. M., Tu S., Murata T., Duri S.: An application of Petri net reduction for Ada tasking deadlock analysis. *IEEE Transactions on Parallel and Distributed Systems*, 1996, Vol. 7, No. 12, s. 1307–1322.
- [65] Sibertin-Blanc C.: High level Petri nets with data structure. W: *Proceedings of the 6th European Workshop on Application and Theory of Petri Nets*. Espoo, FI, 1985, s. 141–170.
- [66] Sifakis J.: Structural properties of Petri nets. W: *Mathematical Foundations of Computer Science*. LNCS, Vol. 64, Berlin, Springer-Verlag 1978, s. 474–483.
- [67] Sloan R. H., Buy U. A.: Reduction rules for time Petri nets. *Acta Informatica*, 1996, Vol. 33, No. 7, s. 687–706.
- [68] Sloan R. H., Buy U. A.: Stubborn sets for real-time Petri nets. *Formal Methods in System Design*, 1997, Vol. 11, No. 1, s. 23–40.
- [69] Starke P. H.: On the mutual simulability of different types of Petri nets. W: *Advances in Petri Nets – Concurrency and Nets*. Berlin, Springer-Verlag 1987, s. 481–496.
- [70] Starke P. H.: *Sieci Petri. Podstawy, zastosowania, teoria*. Warszawa, PWN 1987.
- [71] Starke P. H.: Some properties of timed nets under the earliest firing rule. W: *Advances in Petri Nets 1989*, LNCS, Vol. 424, Berlin, Springer-Verlag 1990, s. 418–432.
- [72] Starke P. H.: Reachability analysis of Petri nets using symmetries. *Systems Analysis – Modelling – Simulation*, 1991, Vol. 8, s. 294–303.
- [73] Suraj Z., Komarek B.: *Graf – System graficznej konstrukcji i analizy sieci Petriego*. Warszawa, Akademicka Oficyna Wydawnicza PLJ 1994.
- [74] Suraj Z., Szpyrka M.: *Sieci Petriego i PN-tools*. Rzeszów, Wydawnictwo Wyższej Szkoły Pedagogicznej 1999.
- [75] Szmuc T.: Zaawansowane metody tworzenia oprogramowania systemów czasu rzeczywistego. Kraków, CCATIE 1998.

- [76] Szmuc T., Motet G.: Specyfikacja i projektowanie oprogramowania czasu rzeczywistego. KCCATIE 1998.
- [77] Szpyrka M.: Fast and flexible modelling of real-time systems with RTCP-nets. *Computer Systems Conference*, 2004, Vol. 6, s. 81–94.
- [78] Szpyrka M.: Adder Designer – tools for modelling and analysis of rule-based systems. W: *Proceedings of Computer Methods and Systems Conference*, Vol. 1. Kraków, Poland 2005, s. 341–346.
- [79] Szpyrka M.: Analysis of RTCP-nets with reachability graphs. *Fundamenta Informaticae*, Vol. 74, No. 2–3, s. 375–390.
- [80] Szpyrka M., Matyasiuk P.: APE – tool for modelling and analysis of concurrent systems with Petri nets. W: *Proceedings of Computer Methods and Systems Conference*, Vol. 2, Kraków, 2005, s. 49–54.
- [81] Szpyrka M., Szmuc T.: Application of RTCP-nets for design and analysis of embedded systems. W: *Proceedings of the 12th International Conference Mixed Design of Integrated Circuits and Systems*, Kraków, 2005, s. 565–570.
- [82] Szpyrka M., Szmuc, T.: D-nets – Petri net form of rule-based systems. *Foundations of Computing and Decision Sciences*, 2006, Vol. 31, No. 2, s. 157–167.
- [83] Szpyrka M., Szmuc, T.: Verification of automatic train protection systems with RTCP-nets. W: *Proceedings of the 25th International Conference on Computer Safety, Security and Reliability*, Vol. 4166, Berlin, Springer-Verlag 2006, s. 344–357.
- [84] Valmari A.: Stubborn sets of coloured Petri nets. W: *Proceedings of the 12th International Conference on Application and Theory of Petri Nets*, LNCS, Vol. 674, Berlin, Springer-Verlag 1991, s. 10–21.
- [85] Valmari A.: State of the art report: stubborn sets. *Petri Net Newsletter*, 1994, No. 46, s. 6–14.
- [86] Varpaaniemi K.: On computing symmetries and stubborn sets. Technical Report B12, Helsinki University of Technology 1994.
- [87] Varpaaniemi K.: On stubborn sets in the verification of linear time temporal properties. W: *Proceedings of the 19th International Conference on Application and Theory of Petri Nets*, Vol. 1420, Berlin, Springer-Verlag 1998, s. 124–143.
- [88] Walter B.: Timed Petri nets for modelling and analysis protocols with real-time characteristics. W: *Proceedings of the 3rd IFIP Workshop on Protocol Specification, Testing, and Verification*, Amsterdam, North-Holland 1983, s. 149–159.
- [89] Will J., Heiner M.: Petri nets in biology, chemistry, and medicine – bibliography. Technical Report CSR 04/02, BTU Cottbus 2002, s. 1–36.
- [90] Wilson R. J.: *Wprowadzenie do teorii grafów*. Warszawa, PWN 1998.
- [91] Yakovlev A., Gomes L. (red.): *Hardware design and Petri nets*. Norwell, US, Kluwer Academic Publishers 2000.
- [92] Yoneda T., Ryuba H.: CTL model checking of time Petri nets using geometric regions. *IEEE Transactions on Information and Systems*, 1998, Vol. E99-D, No. 3, s. 1–10.
- [93] Zuberek W. M.: Timed Petri nets, definitions, properties, and applications. *Microelectronics and Reliability*, 1991, Vol. 31, No. 4, s. 627–644.

Skorowidz

C

- ciasna pętla 15
- ciąg przejść 23, 142, 169, 210, 226
- cykl 12
- czas początkowy 208
- przejścia 143, 147, 227
- wykonania ciągu przejść 143, 147, 212, 226
- czasowa sieć kolorowana 208
- czysty zbiór miejsc 24, 170

D

- D-sieć 233
- dozór 164
- droga 12
 - nieskierowana 12
 - pusta 12
 - skierowana 12
 - skończona 12
- drzewo klas stanów 158
- osiągalności 52
- pokrycia 60
- dziedzina czasu wykonania 156

F

- funkcja dozorów 164, 208, 220
- kolorów 164, 208, 220
- liniowa 197
- nieujemna 197
- opóźnień 140, 144
- pojemności miejsc 120
- priorytetów 127, 220
- przydziału portów 177
- przydziału stron 177
- typów 164, 208, 220
- typów fuzji 177
- typów portów 177

funkcja wag czasowych 221

- wag łuków 46, 120, 127, 144, 165, 208, 220
- zaczepienia 9, 164, 208
- zastrzeżeń 164, 208, 220
- fuzja globalna 175, 227
- instancji strony 175
- miejsc 175
- stronicowa 175

G

- gniazdo 176, 177, 227
- wejściowe 176
- wejściowo-wyjściowe 176
- wyjściowe 176
- graf 9
 - acykliczny 12
 - dwudzielnny 12
 - etykietowany 11
 - hierarchii stron 175, 235
 - klas 158
 - osiągalności 52, 152, 188, 247
 - osiągalności z klasami równoważności 193
 - pokrycia 60, 253
 - silnie spójny 12
 - silnie spójnych składowych 65
 - skierowany 9
 - skończony 10
 - spójny 12
 - synchronizacji 15
 - zredukowany ze względu na symetrię 72
- grupa symetrii 70

I

- iloczyn wielozbioru przez stałą 163
- instancja strony 176

K

- klasa stanów 156
- komunikacja asynchroniczna 34
 - synchroniczna 33
- konflikt dynamiczny 29, 142, 146, 223
 - statyczny 29
- koniec łuku 10
- krotność łuku 46

L

- liniowo-algebraiczna reprezentacja sieci 84
- lista pieczętek czasowych 206

Ł

- łącznik 234

M

- macierz incydencji 84, 200
 - uzupełniona 87
 - wejść 83
 - wyjść 83
- maksymalna pułapka 100
- maksymalne opóźnienie statyczne 144
- maksymalnie znakowany zbiór miejsc 103
- maksymalny czas przejścia 143, 212, 227
 - wiek dostępu 252
- maszyna stanowa 15, 171
- metoda scalania kolejnych miejsc 111
 - scalania kolejnych przejść 111
 - scalania równoległych miejsc 111
 - scalania równoległych przejść 111
 - usuwania ciasnych pętli miejsc 112
 - usuwania ciasnych pętli przejść 112
- miejsce 13
 - bezpieczne 24, 169
 - gniazdowe 176
 - k-ograniczone 24, 169
 - nieograniczone 24, 169
 - ograniczone 24, 169
 - portowe 176
 - znakowane 24, 170
 - żywe 28, 174
- minimalne opóźnienie statyczne 144
- minimalnie znakowany zbiór miejsc 103
- minimalny czas przejścia 143, 212, 227
 - nieujemny niezmiennik miejsc 86
 - nieujemny niezmiennik przejść 95
 - wiek dostępu 252
- zatrzask 100

- mocna reguła wykonywania przejść 121
- monitor 33
- MS-składowa 18
- multigraf 9

N

- nadstrona 175
- następnik węzła 11
- niezmiennik miejsc 85, 196, 198
 - nieujemny 86, 95
 - przejść 95, 201
 - realizowalny 96, 201
- nośnik niezmiennika 90, 95

O

- ograniczenie dolne 169
 - górne 24, 169
- wielozbiorowe dolne 169
- wielozbiorowe górne 169
- opóźnienie dynamiczne 140, 144
 - statyczne 140, 144

P

- pieczętka czasowa 139, 208
- P*-niezmiennik 85
- początek łuku 10
- początkowy rozkład pieczętek czasowych 221
- podgraf 65
 - wyznaczony przez zbiór węzłów 65
- podgrupa symetrii 70
- podsieć 17, 103
 - jednolita 103
 - łącząca 234
 - nieblokująca 103
 - silnie nieblokująca 103
- wyznaczona przez zbiór miejsc 17
- wyznaczona przez zbiór przejść 17
- podstawiane przejście 175
- podstawowa strona miejsca 228
 - strona przejścia 228, 231
- podstrona 175
- pojemność miejsca 120
- poprzednik węzła 11
- port 176, 177
 - nieokreślony 176
 - wejściowo-wyjściowy 176, 227
 - wejściowy 176, 227
 - wyjściowy 176, 227

- postać bazowa 87
– kanoniczna RTCP-sieci 227
priorytet przejścia 127, 220
proces 31
program sekwencyjny 30
– wspólnieżny 31
prosta sieć czasowa 139, 140
przedziałowa sieć czasowa 139, 144
przejście 13
– aktywne 22, 46, 83, 120, 121, 128, 141, 166,
 209, 222
– jednolite 171
– kluczowe 79
– $\mathcal{L}0$ -żywe 27, 172, 173
– $\mathcal{L}1$ -żywe 27, 172, 173
– $\mathcal{L}2$ -żywe 27, 172, 173
– $\mathcal{L}3$ -żywe 27, 172, 173
– $\mathcal{L}4$ -żywe 27, 172, 173
– martwe 27, 172, 173
– silnie $\mathcal{L}k$ -żywe 173
– wykonalne 141, 145
– zachowawcze 171
– z własnością maszyny stanowej 171
– żywe 27, 172, 173
pułapka 99
punkt czasowy 208
- R**
- relacja mniejszości wielozbiorów 163
– mniejszości wielozbiorów czasowych 207
– pokrywania stanów 253
– przepływu 13
relacyjny graf osiągalności 193
reprezentacja algebraiczna sieci 81
rozkład na maszyny stanowe 18
– pieczętek czasowych 221
rozmiar wielozbioru 163
roszerzona sieć prosta 16
– sieć swobodnego wyboru 16
rozwiązań bazowe 88
równość wielozbiorów 163
równoważna sieć nieczasowa 152
różnica wielozbiorów 163
– wielozbiorów czasowych 207
RTCP-sieć 220
- S**
- sekcja krytyczna 33
semafor 33
- sieciowa reprezentacja niezmiennika 90, 95
sieć 13
– bezpieczna 24, 169
– częściowo zachowawcza 25, 172
– czysta 15, 84
– decyzyjna 233
– dokładnie $\mathcal{L}k$ -żywa 27, 173
– dokładnie silnie $\mathcal{L}k$ -żywa 173
– kolorowana 164
– kolorowana czasowa 208
– kolorowana hierarchiczna 177
– $\mathcal{L}k$ -żywa 27, 173
– miejsc i przejść 119
– odtwarzalna 26, 153, 210, 226
– odwracalna 26, 153, 210, 226
– ograniczona 24, 169
– pokryta niezmiennikami miejsc 91
– pokryta niezmiennikami przejść 96
– pokryta przez podsieci 18
– priorytetowa 127
– prosta 15
– rozkładalna na maszyny stanowe 18
– silnie $\mathcal{L}k$ -żywa 173
– silnie sprawiedliwa 174
– skończona 13
– sprawiedliwa 30, 174
– strukturalnie żywa 27, 173
– swobodnego wyboru 16
– trwała 29
– uogólniona 45
– zachowawcza 25, 171
– zachowawcza względem wektora wag 25, 171
– znakowana 21
– żywa ze względu na miejsca 28, 174
silnie spójna maszyna stanowa 15
– spójna składowa 65
– spójny graf synchronizacji 15
– sprawiedliwy ciąg przejść 174
składowa sieci 18
– trywialna 65
słaba reguła wykonywania przejść 120
specyfikacja równoważności 192
sprawiedliwy ciąg przejść 30, 174
SSMS-składowa 18
stan martwy 143, 210, 223
– odtwarzalny 153, 210, 226
– osiągalny 142, 210, 226
– początkowy 141, 145, 209, 221
– sieci 141, 145, 209, 221

stan własny 153, 210, 226
 strona główna 177
 – sieci 175
 suma wielozbiorów 162
 symetria 69
 synchronizacja procesów 33
 szablon stron 228

S

średnia reguła wykonywania przejść 120

T

tablica decyzyjna 232
T-niezmiennik 95
 tranzycja 13
 trwały zbiór przejść 77, 79
 typ miejsca 164, 208, 220
 – portu 176
 – wyrażenia 164
 – zmiennej 164

U

upływ czasu 209, 223

W

waga łuku 46
 wektor charakterystyczny 84, 94
 – pieczętek czasowych 221
 węzeł grafu 9
 – izolowany 11
 – końcowy 10, 11
 – martwy 52
 – pełny 55
 – początkowy 10, 11
 – stary 70
 węzły równoważne 70
 wiązanie 165, 208, 221
 wielozbior 162
 – czasowy 206
 – stron głównych 177
 – uogólniony 196
 własność maksymalnie kontrolowanych zatrzasków 104
 – minimalnie kontrolowanych zatrzasków 104

wykonanie przejścia 22, 46, 84, 128, 141, 145, 166, 209, 222
 wyrażenie jednolite 170
 wzajemne wykluczanie 33

Z

zachowawczy zbiór wag miejsc 198
 zasoby współdzielone 32
 zastrzeżenie 164
 zatrzask 99
 – maksymalnie kontrolowany 104
 – minimalnie kontrolowany 104
 zbiór etykiet grafu 11
 – fuzji miejsc 177
 – kolorów 164, 208, 220
 – krawędzi grafu 9
 – łuków grafu 9
 – łuków sieci 13, 164, 208, 220
 – miejsc 13, 164, 208, 220
 – następników 11
 – pieczętek czasowych 206, 208
 – podstawianych przejść 177
 – poprzedników 11
 – portów 177
 – przejścia 13, 164, 208, 220
 – punktów czasowych 208
 – stanów osiągalnych 142, 210, 226
 – stron 177
 – typów 164, 208, 220
 – wag miejsc 198
 – węzłów grafu 9
 – znakowań osiągalnych 23, 169
 zgodna specyfikacja równoważności 192
 znacznik 22
 znakowania równoważne 71
 znakowanie 22, 120, 165, 208, 221
 – martwe 27, 173
 – odtwarzalne 26
 – osiągalne 23
 – początkowe 22, 46, 120, 127, 140, 144, 145, 208, 221
 – pokrywalne 60
 – symetryczne 71, 73
 – uogólnione 60
 – własne 26