Opracowanie do kolokwium nr 1		
Metody formalne	16 IV 2019	s. 429 08:00

Spis treści

1 Zagadnienia od Szymkata

1. Definicja sieci Petriego, miejsca, przejścia, znaczniki, warunki wykonania przejścia, stany (znakowania)
2. Własności sieci Petriego: osiągalność, ograniczoność, zachowawczość, żywotność, sprawiedliwość
3. Definicja kolorowanej sieci Petriego, miejsca, przejścia, znaczniki, kolory, opisy łuków, dozory, warunki wykonania przejścia (wiązania), stany (znakowania)
4. Podstawowe konstrukcje w sieci Petriego, licznik, limiter, inhibitor
5. Graf przestrzeni stanów kolorowanej sieci Petriego, elementy, poprzedniki, następniki, stany zakleszczone, znakowanie początkowe, znakowania docelowe (home)
6. Elementy raportu o wygenerowanym grafie stanów w CPNtools, statystyki (węzły, łuki, status, graf komponentów spójnych), ograniczoność, osiągalność, żywotność, sprawiedlwość
7. Modelowanie współdzielenia zasobów w kolorowanej sieci Petriego
8. Indeterminizm w sieciach Petriego
9. Synchronizacja procesów współbieżnych w sieciach Petriego

2 Uwagi

- Podobno definicje formalne są korzystniejsze od ogólnych

3 Źródła

Opracowanie na podstawie:

- Wykłady Szmuca
- Szpyrka, Szmuc - Metody formalne w inżynierii oprogramowania systemów czasu rzeczywistego
- Szpyrka - sieci Petriego w modelowaniu i analizie systemów współbieżnych
- cpntools.org - Anti places/limit places
- cpntools.org - Inhibitor arcs
- `http://home.hib.no/ansatte/lmkr/talks/lmk_statespaces_pn2014.pdf` - całkiem fajny pdf

4 Opracowanie zagadnień

4.1. Zagadnienie 1

Wszystkie opisy i definicje w tym podrozdziale (Zagadnienie 1) dotyczą *Sieci Petriego miejsc i przejść*. (są to sieci różne od sieci kolorowanych).

Zagadnienie 1

Definicja sieci Petriego, miejsca, przejścia, znaczniki, warunki wykonania przejścia, stany (znakowania).

4.1.1. Definicja sieci Petriego

Definicja - Sieć Petriego (miejsc i przejść)

Siecią Petriego nazywamy piątkę, $PM = (P, T, A, W, s_0)$, gdzie:

- P - zbiór (skończony) miejsc
- T - zbiór (skończony) przejść
- A $\subseteq P \times T \cup T \times P$ - zbiór łuków
- W: $A \rightarrow N$ - funkcja wag przypisująca etykiety (liczby naturalne) do każdego łuku
- $s_0 : P \rightarrow N^*$ - funkcja opisująca oznakowanie początkowe, gdzie N^* oznacza zbiór liczb całkowitych nieujemnych

Ponadto dla każdej sieci Petriego spełniony jest warunek:

$$P \cap T = \emptyset \wedge P \cup T \neq \emptyset$$

4.1.2. Miejsca

Miejsca to niepusty zbiór $P = \{p_1, p_2, \ldots, p_n\}$. Miejsca są graficznie reprezentowane jako elipsy. Miejsca mogą być połączone łukami jedynie z tranzycjami. Miejscom przyporządkowane są żetony.

4.1.3. Przejścia (tranzycje)

Przejścia to niepusty zbiór $T = \{t_1, t_2, \ldots, t_n\}$. Przejścia są graficznie reprezentowane jako prostokąty. Przejścia mogą być połączone łukami jedynie z miejscami. Przejście może być aktywne lub nieaktywne.

4.1.4. Znaczniki

Znaczniki to liczby całkowite przyporządkowane miejscom, które określają stan sieci i oznaczają liczbę żetonów znajdującą się w danym miejscu.

4.1.5. Warunki wykonania przejścia (w sieciach miejsc i przejść)

Przejście może być wykonane gdy jest aktywne

Definicja - Przejście aktywne - Sieć miejsc i 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 oraz każde z jego miejsc wyjściowych ma wystarczającą pojemność, by przyjąć tyle znaczników, ile wynosi waga łuku (t, p) tzn.:

$$\forall p \in In(t) : M(p) \geq W(p, t) \wedge$$
$$\forall p \in Out(t) : M(p) + W(t, p) \leq K(p)$$

4.1.6. Znakowanie

Definicja - Znakowanie

Znakowaniem sieci miejsc i przejść nazywamy dowolną funkcję

$$M : P \to Z^+ \text{ , taką że } \forall p \in P : M(p) \leq K(p)$$

$K(p)$ to funkcja określająca 'pojemność' maksymalną miejsc.

- Podaj definicję sieci Petriego (miejsc i przejść).
- Podaj warunki wykonania przejścia w sieciach Petriego
- Podaj definicję znakowania

4.2. Zagadnienie 2

Zagadnienie 2

Własności sieci Petriego: osiągalność, ograniczoność, zachowawczość, żywotność, sprawiedliwość

4.2.1. Osiągalność

Definicja - Osiągalność

Niech będzie dana sieć (PN, s_0). Mówimy, że stan s jest osiągalny ze stanu s_0 wtedy i tylko wtedy jeśli $s \in R(PN, s_0)$.
gdzie $R(PN, s_0)$ oznacza zbiór wszystkich stanów osiągalnych ze stanu początkowego s_0.

$R(PN, s_0) = Ran(\rho^*) = Ran(\cup_{n=0}\rho^n)$ - to przeciwdziedzina iteracji relacji bezpośredniego następstwa stanów ρ w sieci Petriego.

4.2.2. Ograniczoność

Definicja - Ograniczoność

Znakowana sieć (PN, s_0) jest k-ograniczona, jeśli przy dowolnym osiągalnym oznakowaniu w każdym miejscu ilość znaczników nie przekracza skończonej liczby k,
tzn, dla dowolnego $s \in R(PN, s_0)$ oraz dla każdego miejsca $p \in P$ spełniony jest warunek $s(p) \leq k$.

gdzie $R(PN, s_0)$ oznacza zbiór wszystkich stanów osiągalnych ze stanu początkowego s_0.

$R(PN, s_0) = Ran(\rho^*) = Ran(\cup_{n=0}\rho^n)$ - to przeciwdziedzina iteracji relacji bezpośredniego następstwa stanów ρ w sieci Petriego.

Znakowana sieć $PM = (PN, s_0)$ jest ograniczona jeśli jest k-ograniczona dla pewnego skończonego k

4.2.3. Zachowawczość

Definicja - Zachowawczość

Znakowana sieć (PN, s_0) jest (bezwzględnie) zachowawcza jeśli dla łączna liczba znaczników w sieci pozostaje stała dla każdego znakowania osiągalnego ze znakowania początkowego,
tzn. dla dowolnego stanu $s \in R(PN, s_0)$ spełniony jest warunek

$$\sum_{p \in P} s(p) = \sum_{p \in P} s_0(p)$$

gdzie $R(PN, s_0)$ oznacza zbiór wszystkich stanów osiągalnych ze stanu początkowego s_0.

$R(PN, s_0) = Ran(\rho^*) = Ran(\cup_{n=0}\rho^n)$ - to przeciwdziedzina iteracji relacji bezpośredniego następstwa stanów ρ w sieci Petriego.

Znakowana sieć (PN, s_0) jest zachowawcza względem wektora wag $(w_1, w_2, \ldots, w_n)$ jeżeli łączna ważona liczba znaczników w sieci pozostaje stała dla każdego znakowania osiągalnego ze znakowania początkowego
tzn. dla dowolnego stanu $s \in R(PN, s_0)$ spełniony jest warunek

$$\sum_i w_i s(p_i) = \sum_i w_i s_0(p_i)$$

4.2.4. Żywotność

Żywotność określa wykonywalność pewnych tranzycji w czasie 'wykonywania' sieci. Własności te są różne w zależności od tego, które z przejść i jak często są wykonywane dlatego konieczne jest rozbicie definicji na 5 przypadków.

Definicja - Żywotność

Niech dana będzie sieć Petriego (PN, s_0). Przez $L(s_0)$ oznaczamy zbiór wszystkich ciągów przejść, które można wykonać rozpoczynając od stanu s_0.
Mówimy, że dowolne przejście $t \in T$ jest:

- L0-żywotne (martwe) - jeśli to przejście nie może być wykonane dla każdej sekwencji odpaleń $L(s_0)$
- L1-żywotne (potencjalnie odpalalne) jeśli może być wykonane przynajmniej raz dla pewnej sekwencji odpaleń $L(s_0)$
- L2-żywotne jeśli to przejście może być wykonane przynajmniej k razy dla pewnej sekwencji odpaleń $L(s_0)$
- L3-żywotne jeśli t występuje nieskończenie wiele razy w pewnej sekwencji odpaleń $L(s_0)$
- L4-żywotne (żywotne) jeśli t jest L1-żywotne w każdym oznakowaniu osiągalnym $s \in R(PN, s_0)$

gdzie $R(PN, s_0)$ oznacza zbiór wszystkich stanów osiągalnych ze stanu początkowego s_0.

4.2.5. Sprawiedliwość

Definicja - Sprawiedliwość

Niech dana będzie sieć Petriego (PN, s_0). Niech σ oznacz sekwencję wykonań przejść.

- Sekwencja wykonań σ sieci PM jest (globalnie) sprawiedliwa jeśli jest skończona lub każde przejście występuje nieskończenie wiele razy w σ
- Sieć PM jest (globalnie) sprawiedliwa jeśli każda sekwencja wykonań tej sieci jest globalnie sprawiedliwa.

- Podaj definicję osiągalności w sieciach Petriego.
- Podaj definicję ograniczoności w sieciach Petriego.
- Podaj definicję zachowawczości w sieciach Petriego.
- Podaj definicję żywotności w sieciach Petriego.
- Podaj definicję sprawiedliwości w sieciach Petriego.

4.3. Zagadnienie 3

Wszystkie opisy i definicje w tym podrozdziale (Zagadnienie 3) dotyczą *Kolorowanych Sieci Petriego (CPN)*. (są to sieci różne od sieci miejsc i przejść - Zagadnienie 1).

Zagadnienie 3

Definicja kolorowanej sieci Petriego, miejsca, przejścia, znaczniki, kolory, opisy łuków, dozory, warunki wykonania przejścia (wiązania), stany (znakowania)

4.3.1. Definicja kolorowanej sieci Petriego (CPN)

Definicja - Sieć kolorowana

Siecią kolorowaną nazywamy krotkę

$$CPN = (\Sigma, P, T, A, N, C, G, E, I)$$

gdzie:

- Σ - niepusty, skończony, zbiór typów zwany zbiorem kolorów
- P - skończony zbiór miejsc
- T - skończony zbiór przejść
- A - skończony zbiór łuków, taki, że $P \cap T = P \cap A = T \cap A = \emptyset$
- $N : A \to (P \times T) \cup (T \times P)$ - funkcja zaczepienia przypisująca każdemu łukowi uporządkowaną parę węzłów
- $C : P \to \Sigma$ - jest funkcją typów (kolorów), która określa, jakiego typu znaczniki każde z miejsc może zawierać
- G - funkcja dozorów (zastrzeżeń), każdemu z przejść przypisuje wyrażenie (tzw. dozór), które może zawierać zmienne typów należących do Σ i którego dowolne wartościowanie daje w wyniku wartość logiczną
- E - funkcja wag łuków, przypisuje każdemu z łuków wyrażenie, które może zawierać zmienne należące do Σ i którego dowolne wartościowanie daje w wyniku wielozbiór nad typem przypisanym do danego miejsca
- I - funkcja inicjalizacji, która każdemu miejscu przyporządkowuje wielozbiór nad typem przypisanym do tego miejsca.

4.3.2. Miejsca

W CPN każde miejsce musi mieć przypisany kolor, który określa jakiego typu znaczniki to miejsce może posiadać.

4.3.3. Przejścia

W CPN przejściom przypisywane są wyrażenia, które zawierają zmienne należące do Σ i których dowolne wartościowanie daje w wyniku wartość logiczną

4.3.4. Parę pojęć pomocniczych dla dalszych definicji

- $Var(expr)$ - zbiór zmiennych występujących w wyrażeniu
- $Type(v)$ - typ zmiennej v
- $expr < b >$ - wartość wyrażenia expr przy wartościowaniu b, np. $G < b >$ to wartość dozoru przy wartościowaniu b
- $b(v)$ - wiązanie zmiennej v

4.3.5. Wiązanie przejścia

Wiązaniem przejścia $t \in T$ nazywamy odwzorowanie b takie, że każdej zmiennej przypisano wartość należącą do typu tej zmiennej i jednocześnie jest spełnione zastrzeżenie(dozór) przejścia t.
tzn.

$$\forall v \in Var(t) : b(v) \in Type(v) \wedge G(t) < b >$$

Zbiór wszystkich wiązań dla t oznaczamy $B(t)$.

4.3.6. Element znacznikowy

Jest to dowolna para (p, c) składająca się z koloru i miejsca.

4.3.7. Element wiązania

To para (t, b), składająca się z przejścia i wiązania przejścia, gdzie $t \in T$ oraz $b \in B(t)$

4.3.8. Zbiory TE oraz BE

Zbiór wszystkich elementów znacznikowych oznaczamy TE.
Zbiór wszystkich elementów wiązań oznaczamy BE.

4.3.9. Oznakowanie

W sieciach kolorowanych oznakowaniem jest odwzorowanie, które każdemu miejscu przypisuje wielozbiór nad TE (TE to zbiór wszystkich elementów znacznikowych).

$$\forall p \in P : M(p) \in 2^{C(p)^*}$$

4.3.10. Warunki wykonania przejścia (w sieciach CPN)

W sieciach kolorowanych krokiem jest niepusty i skończony wielozbiór nad BE (BE to zbiór wszystkich elementów wiązań), tj. wielozbiór nad wartościami spełniającymi warunek dozoru.
Przejście $t \in T$ jest aktywne (czyli dowolny krok jest wzbudzony) przy oznakowaniu M wtedy i tylko wtedy gdy spełniony jest warunek:

$$\forall p \in In(t) : E(p, t) < b > \ \leq M(p)$$

tzn. jeżeli każde miejsce wejściowe przejścia t zawiera przy wiązaniu b odpowiednią liczbę i odpowiednie wartości znaczników.

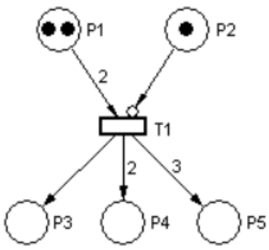
- Podaj definicję CPN.
- Podaj definicję oznakowania w CPN.
- Podaj warunki wykonania przejścia w CPN.

4.4. Zagadnienie 4

Zagadnienie 4
Podstawowe konstrukcje w sieci Petriego, licznik, limiter, inhibitor

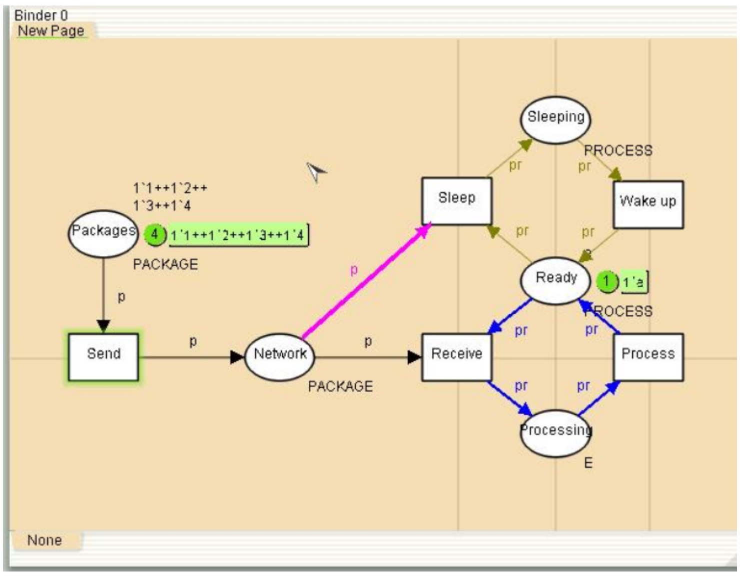
4.4.1. Inhibitor

Łukiem *inhibior* w sieciach Petriego nazywany jest łuk, który łączy miejsce z przejściem, ale którego logika jest odwrotna do normalnego działania sieci Petriego. Przejście jest aktywne wtedy gdy w miejscu wejściowym nie ma znacznika, a nie wtedy gdy tam jest. Przykład



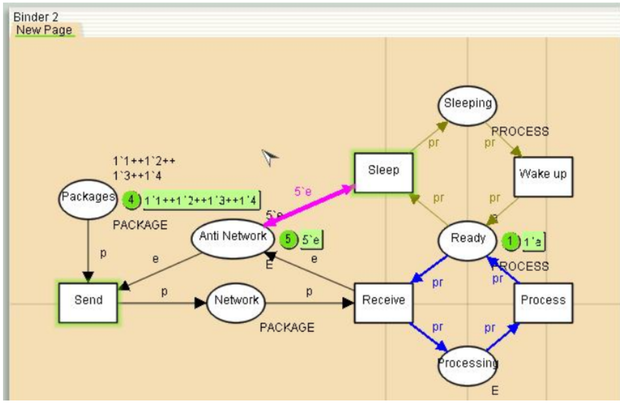
Rysunek 1: Przykładowy inhibitor

Na Rysunku 1 przejście T1 jest aktywne wtedy gdy w miejscu P2 nie ma żetonu, a nie gdy tam jest. Przykładowo w sytuacji pokazanej na Rysunku poniżej chcielibyśmy mieć inhibtor w miejscu łuku różowego, po to aby konsumer mógł iść spać jeśli nic nie ma do odebrania z sieci. (Przejście sleep powinno być aktywne gdy w miejscu Network nie ma żadnego żetonu.



Rysunek 2: Przykład gdzie przydatne jest wykorzystanie łuku Inhibitor

W CPN Tools nie ma takiego łuku jak Inhibitor dlatego trzeba radzić sobie pewnymi konstrukcjami zastępczymi, które działają tak jak łuk inhibitor.
Inhibitor można zaimplementować przy pomocy tzw. anty-miejsca. Polega to na dodaniu miejsca dla którego liczba żetonów zwiększa się wraz z przybywaniem ich wiadomości do sieci, a gdy wszystkie wiadomości zostały przesłane to wówczas aktywne jest przejście Sleep. Przykład:



Rysunek 3: Implementacja inhibitora z wykorzystaniem anty-miejsca

Inhibitor można także zaimplementować z wykorzystaniem licznika.

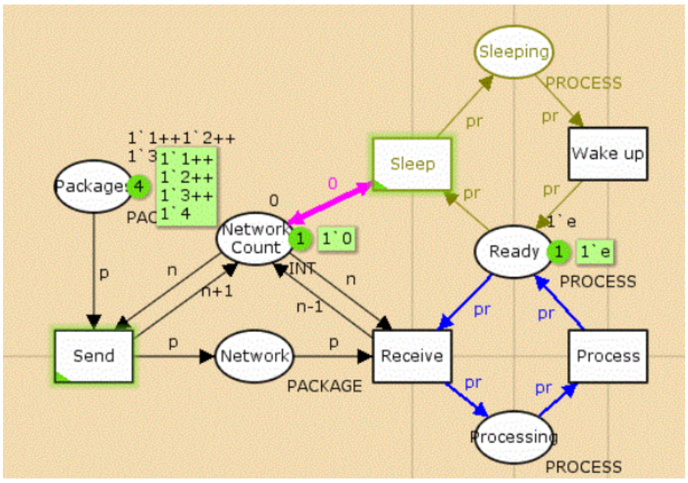
- Czym jest inhibitor ?
- Jak zamodelować inhibitor przy użyciu anty-miejsca ?

4.4.2. Licznik

Jest to osobne miejsce, które oblicza liczbę zajść pewnego zdarzenia w sieci. Charakterystyczne dla licznika jest to, że z każdą tranzycją związane są dwa łuki

- Wchodzący do tranzycji od licznika - daje informację o aktualnej wartości licznika, zmienna n
- Wychodzący od tranzycji do licznika - zmniejsza/zwiększa licznik w wyniku odpalenie jakiegoś przejścia, np. n+1 lub n-1

Przykład

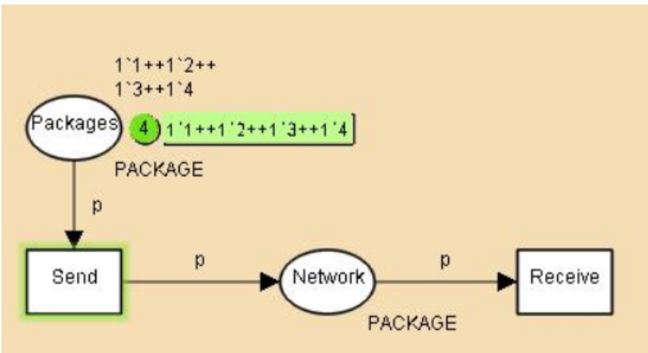


Rysunek 4: Implementacja inhibitora przy użyciu licznika

- Czym jest licznik?
- Jak zamodelować licznik w sieciach Petriego ?

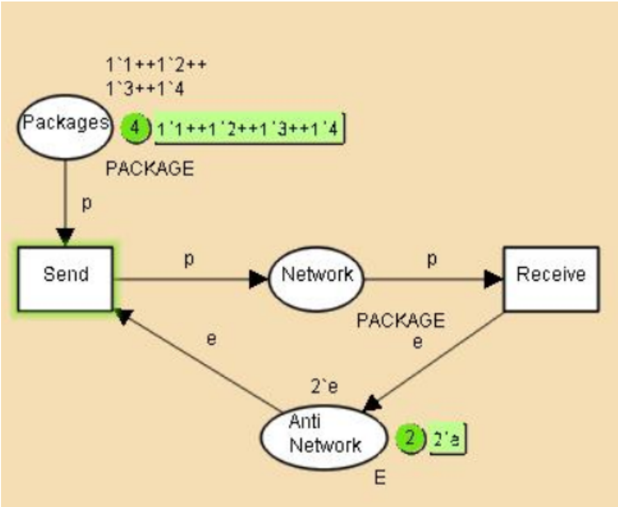
4.4.3. Limiter

Limiter do konstrukcja, która ma umożliwić ograniczenie liczb przechowywanych żetonów w danym miejscu. Np. na Rysunku poniżej mamy sieć i chcemy aby w sieci znajdywały się maksymalnie dwa pakiety (dwa żetony).



Rysunek 5: Sieć gdzie potrzebny jest limiter

Można to osiągnąć przez wprowadzenie anty-miejsca, które będzie działać na tranzycję send w ten sposób, że dopiero po odebraniu dwóch pakietów możliwe będzie dalsze wysyłanie.



Rysunek 6: Przykład implementacji limitera z wykorzystaniem anty-miejsca.

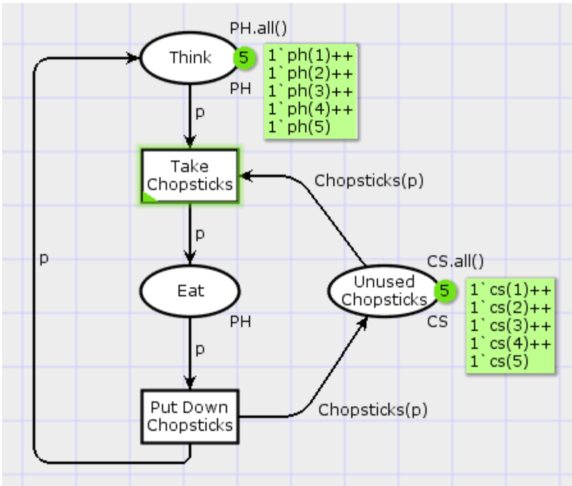
- Czym jest limiter ?
- Jak zamodelować limiter w sieciach Petriego ?

4.5. Zagadnienie 5

Zagadnienie 5

Graf przestrzeni stanów kolorowanej sieci Petriego, elementy, poprzedniki, następniki, stany zakleszczone, znakowanie początkowe, znakowania docelowe (home)

To zagadnienie omówione zostanie na przykładzie 5 filizofów z CPN Tools.



Rysunek 7: Pięciu filozofów

Dla tego grafu generujemy sobie raport i dostajemy jako pierwsze:

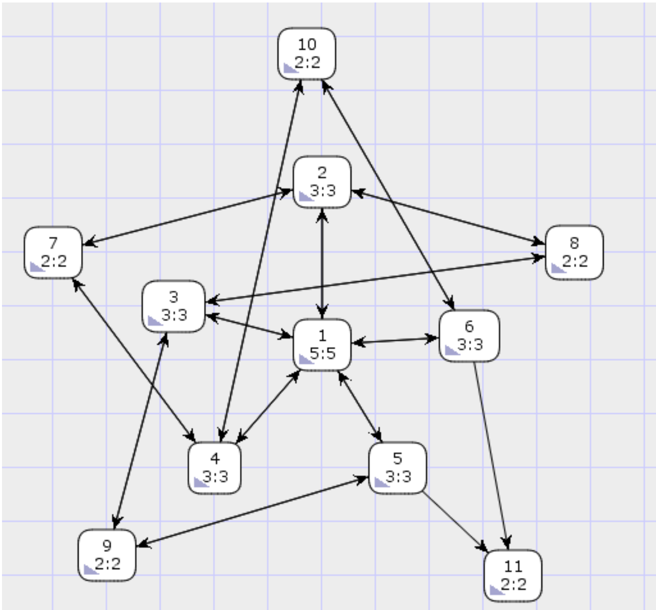
```
Statistics
------------------------------------------------------------------------

 State Space
    Nodes:  11
    Arcs:   30
    Secs:   0
    Status: Full

 Scc Graph
    Nodes:  1
    Arcs:   0
    Secs:   0
```

Rysunek 8: Pięciu filozofów - statystyki z raportu

4.5.1. Graf przestrzeni stanów

W tym zagadnieniu interesuje nas graf przestrzeni stanów (State space), SCC omówione będzie w następnym zagadnieniu.

Jak widać mamy 11 węzłów, 30 łuków, czas(jakiś bardzo krótki) i status Full. Dane te odpowiadają strukturze grafu przestrzeni stanów, który dla 5 filozofów wygląda tak:



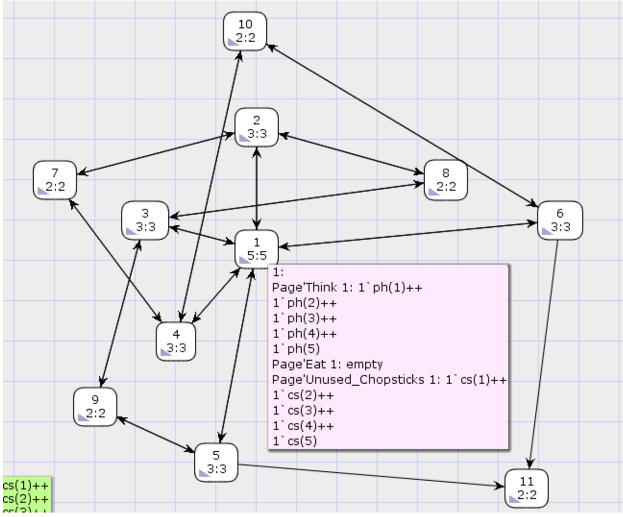
Rysunek 9: Graf przestrzeni stanów dla 5 filozofów

4.5.2. Węzły, łuki, status

Jak widać jest na nim rzeczywiście 11 węzłów i 30 łuków (liczymy łuk dwustronny jako 2), o tym właśnie mówi statistics. Status full oznacza, że wszystkie możliwe znakowania dla tej sieci zostały osiągnięte w symulacji.
Pojedynczy węzeł reprezentuje unikalne znakowanie sieci, czyli np. węzeł o numerze 1 reprezentuje stan w którym:

- w miejscu Think jest 5 rzetonów
- w miejscu Unused chopsticks jest 5 rzetonów
- w miejscu PutDown nie ma ani jednego żetona

Pojedynczy węzeł zawiera u góry liczbę porządkową węzła (Uwaga: znakowanie początkowe to węzeł o numerze 1), w lewym dolnym rogu liczbę poprzedników, a prawym dolnym liczbę następników. Skąd wiadomo, że ten węzeł to właśnie takie znakowanie ? Wystarczy najechać na strzałkę w lewym dolnym rogu i dostaniemy dokładną informację:

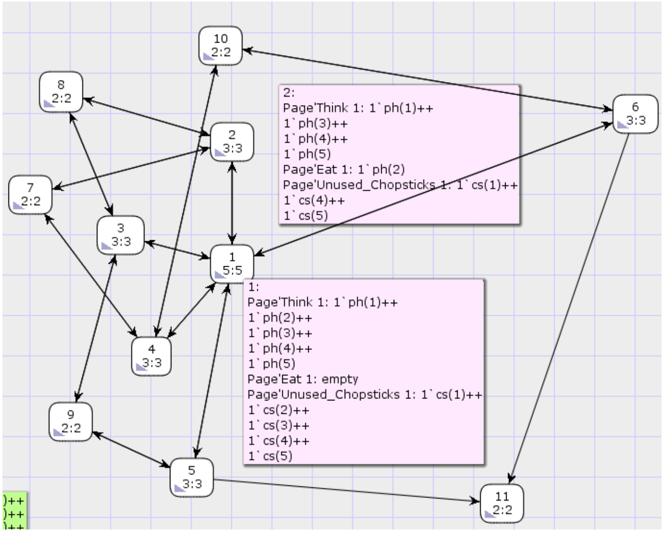


Rysunek 10: Każdy węzeł odpowiada innemu, unikalnemu znakowaniu w grafie przestrzeni stanów

- Czym jest State Space (graf przestrzeni stanów) ?
- Czym jest pojedynczy węzeł w grafie przestrzeni stanów ?
- Co oznaczają nodes, arcs, status w raporcie dla State Space ?

4.5.3. Poprzednicy, następnicy

Liczba następników to liczba łuków wychodzących i oznacza ona liczbę znakowań(a więc i węzłów), które możliwe są po wykonaniu tranzycji (którejkolwiek z aktywnych). W tym przypadku można np. przejść z węzła 1 do węzła 2, co oznacza, że jeden z filozofów zaczął jeść. Następnicy to zatem wszystkie możliwe stany które mogą nastąpić po wykonaniu którejkolwiek aktywnej tranzycji w danym stanie, a *na grafie następnicy węzła to węzły, do których prowadzi łuk od tego węzła.*



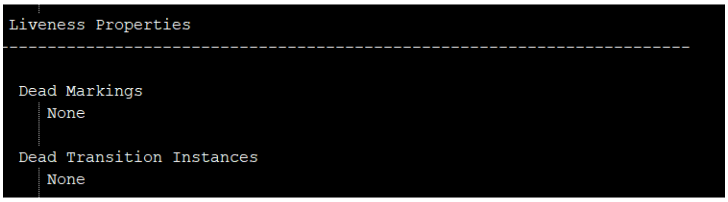
Rysunek 11: Przejście z węzła 1 do 2, to zmiana znakowania po wykonaniu tranzycji.

Poprzednicy danego węzła oznaczają wszystkie znakowania z których osiągalny jest dany stan, a *na grafie to wszystkie węzły z których prowadzi łuk do danego węzła.*

- Czym są następnicy ? Jak rozpoznać ich na grafie stanów ?
- Czym są poprzednicy ? Jak rozpoznać ich na grafie stanów ?

4.5.4. Stany zakleszczone

W raporcie uzyskujemy też, że



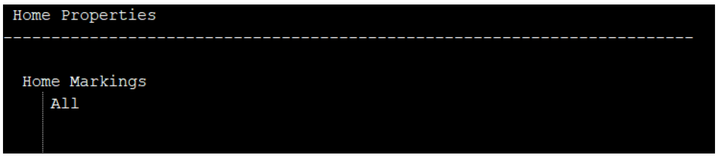
Rysunek 12: Dead markings w raporcie dla 5 filozofów.

Dead markings są to oznakowania z których nie ma przejścia do innego znakowania. *Na grafie stanów można je rozpoznać jako węzły, które nie mają następników.* Z jednej strony mogą to być zakleszczenia w systemie. Z drugiej jeśli nasza symulacja 'nie kręci się w kółko' to może być po prostu stan w którym się zakończyła i nie można iść już do żadnego następnego stanu.
Dead transition to przejścia, które nie pojawiają się w grafie stanów.

- Czym są dead markings (zakleszczenia) ? Jak rozpoznać je na grafie stanów ?

4.5.5. Znakowanie docelowe (home), znakowanie początkowe

Home marking to nie jest znakowanie początkowe !
Home marking oznacza znakowanie, które jest osiągalne z jakiegokolwiek osiągalnego znakowania. Tzn. będąc w danym stanie, zawsze znajdzie się sekwencja przejść prowadząca do home marking. Dla pięciu filozofów mamy po prostu home markings all, ponieważ z dowolnego znakowania można przejść do dowolnego innego po pewnej liczbie wykonanych tranzycji. *Na grafie widać to w taki sposób, że biorąc dowolny węzeł zawsze znajdziemy drogę po łukach do home marking.*



Rysunek 13: Home markings w raporcie dla 5 filozofów.

W raporcie nie ma czegoś takiego jak znakowanie początkowe, ale chyba chodzi po prostu o... znakowanie początkowe... i o to, że na grafie ma węzeł numer 1.

- Czym jest znakowanie docelowe (home marking) ? Jak rozpoznać takie znakowanie na grafie stanów ?
- Jak rozpoznać węzeł oznakowania początkowego na grafie stanów ?

4.6. Zagadnienie 6

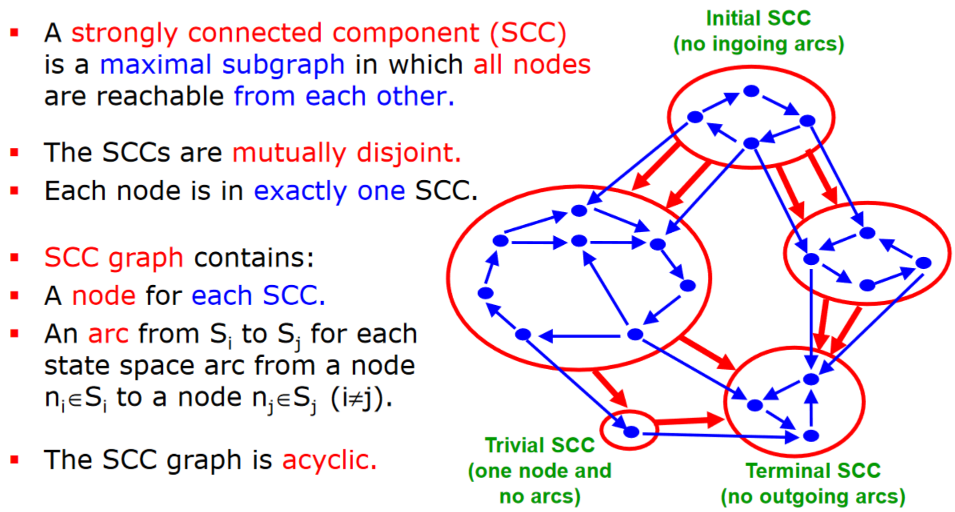
Zagadnienie 6

Elementy raportu o wygenerowanym grafie stanów w CPNtools, statystyki (węzły, łuki, status, graf komponentów spójnych), ograniczoność, osiągalność, żywotność, sprawiedlwość

Węzły i łuki dla grafu stanów oraz status omówione zostały w zagadnieniu 5.

4.6.1. Graf komponentów spójnych

Jest to swego rodzaju 'nakładka' na graf stanów taki jak na Rysunku 9. Najpierw trzeba wiedzieć czym jest komponent spójny.
Komponentem spójnym jest maksymalny pograf w którym dowolny stan może być osiągnięty z dowolnego innego stanu. Przykład na rysunku poniżej:



Rysunek 14: Przykładowy graf scc.

Kolorem czerwonym zaznaczone są węzły oraz łuki grafu SCC. Dla 5 filozofów otrzymano w raporcie:


```
Statistics
------------------------------------------------------------------------

 State Space
    Nodes:  11
    Arcs:   30
    Secs:   0
    Status: Full

 Scc Graph
    Nodes:  1
    Arcs:   0
    Secs:   0
```

Rysunek 15: SCC raport dla philo.

- Czym jest graf komponentów spójnych ? Co oznaczają węzły w tym grafie ?
- Co oznaczają arcs, nodes dla grafu SCC w raporcie ?

4.6.2. Ograniczoność

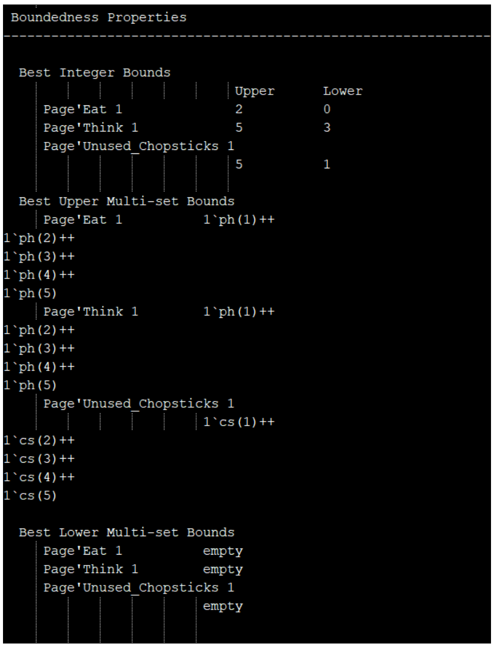
Integer Bounds to wskaźnik określający liczbę rzetonów w danym miejscu. Wskaźnik ten wyrażany jest liczbą całkowitą.

- Best upper integer bound - to maksymalna liczba rzetonów w danym miejscu w pewnym osiągalnym znakowaniu
- Best lower integer bound - to minimalna liczba rzetonów w danym miejscu w pewnym osiągalnym znakowaniu

Multiset Bounds to wskaźnik określający liczbę wystąpień danego (konkretnego) rzetona w danym miejscu. Wskaźnik ten wyrażany jest przez wielozbiór.

- Best upper multiset bounds - określa ile maksymalnie konkretnych rzetonów może znaleźć się w danym miejscu podczas działania systemu
- Best lower multiset bounds - określa ile minimalnie konkretnych rzetonów może znaleźć się w danym miejscu podczas działania systemu

W przykładzie poniżej mamy np. że jeśli weźmiemy rzeton ph(1) (filozof numer 1), to może być maksymalnie jeden taki w każdym z miejsc podczas symulacji (informuje o tym liczba 1 przed ph(1)) - ważne są współczynniki przed elementami wielozbioru.
Również biorąc ten sam żeton widzimy, że minimalna liczba jego wystąpień w danym miejscu podczas symulacji wynosi 0 (empty).



```
Boundedness Properties
------------------------------------------------------------

  Best Integer Bounds
                              Upper      Lower
     Page'Eat 1               2          0
     Page'Think 1             5          3
     Page'Unused_Chopsticks 1
                              5          1

  Best Upper Multi-set Bounds
     Page'Eat 1           1`ph(1)++
1`ph(2)++
1`ph(3)++
1`ph(4)++
1`ph(5)
     Page'Think 1         1`ph(1)++
1`ph(2)++
1`ph(3)++
1`ph(4)++
1`ph(5)
     Page'Unused_Chopsticks 1
                          1`cs(1)++
1`cs(2)++
1`cs(3)++
1`cs(4)++
1`cs(5)

  Best Lower Multi-set Bounds
     Page'Eat 1           empty
     Page'Think 1         empty
     Page'Unused_Chopsticks 1
                          empty
```

Rysunek 16: Właściwości ograniczoności dla 5 filozofów w raporcie.

- Czym są integer bounds ? Co oznaca Best upper integer bound, a co best lower integer bound ?
- Czym są multiset bounds ? Co oznacza best upper multiset bounds, a co lower multiset bounds ?

4.6.3. Osiągalność

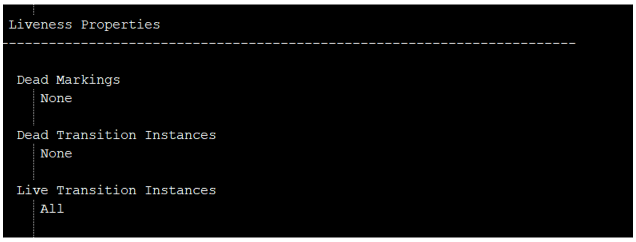
Raport nie zawiera informacji nt. osiągalności (czyli czy z danego węzła można dojść po łukach do innego), ale można ją łatwo sprawdzić korzystając z grafu komponentów spójnych sprawdzając czy istnieje ścieżka pomiędzy węzłem zawierającym znakowanie startowe i końcowe. Jeśli natomiast w statystykach wygeneruje się, że graf SCC ma więcej niż jeden węzeł oznacza to, że nie wszystkie stany są wzajemnie osiągalne, czyli nie prawdą jest, że biorąc dowolne dwa stany można dojść z jednego do drugiego i na odwrót.
Jeśli graf SCC ma 1 węzeł (jak w przypadku filozofów), to wtedy wszystkie znakowania są wzajemnie osiągalne.

- Czym jest osiągalność ? Jak sprawdzić czy wszystkie stany są wzajemnie osiągalne ?

4.6.4. Żywotność

Dead marking to znakowanie przy którym nie ma aktywnego ani jednego przejścia. W raporcie jest ono oznaczane liczbami np: [4356], wtedy oznacza to, że dead marking jest reprezentowane w grafie stanów

przez węzeł numer 4356.
Dead transitions to tranzycja, która nigdy nie jest aktywna
Live tansition to tranzycja dla której wybierając dowolne osiągalne oznakowanie można znaleźć taki ciąg przejść, że zawarta w nim będzie ta tranzycja. Jeśli mamy choćby jedno dead marking, to na pewno live transition będzie miało wartość None.



Rysunek 17: Właściwości żywotności dla 5 filozofów w raporcie.

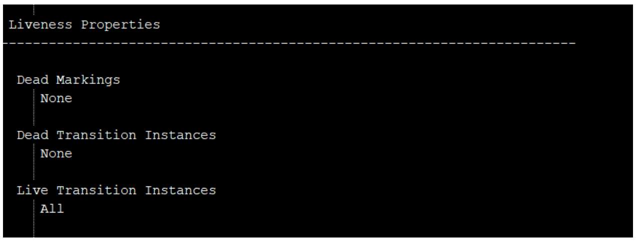
- Czym jest dead marking ? Jak jest oznaczane ?
- Czym jest dead transition ?
- Czym jest live transition ?

4.6.5. Sprawiedliwość

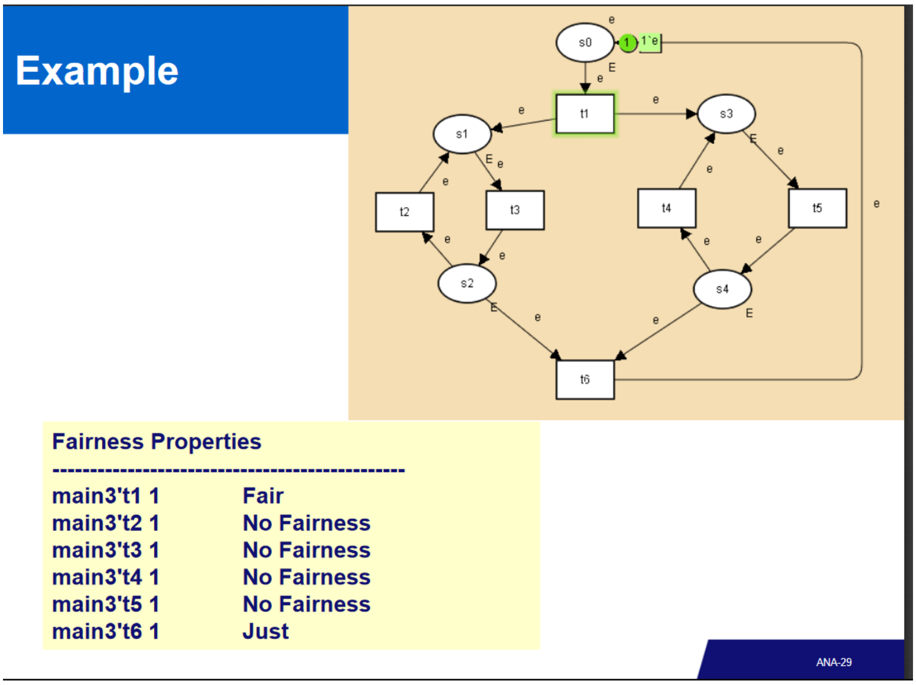
Impartial transition to tranzycja, która pojawia się w każdej nieskończonej sekwencji wykonywania tranzycji nieskończenie wiele razy.
Fair Transtion Instances to tranzycje, które pojawiają się nieskończenie wiele razy, ale tylko w niektórych sekwencjach wykonywania tranzycji (...czyli tylko w tych w których występuje nieskończenie wiele razy)
Just Transition Instances to tranzycje, które pojawiają się nieskończenie wiele razy, ale tylko w tych sekwencjach wykonywania tranzycji w których są ciągle aktywne od pewnego momentu
No fairness przeciwieństwo just, czyli tranzycja dla której istnieje taka sekwencja wykonywania przejść, że tranzycja t jest aktywna od pewnego momentu w czasie, ale nie jest nigdy więcej wykonywana

- Czym jest impartial transition ?
- Czym jest fair transition instances ?
- Czym jest just tranistion instances ?
- Czym jest no fairness ?

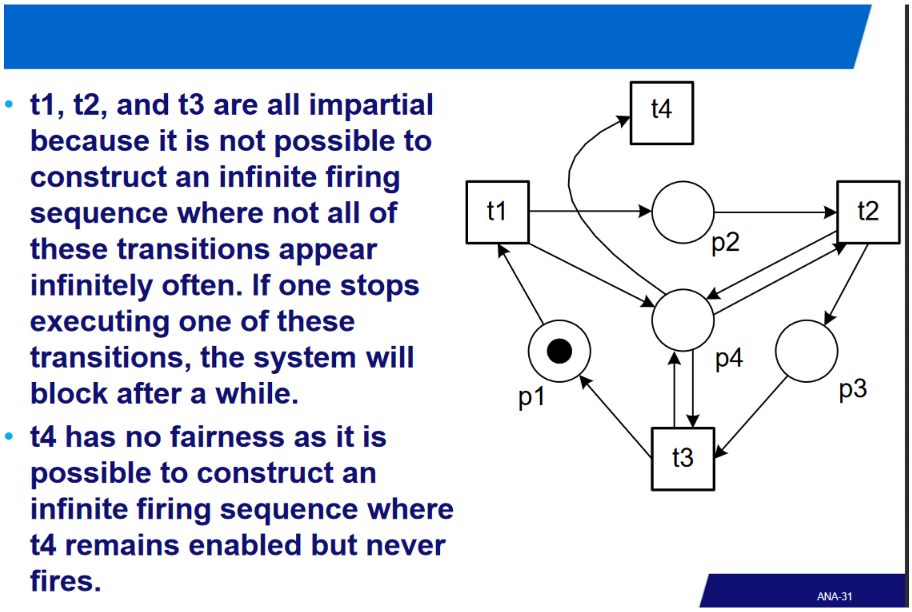
Przykłady:



Rysunek 18: Właściwości żywotności dla 5 filozofów w raporcie.



Rysunek 19: Przykłady własności sprawiedliwości.

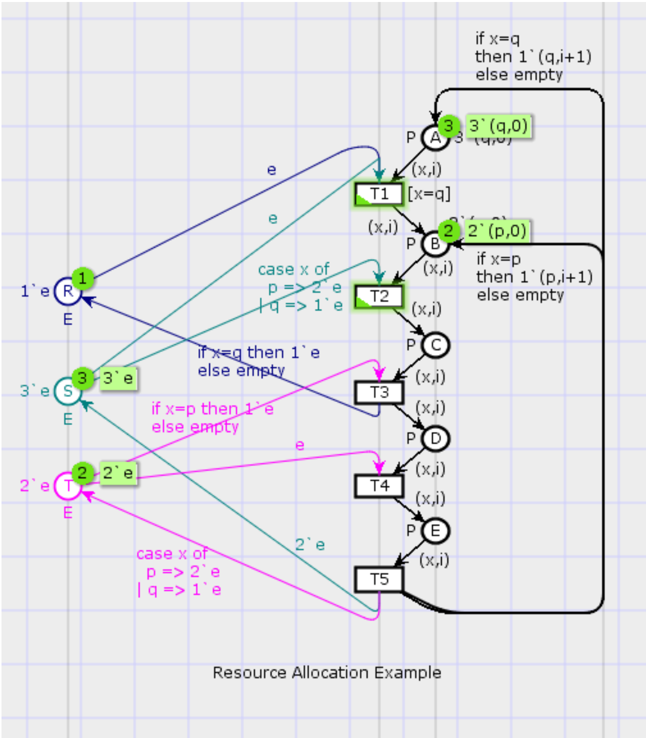


Rysunek 20: Przykłady własności sprawiedliwości.

4.7. Zagadnienie 7

Zagadnienie 7
Modelowanie współdzielenia zasobów w kolorowanej sieci Petriego

(?) Np. coś jak resource allocation, miejsca mają żetony, które reprezentują zasoby. (?)



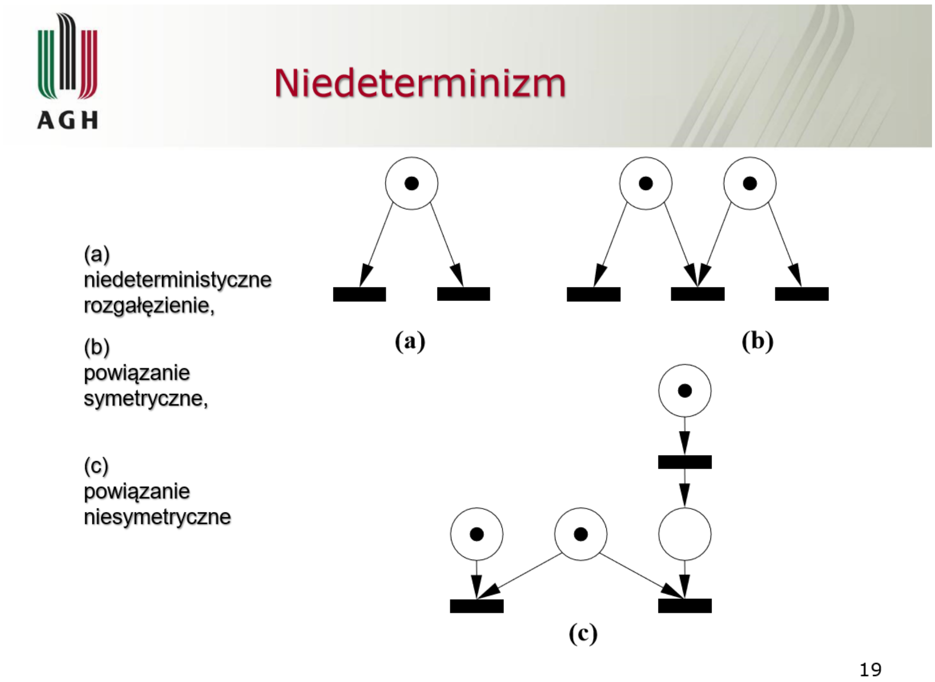
Rysunek 21: Przykład współdzielenie zasobów przez kilka wątków w CPN.

4.8. Zagadnienie 8

Zagadnienie 8

Indeterminizm w sieciach Petriego

(?) Jeśli mamy kilka aktywnych przejść, to nie wiemy, które z nich się wykona. (?)

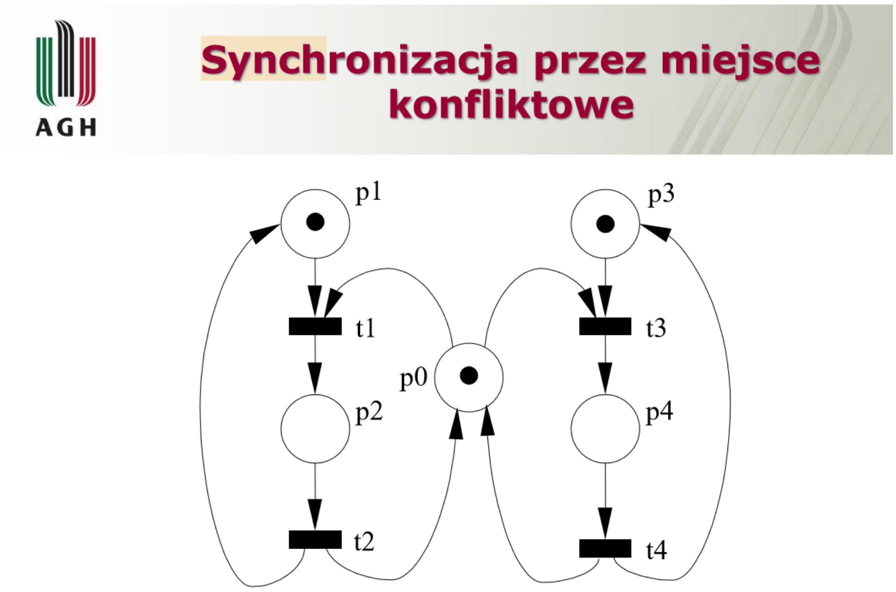


Rysunek 22: Przykład indeterminizmu w sieciach Petriego.

4.9. Zagadnienie 9

Zagadnienie 9

Synchronizacja procesów współbieżnych w sieciach Petriego

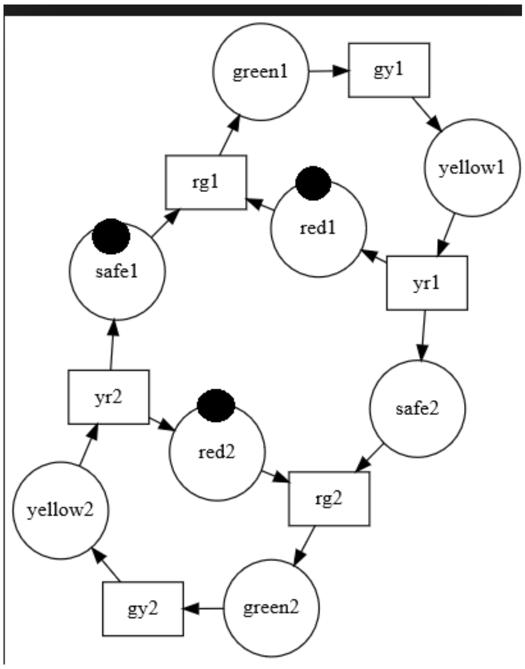


Rysunek 23: Przykład synchronizacji przez miejsce konfliktowe.

- Zamodelować dwa procesy z synchronizacją.

4.10. Różne inne

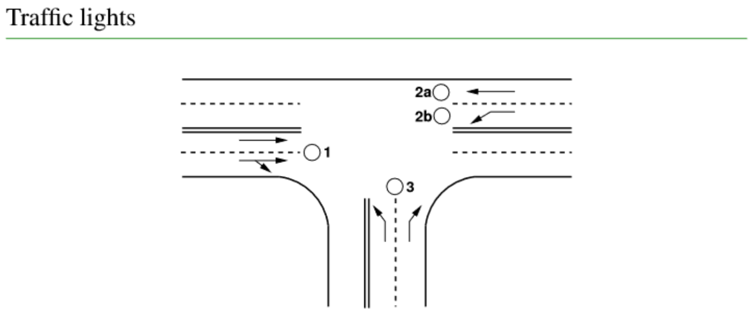
4.11. Sygnalizacja świetlna bez skrętu



Rysunek 24: Sygnalizacja świetlna bez skrętu (np. na zwężeniu mostu)

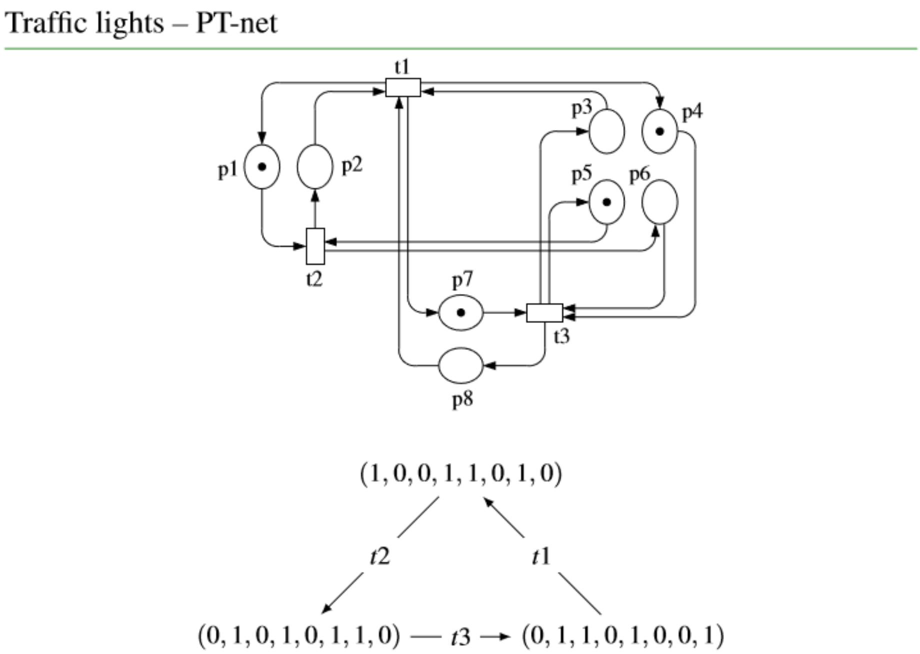
- Zamodelować sygnalizację bez skrętu.

4.12. Sygnalizacja świetlna ze skrętem



State	1	2a	2b	3
1	green	green	red	red
2	red	green	green	red
3	red	red	red	green

Rysunek 25: Zjawisko do zamodelowania.



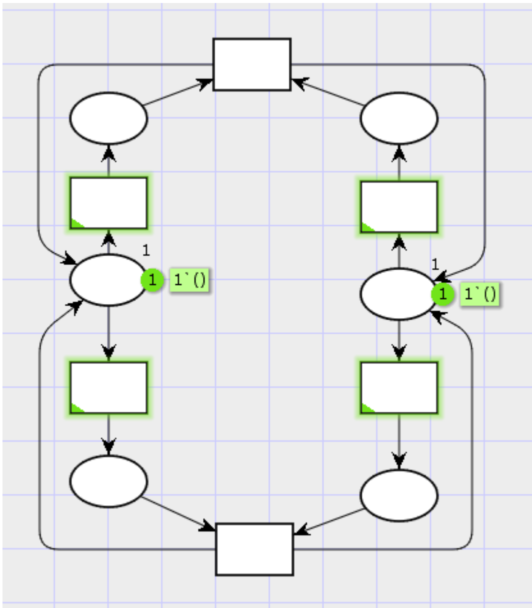
Rysunek 26: Model skrzyżowania ze skrętem.

- Zamodelować sygnalizację ze skrętem.

4.12.1. Zadanka z kolosa 2018

- Zaproponować sieć Petriego, która po odpaleniu kilku przejść wchodzi w stan zakleszczenia.

Np. dwóch filozofów:



Rysunek 27: Przykład sieci Petriego, gdzie będzie deadlock.

- Zaprojektować sieć Petriego, w której zachodzi wykluczanie jednoczesnej aktywności wybranej pary przejść.

Np. tak jak w zagadnieniu 9, gdzie wykluczają się t2 i t4.