Rozszerzenia sieci Petriego

Ograniczenia klasycznej sieci Petriego

- Trudność w modelowaniu specyficznych przepływów: testowania braku żetonów w danym miejscu, blokowania odpalania, itp.
- Brak determinizmu dla konfliktowych przejść
- Dla złożonych procesów, tworzone modele ze względu na wielkość są trudne do specyfikacji i interpretacji
- Modele nie odzwierciedlają aspektów czasowych, np. czasu pozostawania żetonów w miejscach, uniemożliwiając wykonywanie analizy wydajnościowej
- Brak możliwości modelowania semantyki przepływu i przetwarzania danych

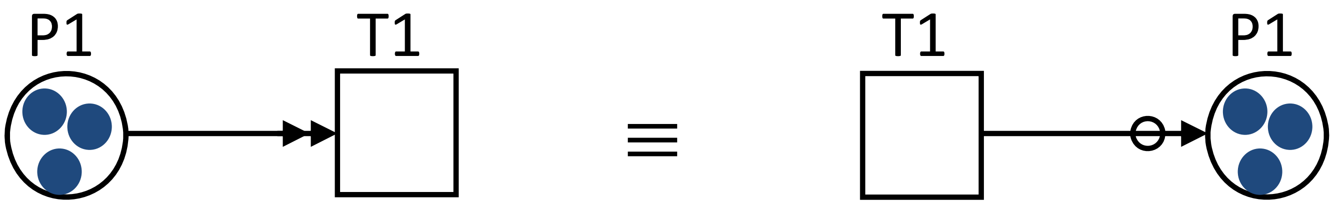
Rozszerzenia sieci Petriego

Sieci Petriego są rozszerzane o następujące własności:

- Nowe typy łuków: zerujące i inhibitory
- Sieci priorytetowe
- Czasowe sieci Petriego
- Kolorowane sieci Petriego
- Czasowe i kolorowane sieci Petriego
- Hierarchiczna struktura sieci

Łuki zerujące

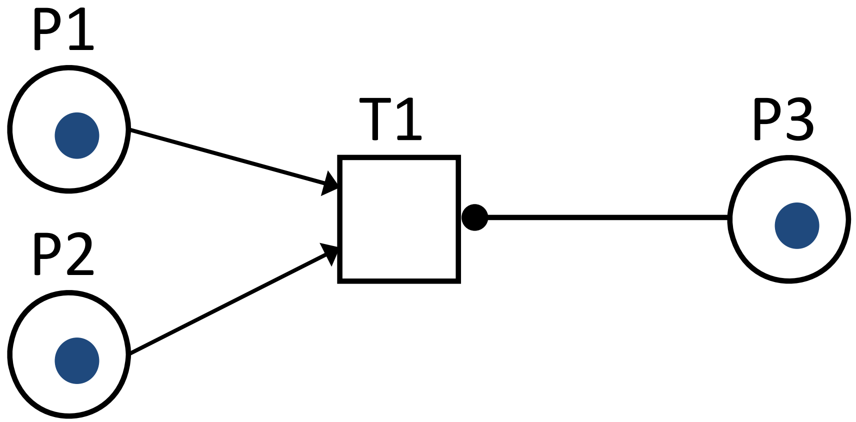
- Łuki zerujące są specjalnym rodzajem łuków służących do całkowitego zerowania zawartości przyłączonych do nich miejsc w wyniku odpalenia przejścia
- Dwie alternatywne notacje:



- Przejście konsumuje wszystkie żetony z miejsca wejściowego
- Odpalenie przejścia zeruje wszystkie żetony w miejscu wyjściowym

Inhibitory

- Łuki typu – inhibitor, blokują odpalanie aktywnych przejść, kiedy miejsce połączone z przejściem za pomocą inhibitora zawiera żetony



- Aktywne przejście T1 nie może odpalić z powodu zablokowania przez miejsce P3.

Priorytetowe sieci Petriego

- Przez przypisanie przejściom sieci Petriego priorytetów można określić częściowy porządek w zbiorze przejść, który wyeliminuje lub ograniczy sytuacje niejednoznaczne – brak determinizmu w działaniu sieci.
- Jeżeli aktywne są dwa przejścia, które rywalizują o ten sam zbiór żetonów, to **odpalone zostanie przejście o wyższym priorytecie**.

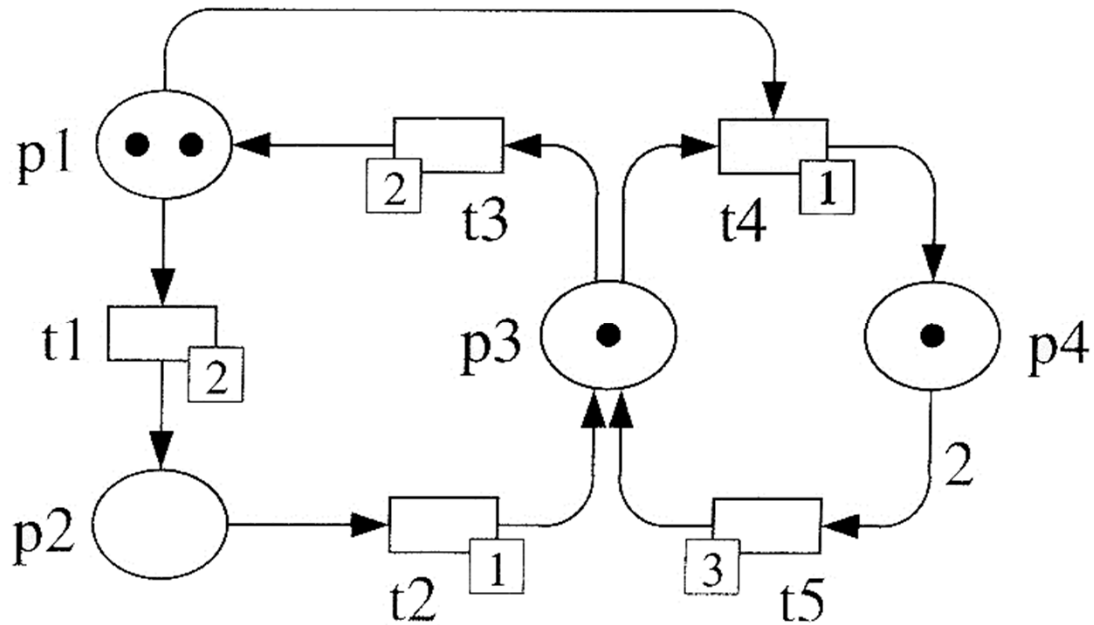
Priorytetowe sieci Petriego

Formalnie, priorytetowa sieć Petriego jest uporządkowaną szóstką: (P, T, I, O, R, S), gdzie:

- P jest skończonym zbiorem miejsc,
- T jest skończonym zbiorem przejść,
- I : P x T -> **N** jest funkcją wejść, N jest liczbą krawędzi między p i t,
- O : T x P -> **N** jest funkcją wyjść, N jest liczbą krawędzi między t i p,
- **R : T -> N $\cup$ 0 jest funkcją priorytetów, przypisującą każdemu z przejść liczbę całkowitą nieujemną,**
- S: P-> **N** jest to funkcja odwzorowująca zbiór miejsc w zbiór liczb naturalnych reprezentujących liczbę żetonów w danym miejscu.

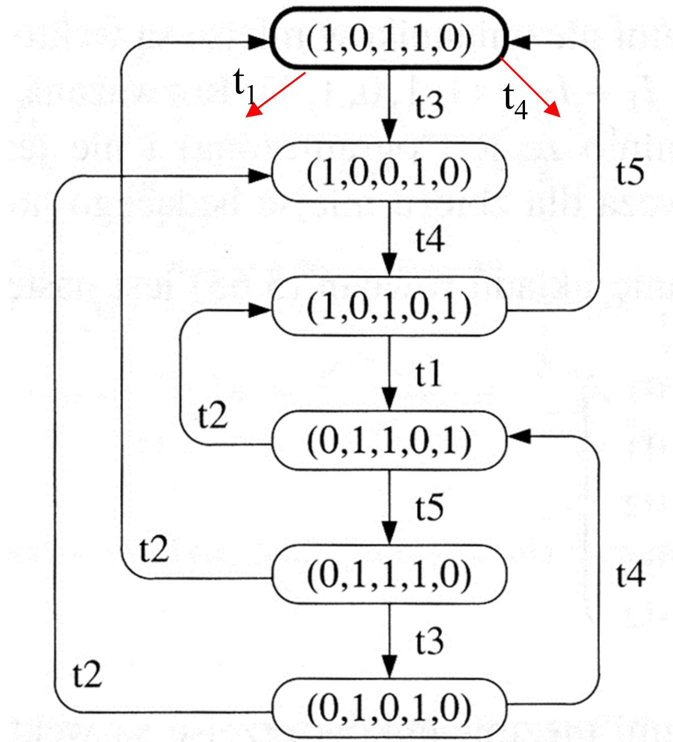
Przykład sieci priorytetowej

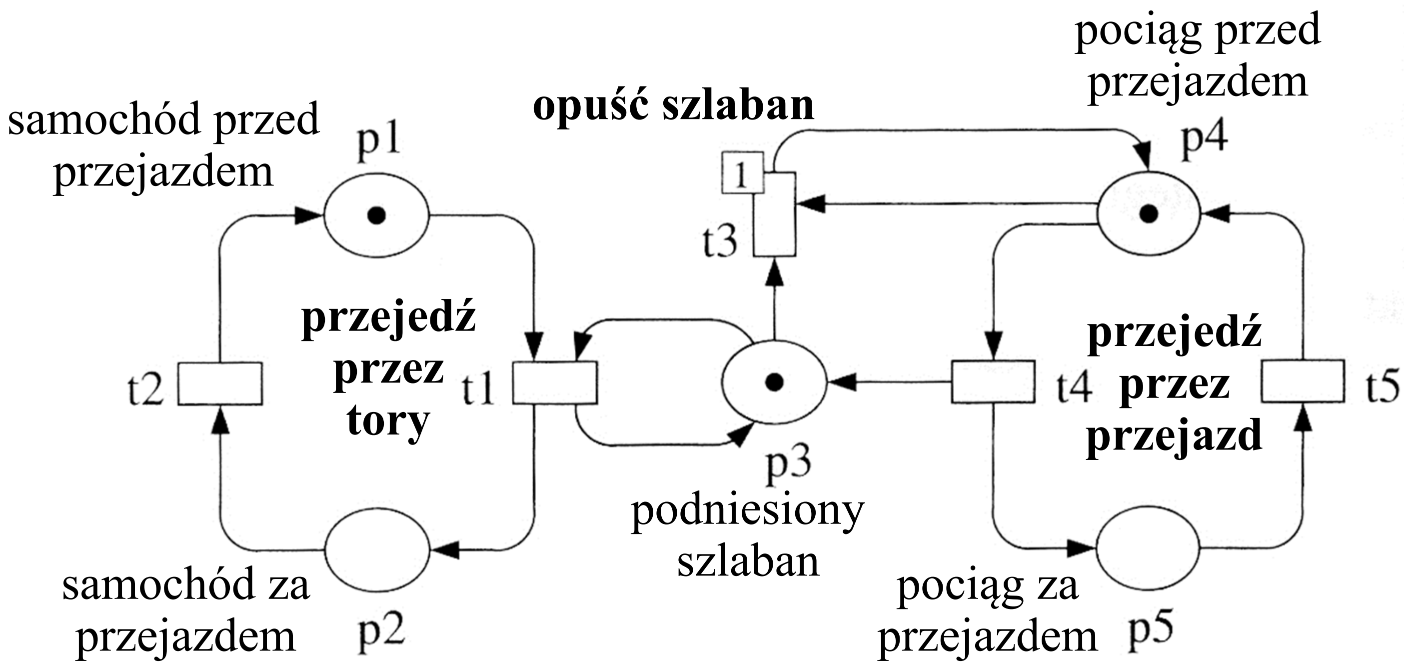
- W danym stanie sieci priorytetowej aktywne są przejścia, których miejsca wejściowe mają wystarczającą liczbę znaczników oraz ich priorytet **jest nie mniejszy** niż dowolnego innego przejścia, z którym jest w konflikcie.
- W danym stanie sieci aktywne są jedynie przejścia t_1 i t_3. Miejsce t_4 nie jest aktywne, bo ma priorytet mniejszy niż t_1 i t_3.



Przykład sieci priorytetowej

- Przejazd kolejowy – przejazd pociągu ma większy priorytet od przejazdu samochodów (domyślny priorytet równy 0).





Sieci czasowe

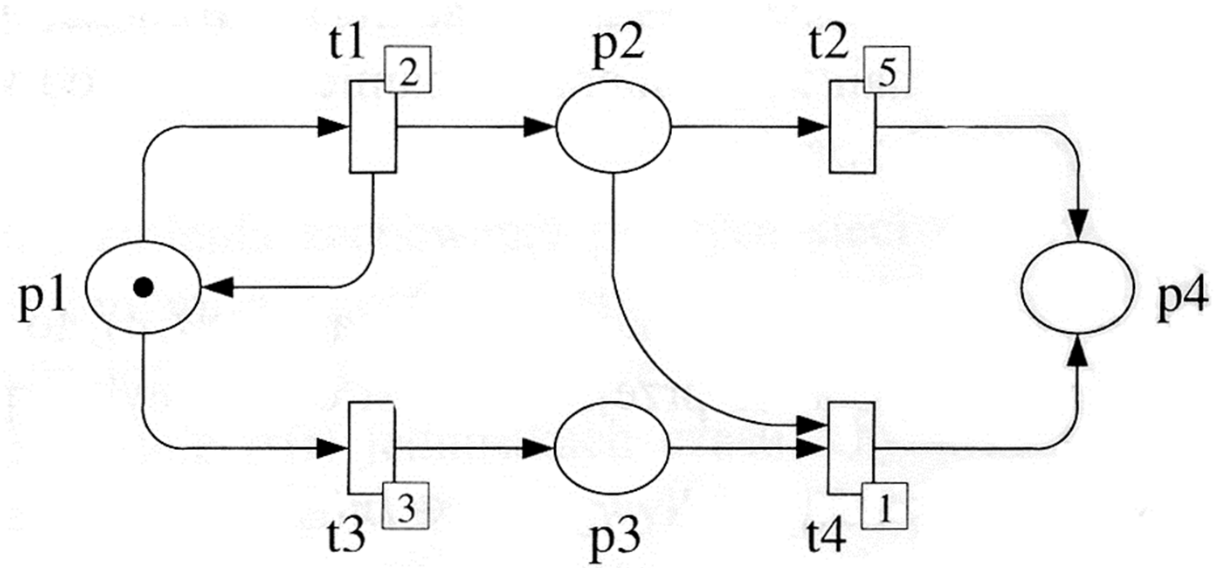
Sieci czasowe umożliwiają modelowanie czasu odpalenia przejść

- Proste – pozwalają przypisać przejściom czas ich odpalenia liczony od momentu uaktywnienia przejścia
- Przedziałowe – pozwalają przypisać przejściom przedział czasu [t_{min}, t_{max}], w którym mogą być odpalone, liczony od momentu uaktywnienia danego przejścia.

Proste sieci czasowe

Formalnie, czasowa sieć Petriego jest uporządkowaną szóstką: (P, T, I, O, σ, S), gdzie:

- σ : T -> Q, gdzie Q jest liczbą reprezentującą opóźnienie odpalenia przejścia $t \in T$. Aktywne przejście musi być odpalone po dokładnie σ jednostkach czasu, chyba że wcześniej zostanie dezaktywowane.

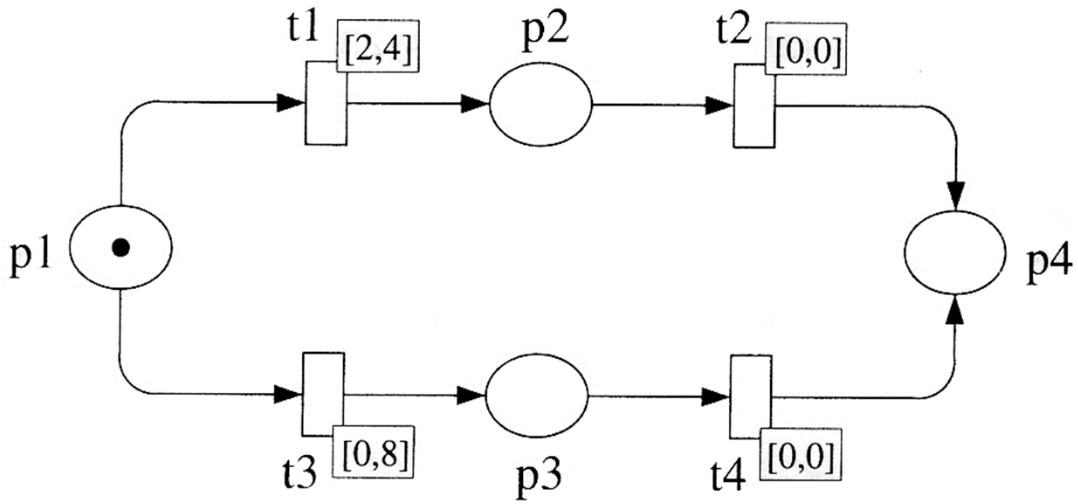


Aktywne są przejścia: t1 i t3. Po dwóch jednostkach czasu odpali przejście t1.
Jeżeli dwa aktywne przejścia mają takie same opóźnienie, odpalone zostanie niedeterministycznie wybrane jedno z nich.

Przedziałowe sieci czasowe

Formalnie, przedziałowa czasowa sieć Petriego jest uporządkowaną szóstką: (P, T, I, O, σ, S), gdzie:

- σ : T -> Q $\times$ Q$\cup\infty$, gdzie Q są liczbami reprezentującymi minimalny i maksymalny czas odpalenia przejścia t$\in$T



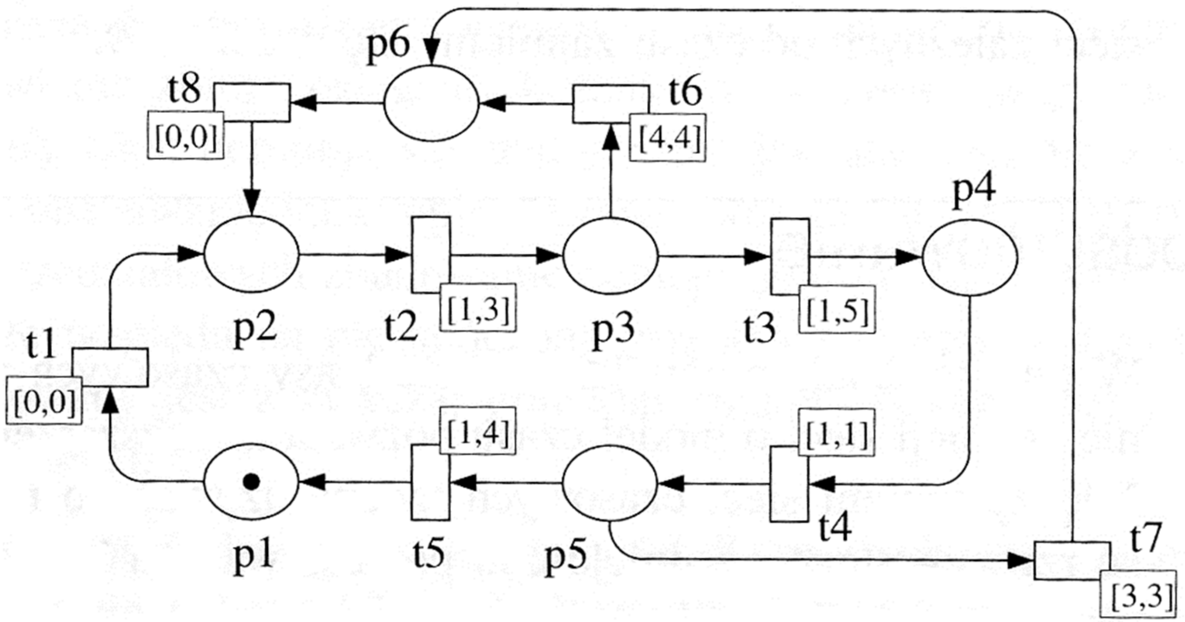
Aktywne są przejścia: t1 i t3. W przedziale czasu [0-4] niedereministycznie może odpalić przejście t3, w przedziale czasu [2-4] przejście t1.

Przykład - model protokołu komunikacyjnego

W każdym ze stanów poniższej sieci Petriego, znajduje się dokładnie jeden znacznik reprezentujący jeden z 6 stanów, w którym może znajdować się protokół.

t1 – wygenerowanie wiadomości
t2 – wysłanie wiadomości przez nadawcę
t3 – odebranie wiadomości przez odbiorcę
t4 – wysłanie potwierdzenia przez odbiorcę
t5 – odebranie potwierdzenia przez nadawcę

t6 – zgubienie wiadomości
t7 – zgubienie potwierdzenia
t8 – żądanie ponowienia transmisji



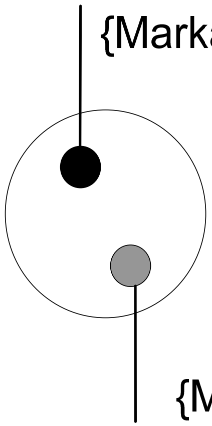
Kolorowane sieci Petriego nowe własności

- Miejsca mają przypisane typy danych
- Znaczniki mają wartości
- Łuki mają przypisane wyrażenia logiczne (zastrzeżenia), które opisują warunki konsumowania żetonów z miejsc wejściowych
- Łuki mają przypisane wyrażenia typu przypisanego miejscom, określające sposób wyznaczania wartości konsumowanych i produkowanych żetonów

Typy i wartości danych

- Żetony mają kolor, czyli przypisane są im wartości
- Miejsca są typowane, mają przypisany zbiór kolorów

{Marka="BMW", Numer rejestracyjny="PO-789CE", Rok=2003, Kolor="czarny"}

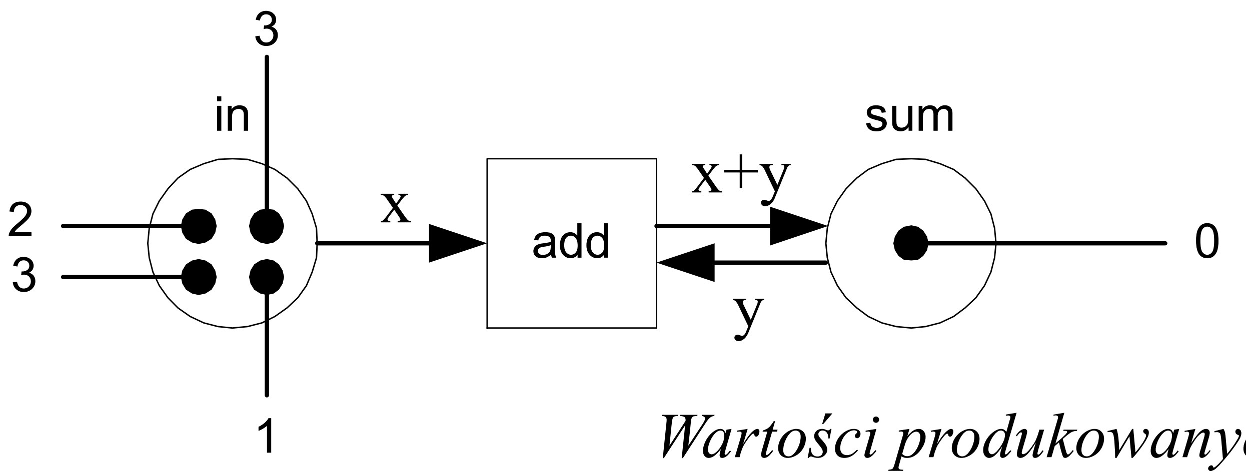


record Marka:string * Numer rejestracyjny:string * Rok:int * Kolor:string

{Marka="Ford", Numer rejestracyjny="PZ-123PX", Rok=1996, Kolor="szary”}

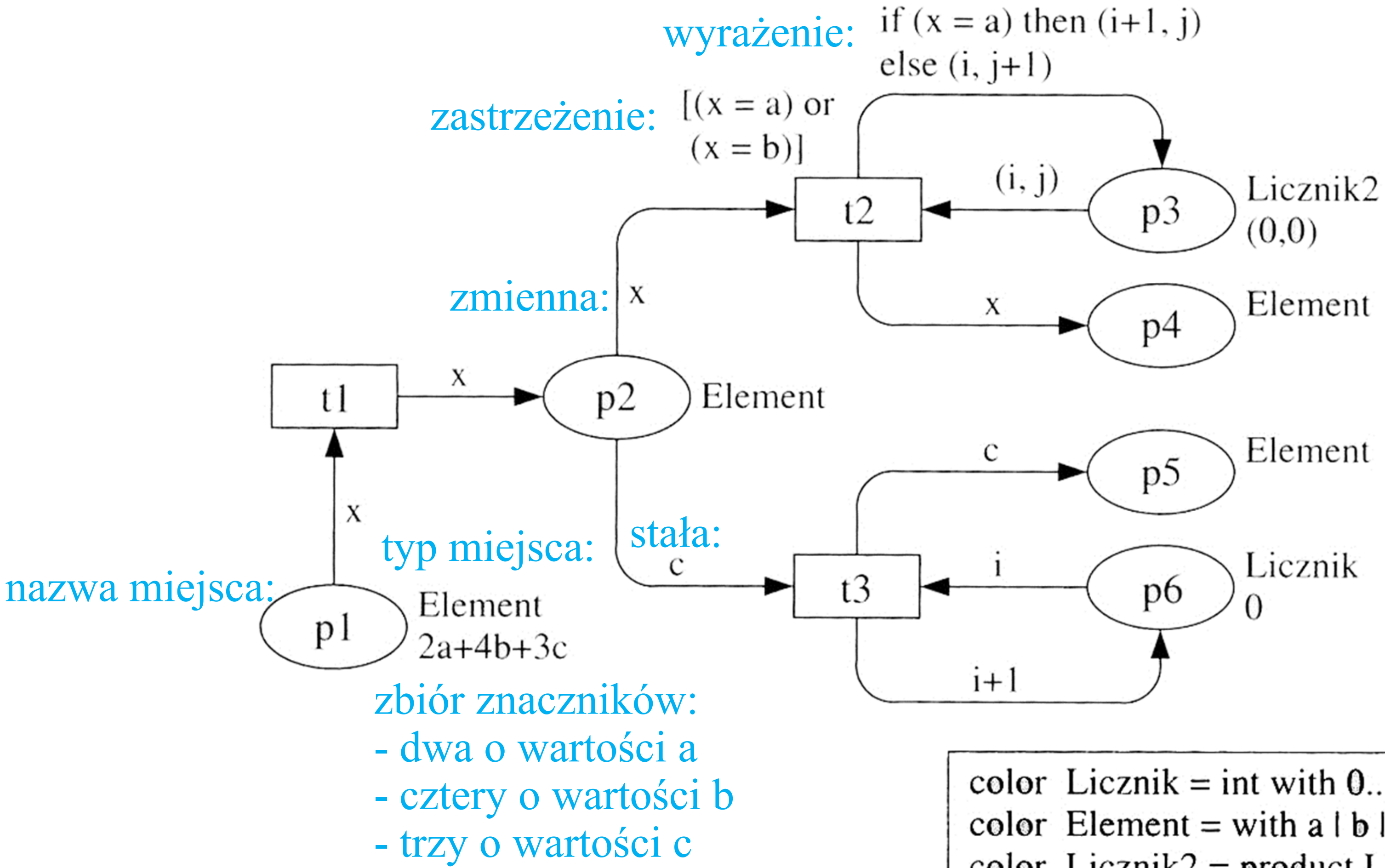
Wyznaczanie wartości żetonów

- W przejściach można zdefiniować wyrażenia wyznaczające wartości produkowanych żetonów.



*Wartości produkowanych żetonów w miejscu **sum** jest sumą wartości konsumowanych żetonów.*

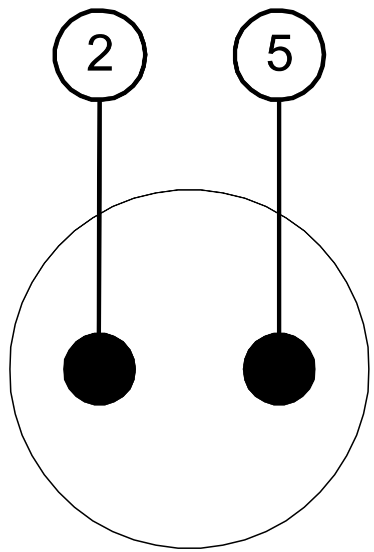
Kolorowane sieci Petriego - przykład




```
color Licznik = int with 0..100;
color Element = with a | b | c;
color Licznik2 = product Licznik * Licznik;
var i, j : Licznik;
var x : Element;
```

Rozszerzenie kolorowanych sieci Petriego o element czasu

- Każdy żeton ma przypisany **znacznik czasowy**
- Znacznik czasowy żetonu określa **czas** kiedy żeton może być **najwcześniej** skonsumowany

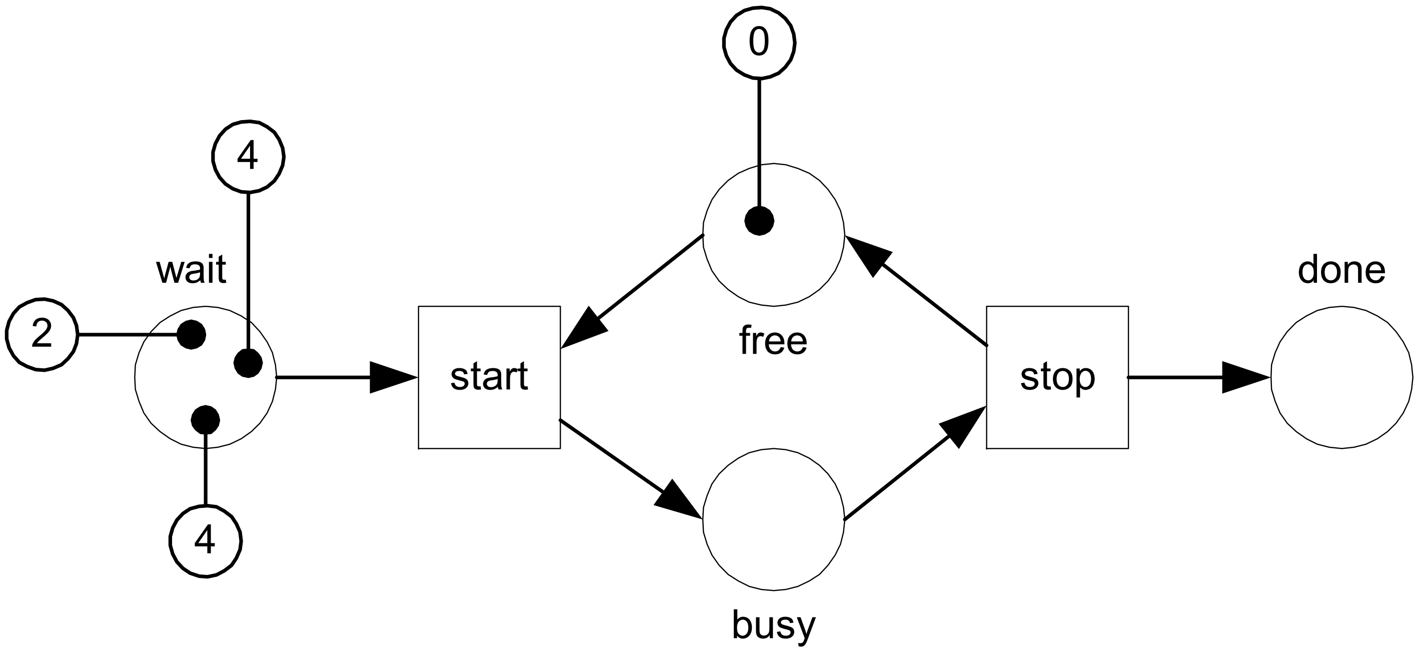


Rozszerzenie o element czasu

- Czas aktywowania przejścia odpowiada największemu znacznikowi czasowemu ze zbioru żetonów przeznaczonych do konsumpcji przez to przejście.
- Jeżeli w danym miejscu jest wiele żetonów, żeton o najmniejszym znaczniku czasowym jest konsumowany jako pierwszy.
- Przejście o najmniejszym czasie odpalenia będzie odpalone jako pierwsze.
- Przejścia są odpalane natychmiast, kiedy tylko mogą być odpalone.
- Produkowane żetony mogą mieć opóźnienie.
- Znaczniki czasowe produkowanych żetonów są równe czasowi odpalenia powiększonym o czas opóźnienia.

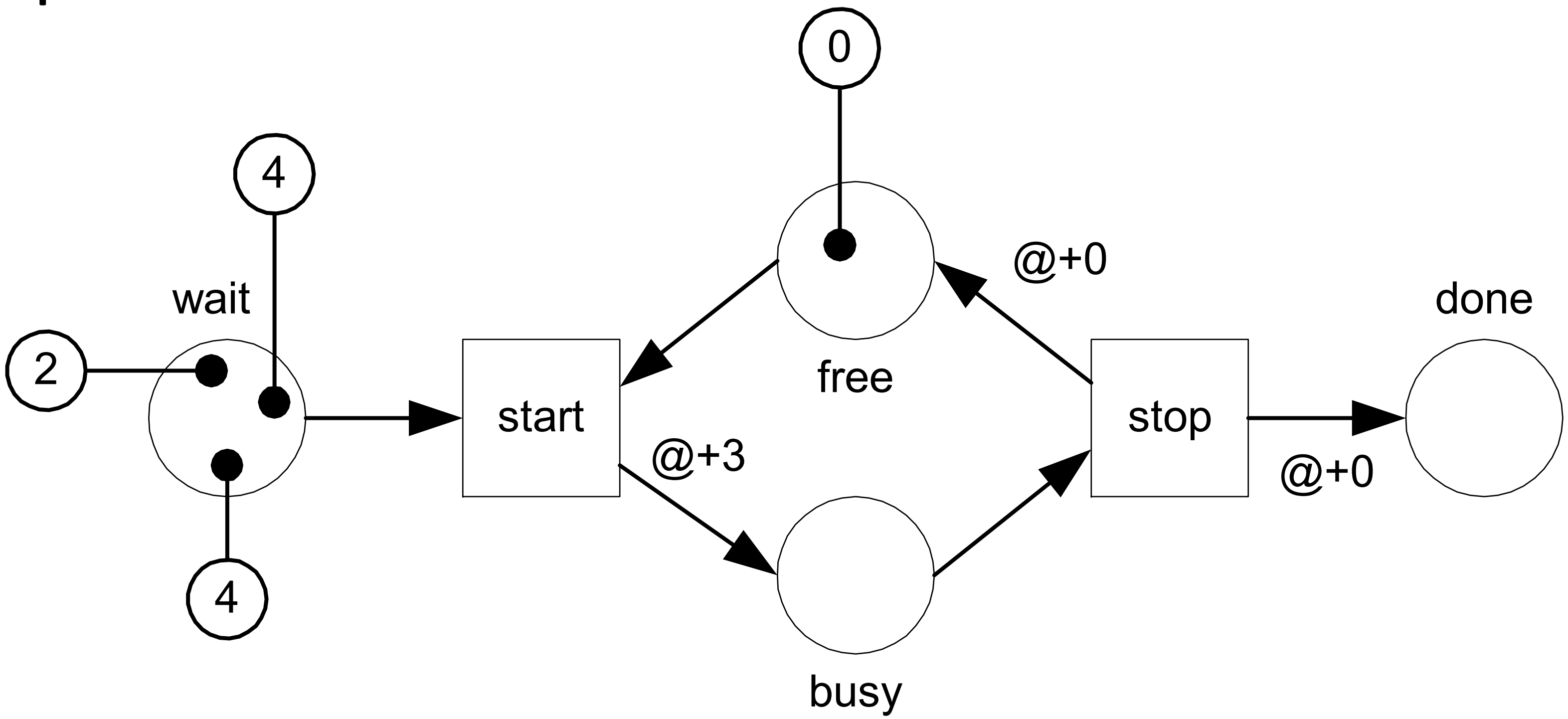
Ustalenie czasu aktywacji

- Czas aktywacji przejścia *start* w poniższym przykładzie wynosi: 2 = max{0,min{2,4,4}}.



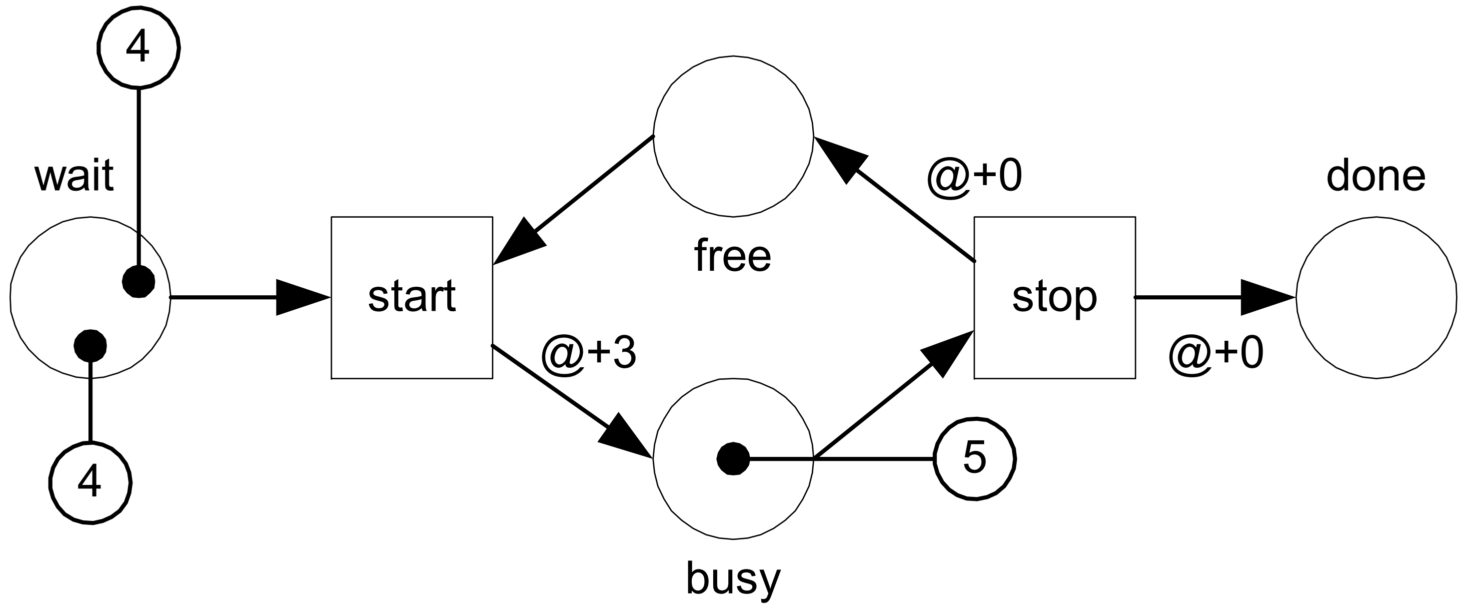
Czas produkowania żetonów

- Żetony produkowane do miejsca *busy* mają opóźnienie równe 3
- @+3 = czas odpalenia przejścia start plus 3 jednostki opóźnienia



Czas odpalenia przejścia

- Przejście *start* zostanie odpalone w jednostce czasu 2.
- Żeton wyprodukowany do miejsca busy będzie miał wartość 2 + 3 =5

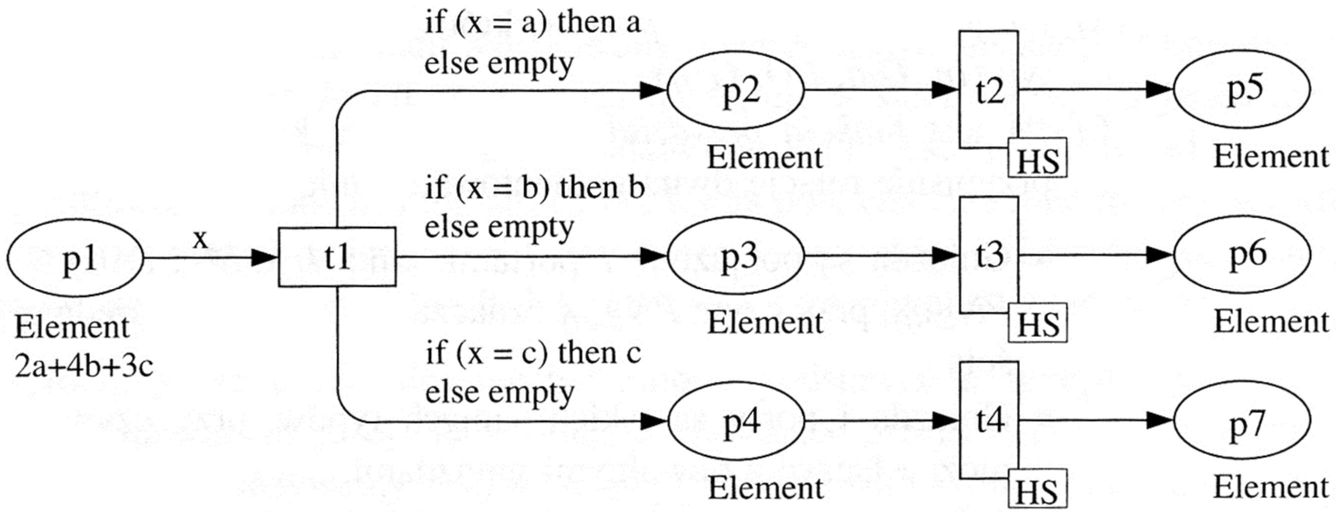


Hierarchiczne sieci Petriego

- Hierarchiczne sieci kolorowane umożliwiają tworzenie dużych modeli przez łączenie mniejszych fragmentów sieci zwanych ***pod-stronami*** w wielopoziomową hierarchię stron. Całe fragmenty sieci mogą być zastępowane pojedynczymi przejściami podstawianymi.
- Strony sieci mogą być łączone za pomocą dwóch konstrukcji:
  - ***Podstawiane przejścia***, które służą do łączenia sieci pod-stron, z sieciami nad-stron poprzez **gniazda** (miejsca w nad-stronach) i **miejsca portowe** (miejsca w pod-stronach)
  - ***Fuzje miejsc***, które są nierozróżnialnymi miejscami występującymi w różnych pod-stronach sieci
- Umożliwia to modelowanie procesów na różnych poziomach abstrakcji

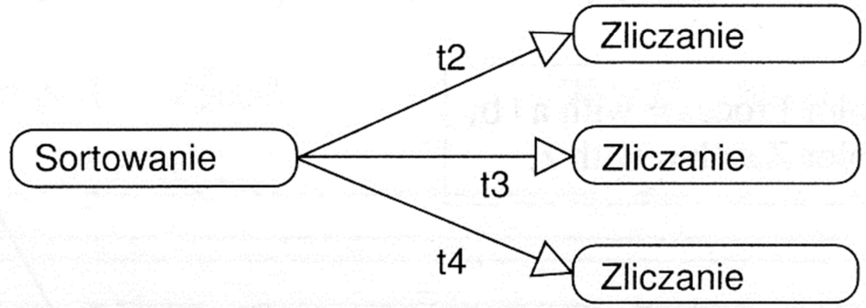
Strona główna sieci

- Strona główna sieci – *Sortowanie* obejmuje trzy przejścia podstawiane – *Zliczanie* HS (ang. hierarchy substitution): t2, t3 i t4 oraz odpowiadające im gniazda wejściowe i wyjściowe (p2,p5), (p3,p6) i (p4,p7).




```
color Licznik = int with 0..100;
color Element = with a | b | c;
var i : Licznik;
var x : Element;
```

Hierarchia stron



Pod-strony sieci

Pod-strona - *Zliczanie*, zawiera dwa miejsca portowe:

- miejsce *p8* typu **IN** – odpowiadające gniazdom wejściowym,
- miejsce *p9* typu **OUT** – odpowiadające gniazdom wyjściowym,

oraz fuzję miejsc:

- miejsce p10 – które jest współdzielone przez wszystkie trzy wystąpienia strony licznik.