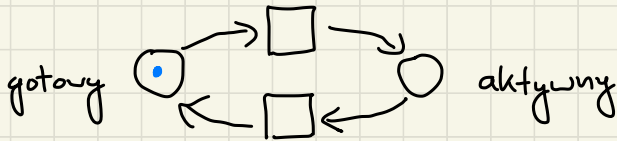


Ćwiczenia 1

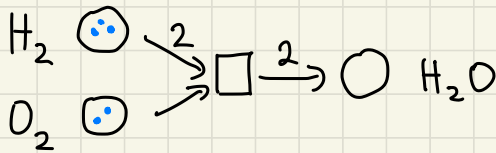
2.10

Sieci Petriego - krótkie wprowadzenie

* przykład: stan procesu



* przykład: reakcje chemiczne



Elementarna sieć Petriego z konfiguracją:

* zbiór miejsc P i tranzyje T , $P \cap T = \emptyset$

* zbiór Tuków $F \in (P \times T) \cup (T \times P)$

* konfiguracja $M \in \mathbb{N}^P$

Ogólna sieć:

* F to Tuki z przypisanymi wagami ($\in \mathbb{N}$)

* $M: P \rightarrow \mathbb{N}$ jako uogólnienie konfiguracji

Kiedy można odpalić tranzycję?

* w elementarnych sieciach, gdy mamy po
żetonie na miejscach wejściowych (ozn. t^+),
a miejsca wyjściowe (t^-) są puste

* w ogólnych sieciach, gdy na miejscach
wejściowych liczby żetonów są co najmniej
takie jak wagi Tuków

od teraz mamy na
myśli sieci ogólne

Ważne własności (mówimy o sieci z konfiguracją)

* osiągalność - czy z konfiguracji początkowej można
na skutek odpalenia pewnej sekwencji tranzycji
przejsć do innej?

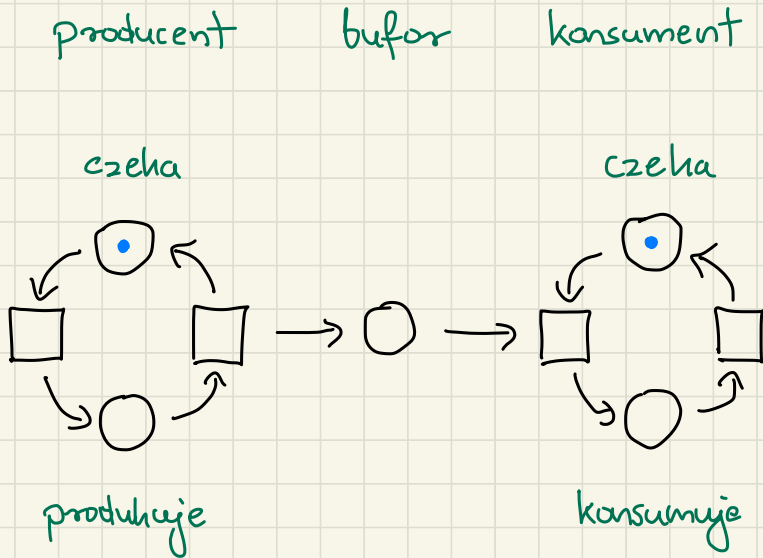
* żywotność - każda tranzycja jest żywa;
tranzycja t jest żywa, gdy z każdej osiągalnej
konfiguracji można odpalić pewną sekwencję
tranzycji, po których możliwe jest odpalenie
tranzycji t

- * **k-ograniczoność** - każde miejsce jest k-ograniczone; miejsce jest k-ograniczone, jeśli w każdej osiągalnej konfiguracji zawiera nie więcej niż k żetonów
- * **ograniczoność** - sieć z konfiguracją jest ograniczona, jeśli istnieje $K \in \mathbb{N}$, takie że mamy k-ograniczoność
- * **brak blokad** - nie można osiągnąć konfiguracji w której wszystkie tranzycje są martwe (nie można ich odpalić)

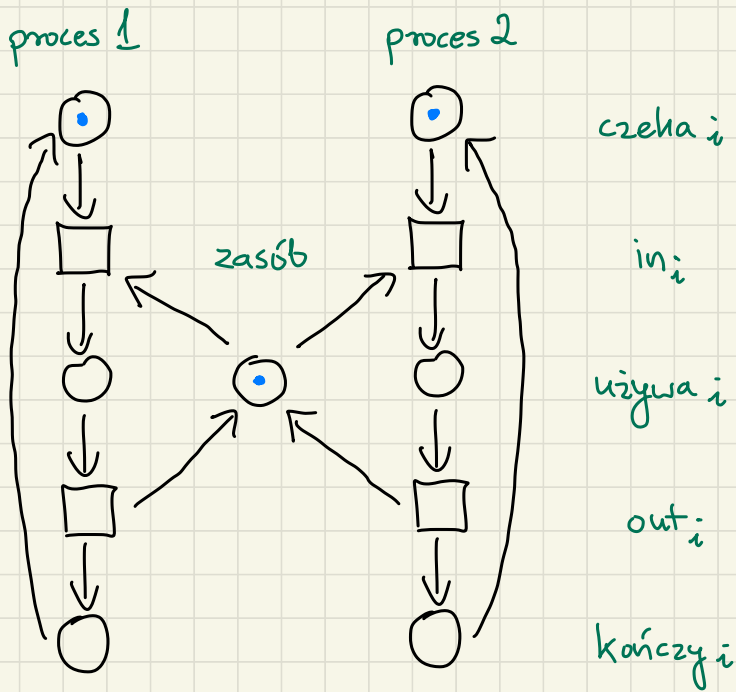
Możemy też mówić o strukturalnych własnościach sieci, które nie zależą od wybranej konfiguracji.

Przykładowo, sieć jest **strukturalnie ograniczona**, jeśli dla każdej możliwej konfiguracji początkowej M sieć z konfiguracją M jest ograniczona.

1. Zamodeluj za pomocą sieci elementarnych klasyczne problemy współbieżności :
producenta / konsumenta i wzajemne wykluczanie



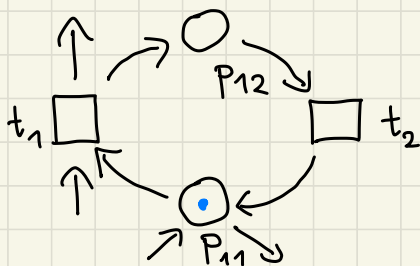
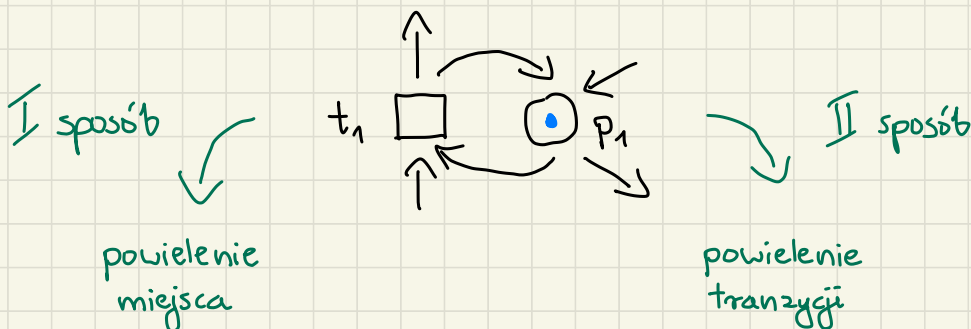
- * producent i konsument nie komunikują się bezpośrednio
- * jedyna komunikacja zachodzi przez bufor



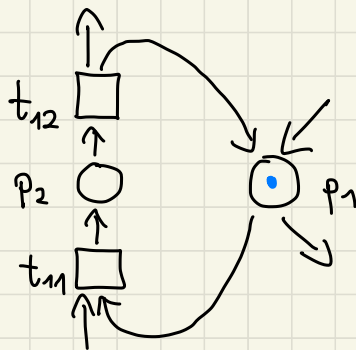
- * procesy 1 i 2 nigdy nie komunikują się ze sobą bezpośrednio, tzn. nie wykonują wspólnych akcji (tranzyjce)
- * łatwo uogólnić na więcej procesów
- * nie pilnujemy tutaj, aby każdy proces, który chce skorzystać z zasobu, mógł tego w końcu dokonać (problem zgodzenia)

2. Eliminacja ciasnych pętli, eliminacja wag > 1

ciasna pętla = miejsce i tranzycja są połączone w obie strony



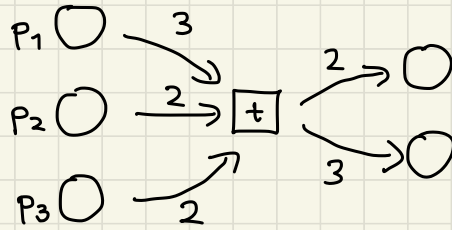
po odpaleniu t_1 tranzycja t_2 może w dowolnym momencie przenieść żeton z P_{12} na P_{11}



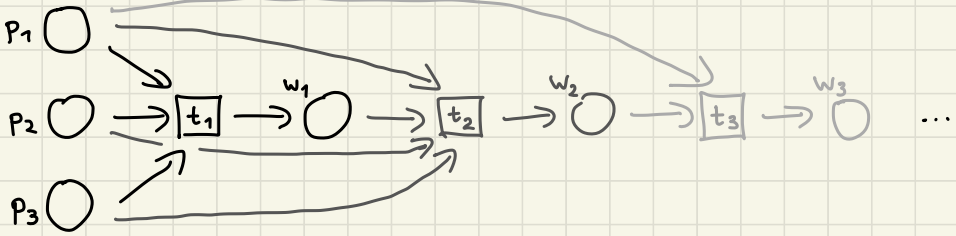
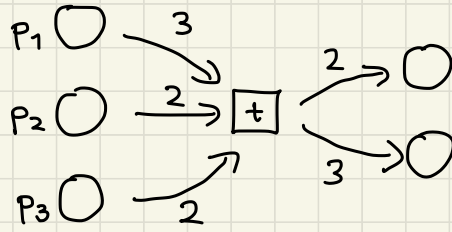
przy tej modyfikacji żeton na miejscu P_2 oznacza coś w stylu "wykonuje się tranzycja t_1 "

* założenie $t \cap t' = \emptyset$ upraszcza analizę własności sieci

Eliminacja wag > 1



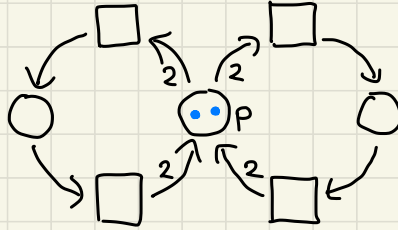
- * skoro mamy dostępne tylko Tuki o wagach 1, pomysł jest taki, aby stopniowo pobierać po jednym żetonie z każdego miejsca
- * w przykładzie tranzycja t pobiera 3 żetony z P_1 oraz po 2 żetony z P_2 i P_3 , dlatego „rozbijemy” „część pobierająca” żetony na 3 nowe tranzyje: t_1, t_2, t_3
- * pierwsza z nich, t_1 , pobierze po 1 żetonie z każdego miejsca, druga zrobi to samo, a trzecia zabierze już tylko jeden żeton z P_1
- * stąd łącznie t_1, t_2 i t_3 zabiorą 3 żetony z P_1 i po 2 żetony z P_2, P_3 , czyli dokładnie tyle, co tranzycja t



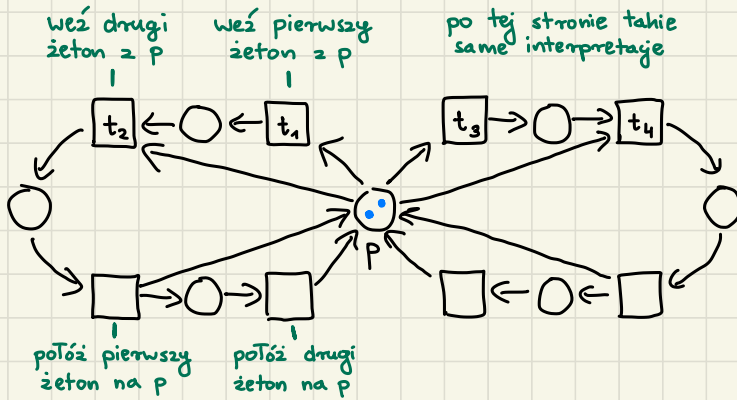
* w powyższej konstrukcji dodaliśmy też miejsca wyjściowe w_1, w_2, w_3 odpowiednio dla t_1, t_2, t_3 , które oznaczają: wzięto po 1, 2, 3 zetonów z miejsc, z których musieliśmy tyle wziąć

* **problem**: w oryginalnej sieci może być niemożliwe osiągnięcie dead-locha, a po tej transformacji sieć może się zahleszczać

oryginalna sieć:

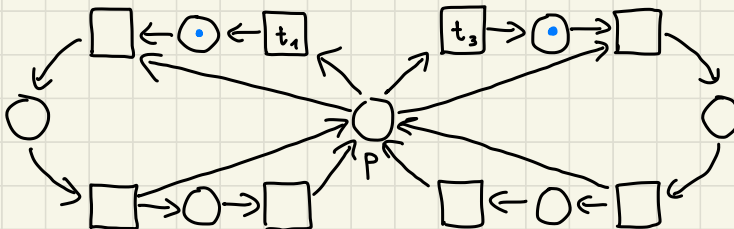


sieć po transformacji:



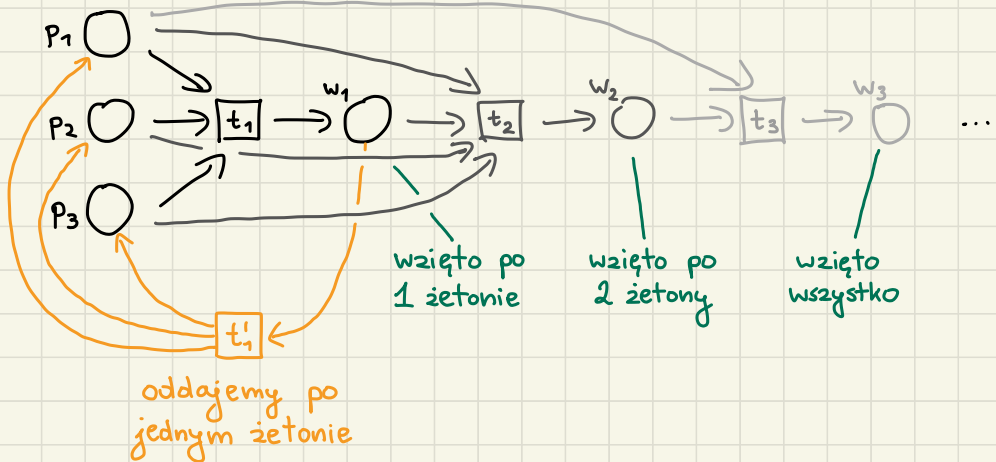
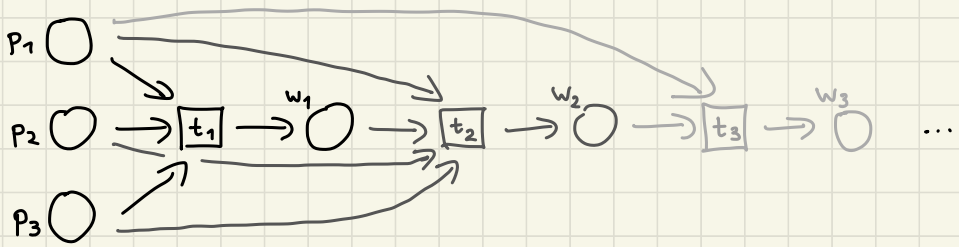
w oryginalnej sieci nie było zakleszczenia,

w nowej po odpaleniu t_1 i t_3 mamy dead-lock

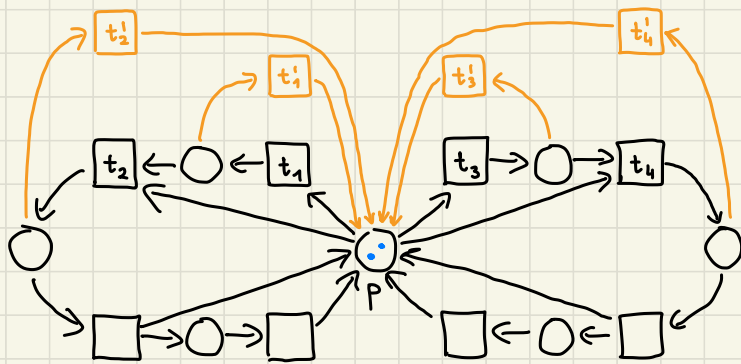


jak to naprawić?

- * wystarczy dodać tranzycje, które będą oddawać żetony z powrotem na początkowe miejsce
- * w pierwszym przykładzie dla tranzycji t_1 tworzymy tranzycję t'_1 , która pobiera żeton z miejsca w_1 (oznaczającego "pobrano po jednym żetonie z p_1, p_2 i p_3 ") i kładzie po jednym żetonie na p_1, p_2 i p_3



* w drugim przykładzie cała nowa sieć będzie wyglądać tak:



* w takiej sieci nie wystąpi już zakleszczenie

* Tylko też dla niej zdefiniować, które jej konfiguracje odpowiadają, którym w oryginalnej sieci

* **uwaga:** jeśli zmienimy początkową konfigurację na taką z jednym żetonem na miejscu p, to oryginalna sieć będzie od razu w dead-locku, a ta powyżej nie (może odpalić t₁ i t'₁ na zmianę)

* **wniosek:** trzeba uważać, jakie własności sieci się zachowują przy jej zmienianiu

3. Równoważność sieci elementarnych oraz 1-ograniczonych sieci ogólnych

←

* możemy założyć, że nie ma wag > 1 ,
wpp tranzycje z takimi T-ami można usunąć

* otrzymana sieć jest siecią elementarną,
bo początkowa konfiguracja składa się na pola
pojedyncze żetony, a w dowolnej osiągalnej
konfiguracji M każda żywa tranzycja t
spełnia $t^0 \cap M = \emptyset$

⇒ 1^D (pierwsze zadanie domowe)

Uwaga: nie możemy po prostu użyć danej sieci
elementarnej i powiedzieć, że traktujemy ją jako ogólną
sieć - np. sieć producenta - konsumenta staje się wtedy
nieograniczona (na bufonie może się wygenerować
dowolna liczba żetonów)

Podpowiedź: w sieci elementarnej każde miejsce albo zawiera zeton albo jest puste, czyli tak jakby ma dwa stany

2.^D Skonstruuj sieć oraz podaj dla niej konfigurację, która jest:

a) żywa, ale nie ograniczona,

b) ograniczona, ale nie żywa.

3.^D Czy żywotność jest monotoniczna w ogólnych sieciach Petriego?

zadania do pomyślenia w domu (nieobowiązkowe)