Laboratorium 12

Sieci Petriego

Danylo Knapp

Teoria Współbieżności

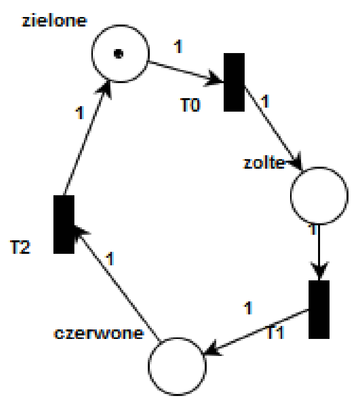
Wydział Informatyki
Akademia Górniczo-Hutnicza
im. Stanisława Staszica w Krakowie
07.01.24

1 Treść zadania

Do zajęć będziemy używać symulatora Pipe2. Jest napisany w Javie i jego uruchomienie nie wymaga uprawnien administratora (jest dostępny pod win i linux).

1.1 Maszyna stanów

Prosty model maszyny stanów swiateł ulicznych przedstawia sieć na rysunku poniżej:



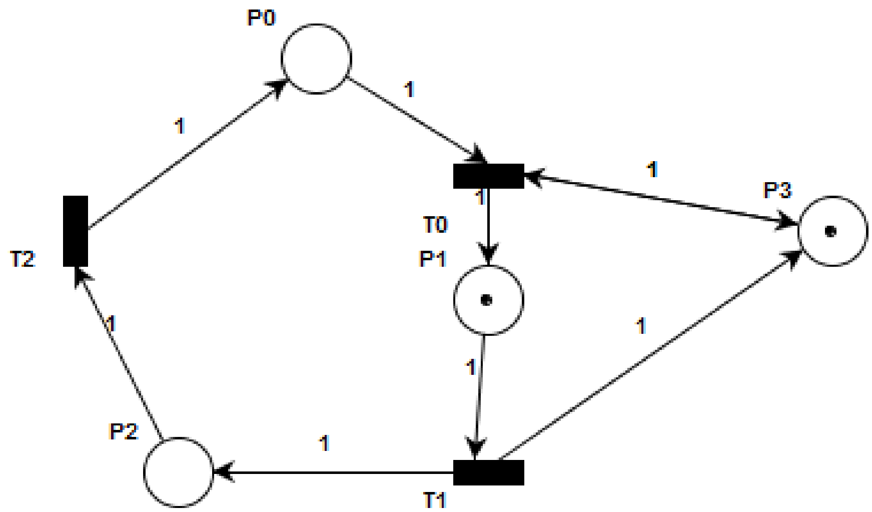
Stanami są miejsca sieci, zaś znacznik pokazuje w jakim stanie aktualnie się znajdujemy.

1.1.1 Ćwiczenia:

- Narysować przykład w symulatorze
- Sprawdzić właściwości sieci (ograniczoność, bezpieczenstwo i możliwy deadlock) w symulatorze Pipe w menu "State Space Analysis".
- Wygenerować graf osiągalności "Reachability/Coverability Graph". Zaobserwować:
  - Jakie znakowania są osiągalne?
  - Ile wynosi maksymalna liczba znaczników w każdym ze znakowań? Jakie możemy wyciągnac z tego wnioski n.t. ograniczoności i bezpieczenstwa?
  - Czy kazde przejście jest przedstawione jako krawędź w grafie? Jaki z tego wniosek n.t. żywotności przejść?
  - Czy wychodząc od dowolnego węzła grafu (znakowania) można wykonać dowolne przejście? Jaki z tego wniosek n.t. zywotnosci sieci? Czy sa możliwe zakleszczenia?
- Wykonać analizę niezmiennikow (wybrac w menu "Invariant Analysis").
  - Wynik analizy niezmienników przejść (T-invariants) pokazuje nam, ile razy trzeba odpalić dane przejście (T), aby przekształcić znakowanie początkowe z powrotem do niego samego (wynik nie mówi nic o kolejności odpaleń). Z wyniku mozemy m.in. wnioskować o odwracalności sieci.
  - Wynik analizy niezmienników miejsc (P-invariants) pokazuje nam zbiory miejsc, w których łączna suma znaczników się nie zmienia. Pozwala to wnioskować n.t. zachowawczości sieci (czyli własności, gdzie suma znaczników pozostaje stała) oraz o ograniczoności miejsc.

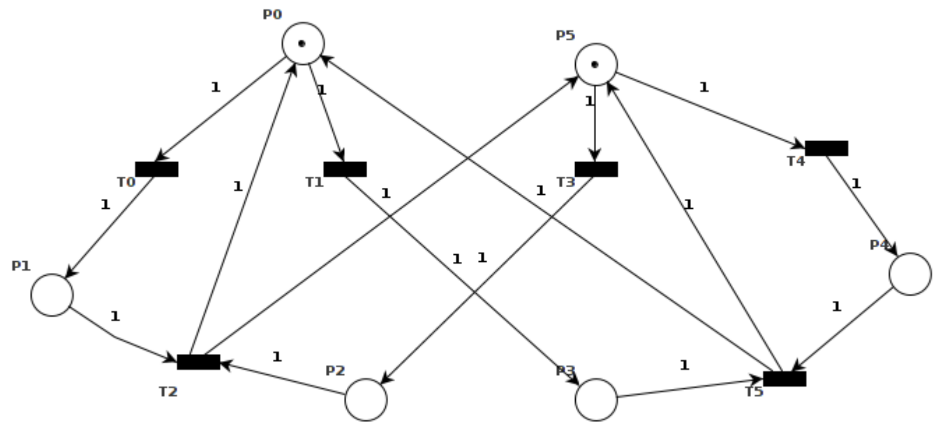
1.2 Zadania

1. Wymyślić własną maszynę stanów, zasymulować przykład i dokonać analizy grafu osiągalności oraz niezmienników j.w.
2. Zasymulować sieć jak poniżej:



   - Dokonać analizy niezmienników przejść. Jaki wniosek można wyciągnąć o odwracalności sieci?
   - Wygenerować graf osiągalności. Proszę wywnioskować z grafu, czy sieć jest żywa.
   - Proszę wywnioskować czy jest ograniczona. Objaśnić wniosek.
3. Zasymulować wzajemne wykluczanie dwoch procesów na wspólnym zasobie.
   - Dokonać analizy niezmienników.
   - Wyjaśnij znaczenie równań (P-invariant equations). Które równanie pokazuje działanie ochrony sekcji krytycznej?
4. Uruchomić problem producenta i konsumenta z ograniczonym buforem (można posłużyć się przykładem, menu:file, examples).
   - Dokonać analizy niezmienników.
   - Czy sieć jest zachowawcza?
   - Ktore równanie mówi nam o rozmiarze bufora?
5. Stworzyć symulacje problemu producenta i konsumenta z nieograniczonym buforem.
   - Dokonać analizy niezmienników.
   - Zaobserwować brak pełnego pokrycia miejsc.
6. Zasymulować prosty przykład ilustrujący zakleszczenie.
   - Wygenerować graf osiągalności i zaobserwować znakowania, z których nie można wykonać przejść.
   - Zaobserwować właściwości sieci w “State Space Analysis”.

Poniżej przykład sieci z możliwością zakleszczenia (można wymyślić inny):



2 Wstęp teoretyczny

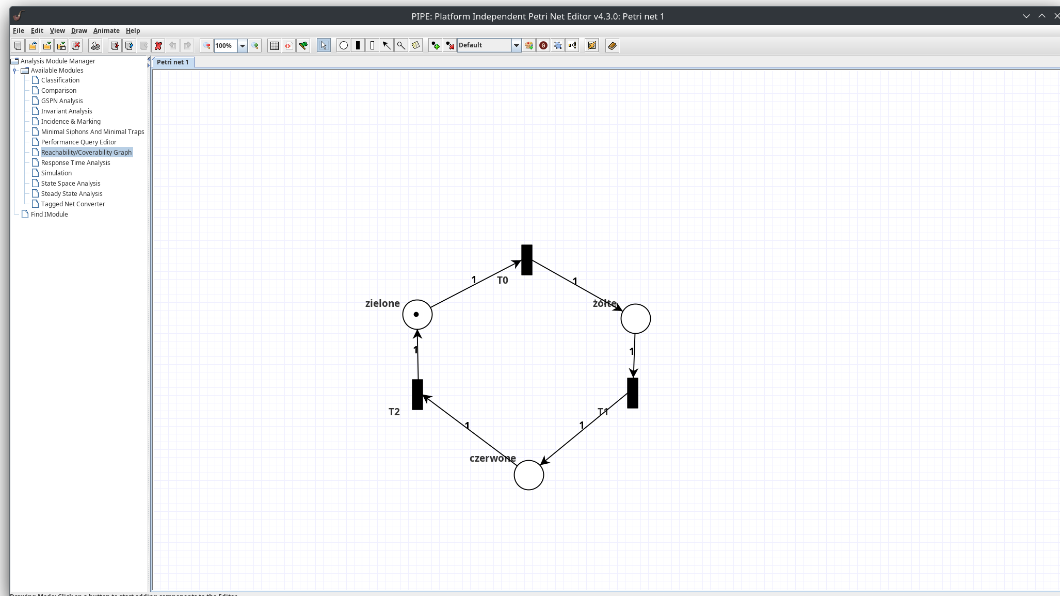
- **Sieci Petriego** - narzędzie wprowadzone przez Carla A. Petriego w 1962 roku do pierwotnie modelowania komunikacji z automatami. Obecnie narzędzie stosowane jest w modelowaniu systemów współbieżnych, dyskretnych, synchronizacji procesów i wielu innych.
- Własności:
  - **Ograniczoność**: odnosi się do maksymalnej liczby żetonów, które mogą znajdować się w danym miejscu. Sieć Petriego jest ograniczona, jeśli w żadnym z jej miejsc w trakcie działania sieci liczba żetonów (znaczników) nie może rosnąć w nieskończoność. Innymi słowy, istnieje pewna maksymalna liczba żetonów, która może znajdować się w każdym miejscu w dowolnym momencie.
  - **Bezpieczeństwo**: to specjalny przypadek ograniczoności, gdzie maksymalna liczba żetonów w miejscu wynosi 1.
  - **Deadlock**: stan, w którym żadne dalsze przejścia nie są możliwe.
  - **Zachowawczość sieci**: własność, gdzie suma żetonów pozostaje stała.
  - **Żywotność sieci** – każde przejście ma szanse się wykonać.
  - **Żywotność miejsca** – miejsce ma szanse zawierać znaczniki.
  - **Żywotność przejścia** – przejście ma szanse się wykonać.
  - **Odwracalność** oznacza, że istnieje możliwość powrotu do stanu początkowego (początkowego znakowania) z dowolnego innego stanu (znakowania) sieci. Innymi słowy, jeśli sieć jest odwracalna, to znaczy, że po serii przejść (odpalenia tranzycji) można wrócić do stanu początkowego.
- **Graf osiągalności** obrazuje zmienianie się warunków logicznych (lokalnych stanów).
- **Znaczniki** to żetony (tokeny), które można przemieszczać pomiędzy miejscami poprzez przejścia, po krawędziach grafu.
- Znaczniki (tokeny, żetony) prezentowane są graficznie w postaci kropek umieszczanych w kółkach reprezentujących miejsca.
- **Niezmienniki przejść (T-invariants)** pokazują, ile razy trzeba odpalić dane przejście (T), aby przekształcić znakowanie początkowe z powrotem do niego samego.

- **Niezmienniki miejsc (P-invariants)** pokazują zbiory miejsc, w których łączna suma znaczników się nie zmienia. Pozwala to wnioskować o zachowawczości sieci oraz o ograniczoności miejsc.
- **Równania (P-invariant equations)** są związane z niezmiennikami miejsc i pokazują zbiory miejsc, w których łączna suma znaczników się nie zmienia.

3 Rozwiązania

3.1 Maszyna stanów

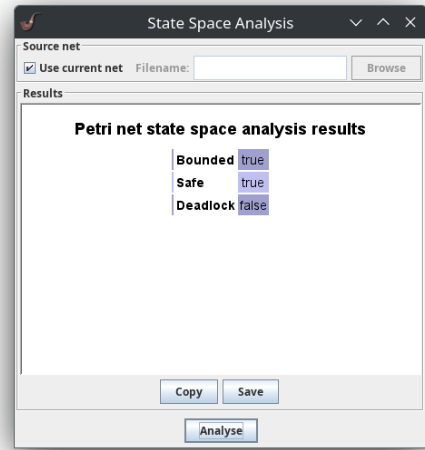
Maszyna została narysowana w symulatorze w sposób następujący:



3.1.1 Analiza stanów

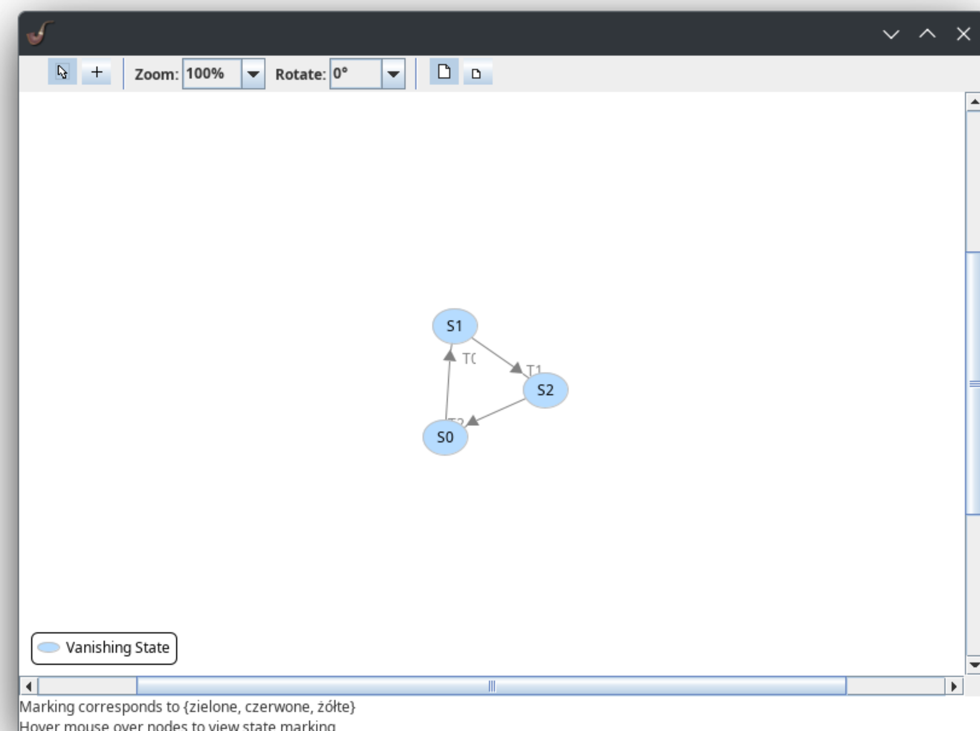
Jak wynika z *State Space Analysis*, ta sieć jest:

- Ograniczona
- Bezpieczna
- Nie zawiera deadlocku



3.1.2 Graf osiągalności

Graf osiągalności został wygenerowany i wygląda w sposób następujący:

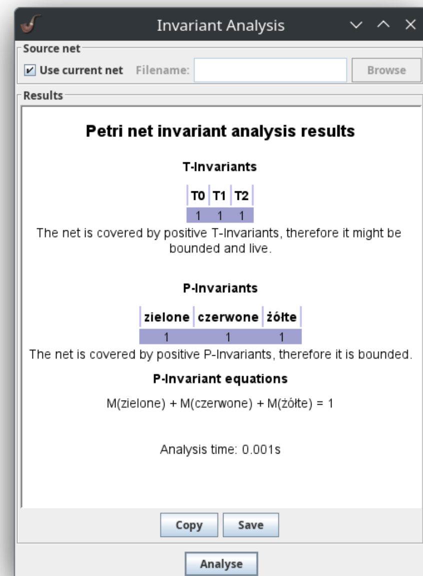


- Jak widać z powyższego grafu, **wszystkie znakowania** (stany, S_0, S_1, S_2) **są osiągalne**.

- W każdym ze znakowań maksymalna liczba znaczników (tokenów) wynosi 1. Cała ta maszyna stanów zawiera tylko 1 token, reprezentujący obecny stan. Przejście między poszczególnymi stanami odbywa się sekwencyjnie. A więc dana sieć jest **ograniczona** i **bezpieczna**.
- Każde przejście (T_0, T_1, T_2) jest przedstawione jako osobna krawędź w grafie. To świadczy o tym, że każde przejście ma szansę się wykonać - a więc **wszystkie przejścia są żywe**.
- Tak, wychodząc od dowolnego węzła grafu w k krokach można wykonać dowolne przejście. To świadczy o tym, że **sieć jest żywa** - każde przejście ma szansę się wykonać.

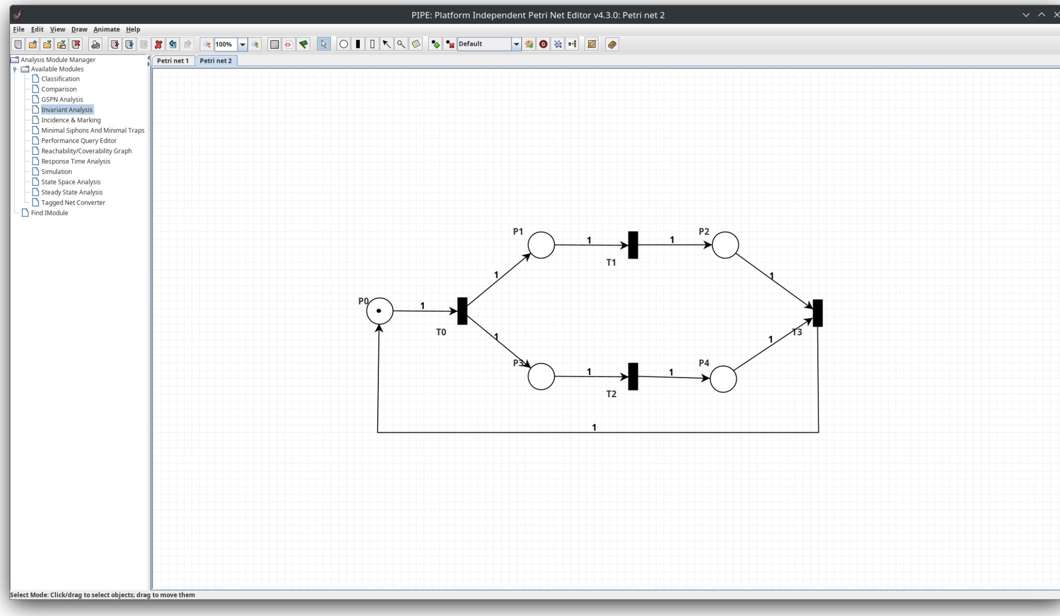
3.1.3 Analiza niezmienników

Wynik dokonanej analizy niezmienników jest widoczny na obrazku poniżej:



3.2 Zadanie 1

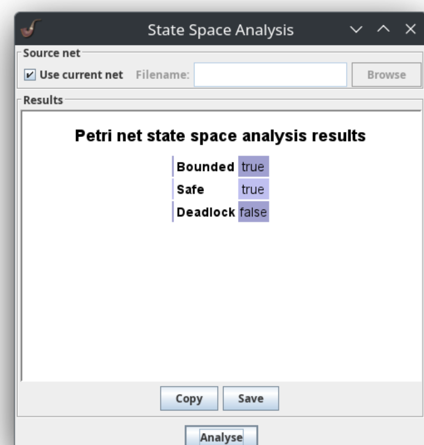
Sieć, reprezentująca równoległe wykonanie czynności:



3.2.1 Analiza stanów

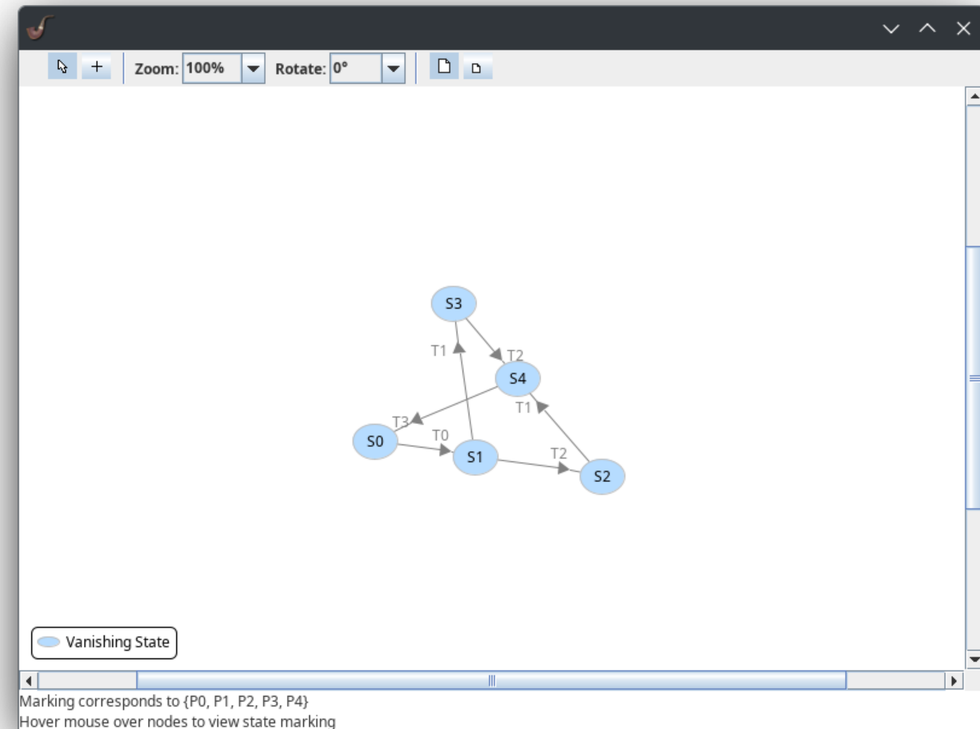
Jak wynika z *State Space Analysis*, ta sieć jest:

- Ograniczona
- Bezpieczna
- Nie zawiera deadlocku



3.2.2 Graf osiągalności

Graf osiągalności został wygenerowany i wygląda w sposób następujący:



W odróżnieniu od poprzedniego przypadku, w tym przypadku jeden stan (S_i) może reprezentować kilka miejsc (P_j) naraz (bo "zadania" wykonują się równolegle):

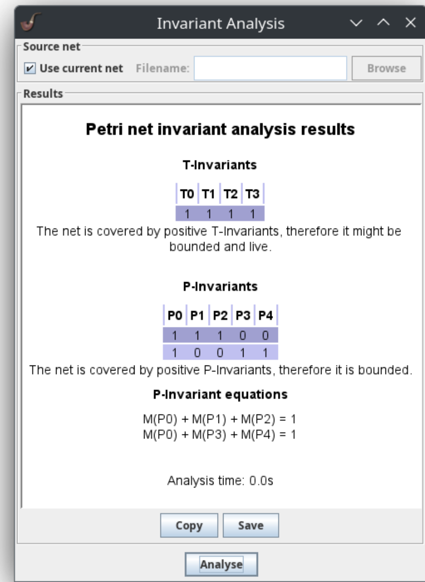
- $S_0 = \{P_0\}$
- $S_1 = \{P_1, P_3\}$
- $S_2 = \{P_1, P_4\}$
- $S_3 = \{P_2, P_3\}$
- $S_4 = \{P_2, P_4\}$

Ponadto:

- Jak widać z powyższego grafu, **wszystkie znakowania są osiągalne**.
- W każdym ze znakowań maksymalna liczba znaczników (tokenów) wynosi 1. Z tego wynika, że ta **sieć jest ograniczona i bezpieczna**.
- Każde przejście (T_0, T_1, T_2, T_3) jest przedstawione jako osobna krawędź (osobne krawędzie) w grafie. To świadczy o tym, że każde przejście ma szansę się wykonać - a więc **wszystkie przejścia są żywe**.
- Tak, wychodząc od dowolnego węzła grafu w k krokach można wykonać dowolne przejście. To świadczy o tym, że **sieć jest żywa** - każde przejście ma szansę się wykonać.

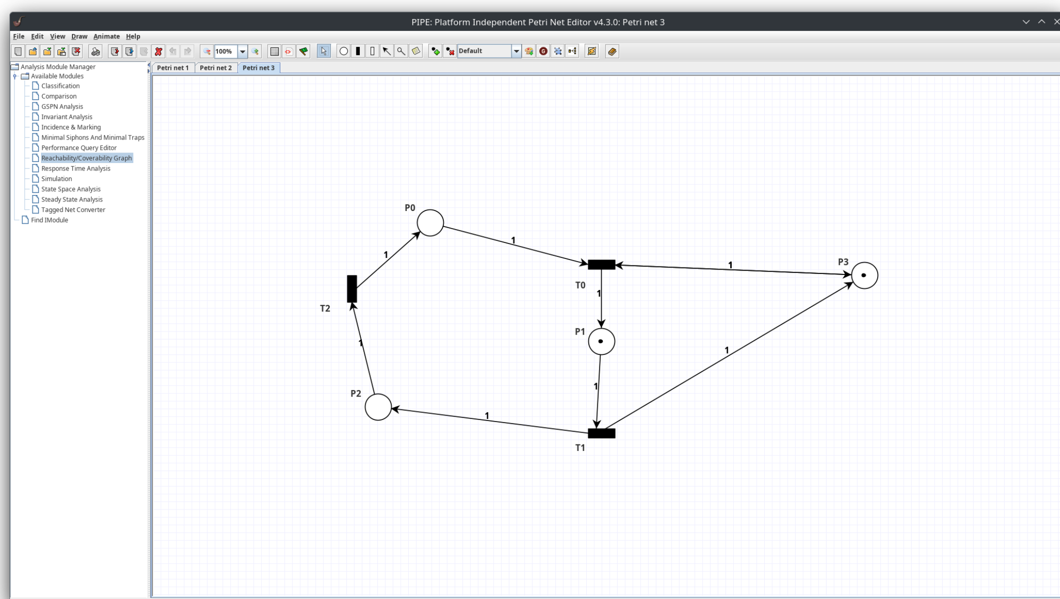
3.2.3 Analiza niezmienników

Wynik dokonanej analizy niezmienników jest przedstawiony na obrazku poniżej:



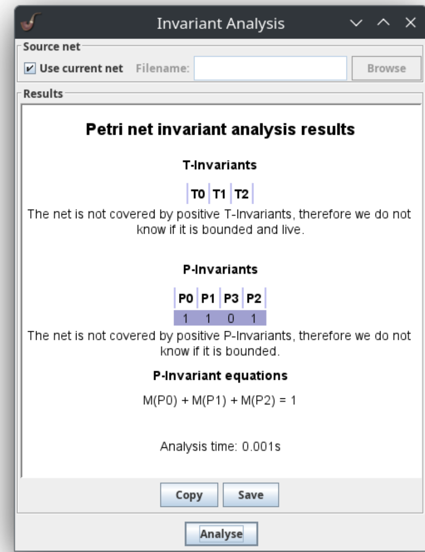
3.3 Zadanie 2

Sieć została narysowana w symulatorze w sposób następujący:



3.3.1 Analiza niezmienników

Wynik dokonanej analizy niezmienników jest przedstawiony na obrazku poniżej:

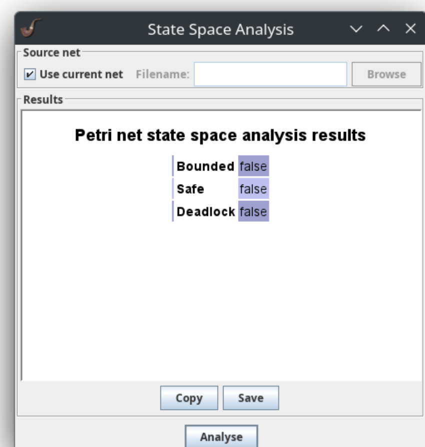


Na podstawie otrzymanych informacji nie można wyciągnąć żadnych wniosków co do tej sieci.

3.3.2 Analiza stanów

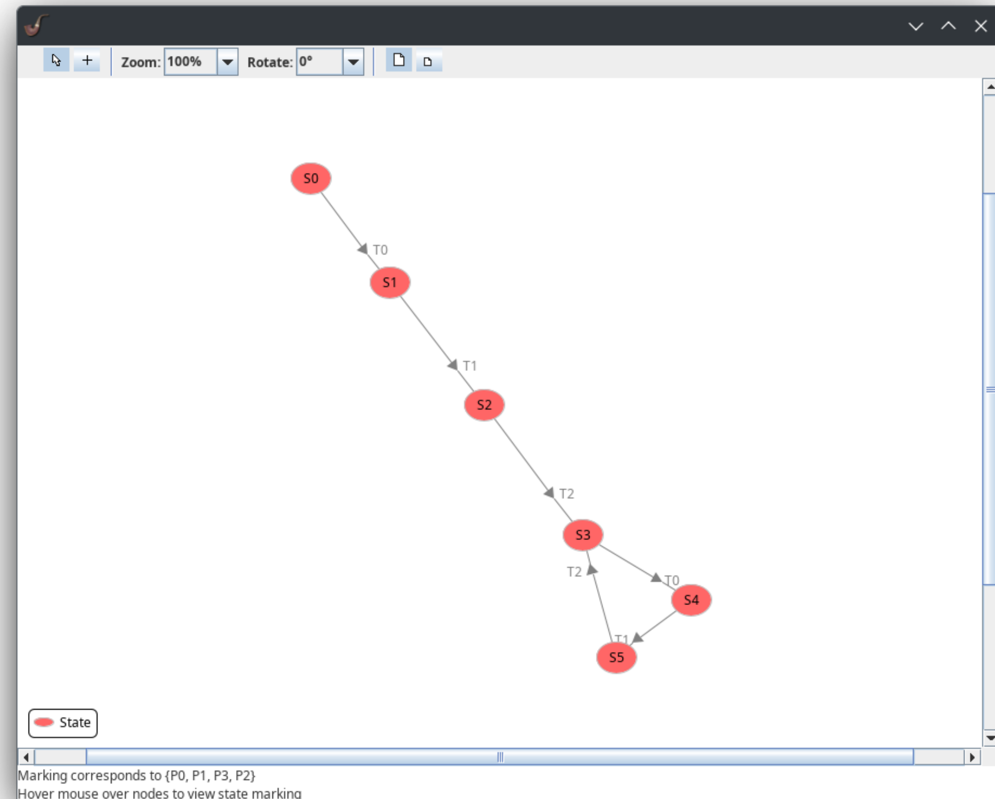
Z kolei analiza stanów pozwala stwierdzić, że sieć:

- Nie jest ograniczona
- Nie jest bezpieczna
- Nie zawiera deadlocku



3.3.3 Graf osiągalności

Został wygenerowany następujący graf osiągalności:

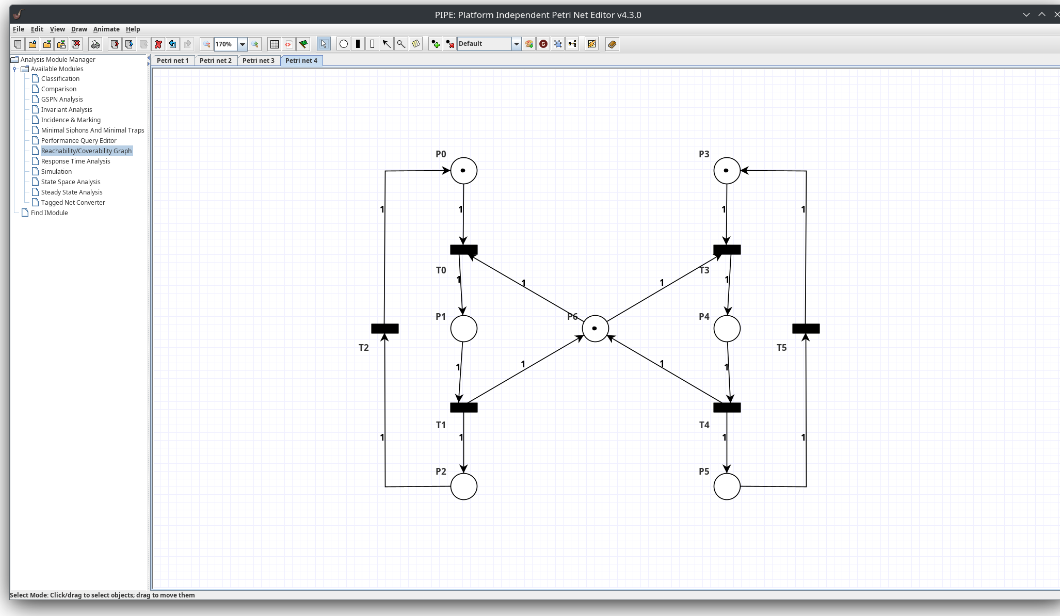


Co ciekawe, wygenerowanie tego grafu zajęło ponad 17 sekund.

Sieć jest żywa - wychodząc od dowolnego węzła grafu w k krokach można wykonać dowolne przejście (T_0, T_1, T_2). A więc każde przejście ma szansę się wykonać.

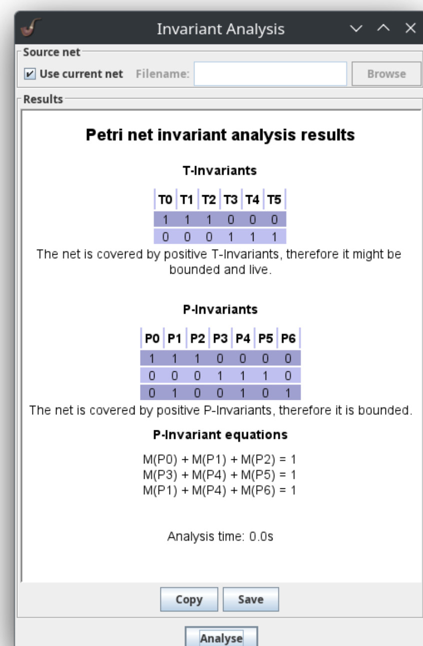
3.4 Zadanie 3

Sieć symulująca wykluczanie dwoch procesów na wspólnym zasobie wygląda w sposób następujący:



3.4.1 Analiza niezmienników

Wynik dokonanej analizy niezmienników jest przedstawiony na obrazku poniżej:

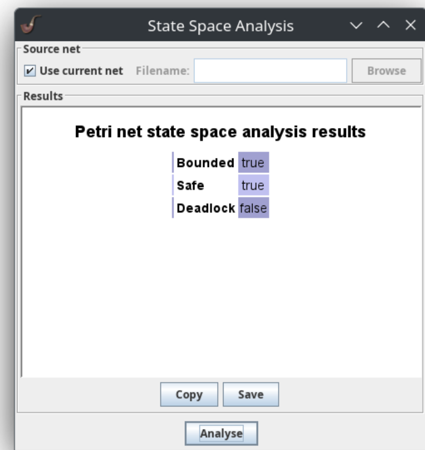


Ostatnie równanie, czyli $M(P_1)+M(P_4)+M(P_6) = 1$ pokazuje działanie ochrony sekcji krytycznej.

3.4.2 Analiza stanów

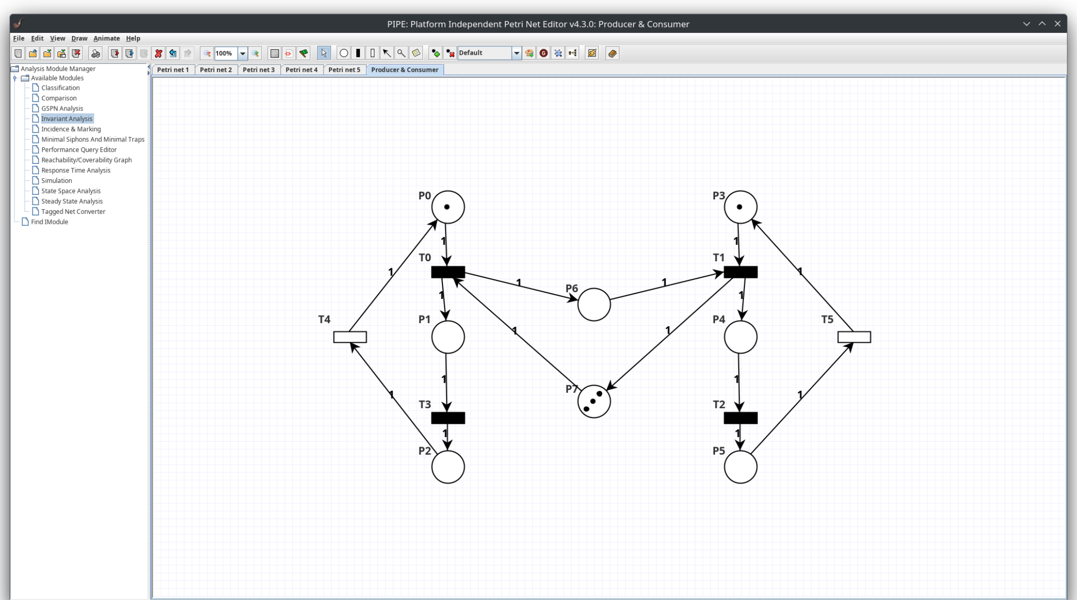
Jak wynika z *State Space Analysis*, ta sieć jest:

- Ograniczona
- Bezpieczna
- Nie zawiera deadlocku



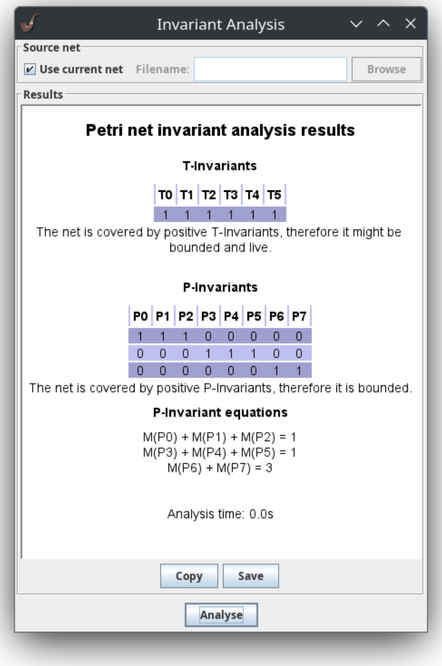
3.5 Zadanie 4

Sieć reprezentująca problem producenta i konsumenta z ograniczonym buforem wygląda następująco:



3.5.1 Analiza niezmienników

Wynik dokonanej analizy niezmienników jest przedstawiony na obrazku poniżej:



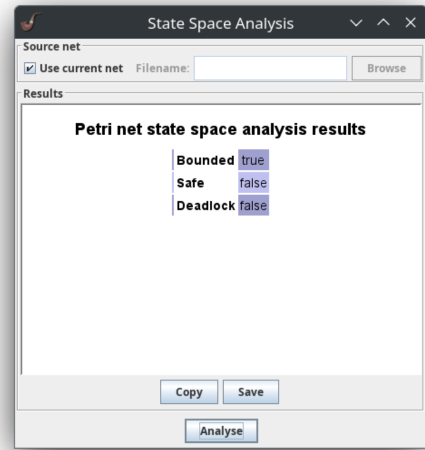
Ostatnie równanie, czyli $M(P_6) + M(P_7) = 3$ mówi nam o rozmiarze bufora.

Sieć jest zachowawcza - suma żetonów (tokenów, znaczników) pozostaje stała.

3.5.2 Analiza stanów

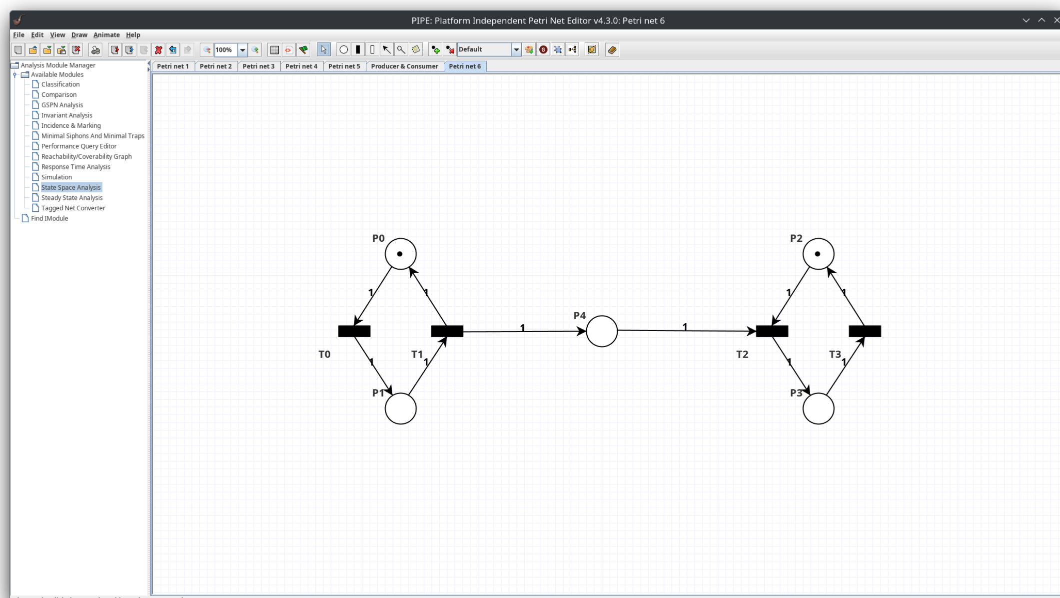
Jak wynika z *State Space Analysis*, ta sieć:

- Jest ograniczona
- Nie jest bezpieczna
- Nie zawiera deadlocku



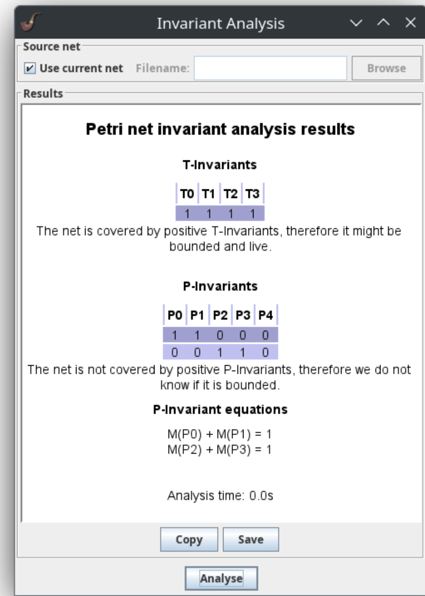
3.6 Zadanie 5

Sieć reprezentująca problem producenta i konsumenta z nieograniczonym buforem wygląda następująco:



3.6.1 Analiza niezmienników

Wynik dokonanej analizy niezmienników jest przedstawiony na obrazku poniżej:

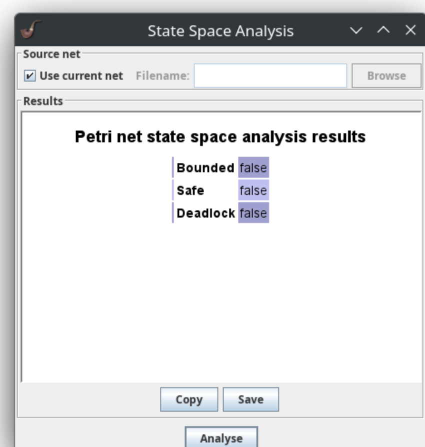


Obserwujemy brak pełnego pokrycia miejsc: miejsce P_4 nie występuje w żadnym równaniu.

3.6.2 Analiza stanów

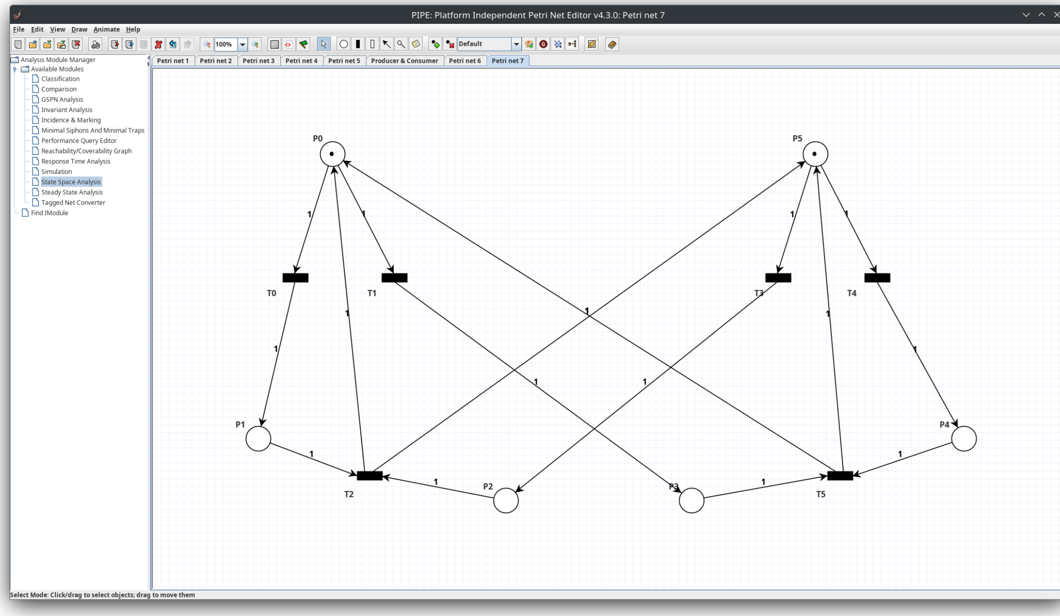
Jak wynika z *State Space Analysis*, ta sieć:

- Nie jest ograniczona
- Nie jest bezpieczna
- Nie zawiera deadlocku



3.7 Zadanie 6

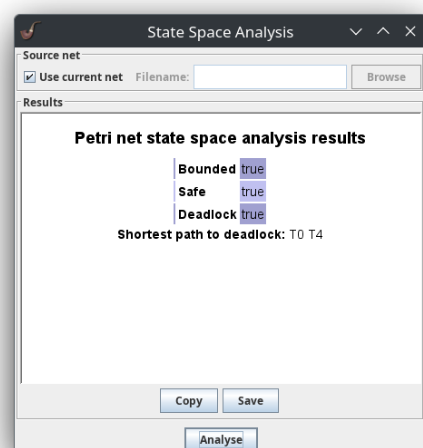
Przykład sieci z możliwością zakleszczenia wygląda w sposób następujący:



3.7.1 Analiza stanów

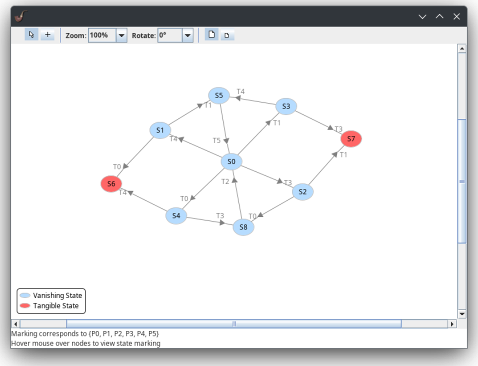
Jak wynika z *State Space Analysis*, ta sieć:

- Jest ograniczona
- Jest bezpieczna
- Zawiera deadlock (najkrótsza ścieżka prowadząca do deadlocku: $T_0 T_4$)



3.7.2 Graf osiągalności

Został wygenerowany następujący graf osiągalności:



Jak widać, ze stanów S_6 i S_7 nie można wykonać żadnych przejść (są zaznaczone na czerwono).

4 Wnioski

- **Sieci Petriego** to graficzny język do modelowania i analizy systemów dyskretnych i współbieżnych, który używa miejsc, przejść i żetonów.
- **Sieci Petriego mają różne własności**, takie jak ograniczoność, bezpieczeństwo, deadlock, zachowawczość, żywotność i odwracalność, które opisują ich zachowanie i możliwości.
- **Graf osiągalności** to graf, który przedstawia wszystkie możliwe znakowania sieci Petriego i przejścia między nimi.
- **Znaczniki** to elementy graficzne, które reprezentują dane, zasoby lub stany systemu. Znaczniki mogą być przemieszczane pomiędzy miejscami poprzez przejścia.
- **Niezmienniki** to równania lub wektory, które opisują zachowanie sieci Petriego niezależnie od znakowania. Niezmienniki są używane do weryfikacji własności sieci Petriego, takich jak zachowawczość czy ograniczoność.
- **Sieci Petriego mają wiele zastosowań** w różnych dziedzinach, takich jak automatyka, bioinformatyka, inżynieria oprogramowania, programowanie równoległe i inne.

5 Bibliografia

1. Materiały do laboratorium 12, dr inż. Włodzimierz Funika:
   https://home.agh.edu.pl/~funika/tw/lab-petri/
2. Sieci Petriego, dr inż. Jędrzej Ułasiewicz:
   http://jedrzej.ulasiewicz.staff.iiar.pwr.wroc.pl/ProgramowanieWspolbiezne/wyklad/Sieci-Petriego15.pdf
3. Platform Independent Petri net Editor 2 (PIPE2):
   https://pipe2.sourceforge.net/