



Teoria Współbieżności:

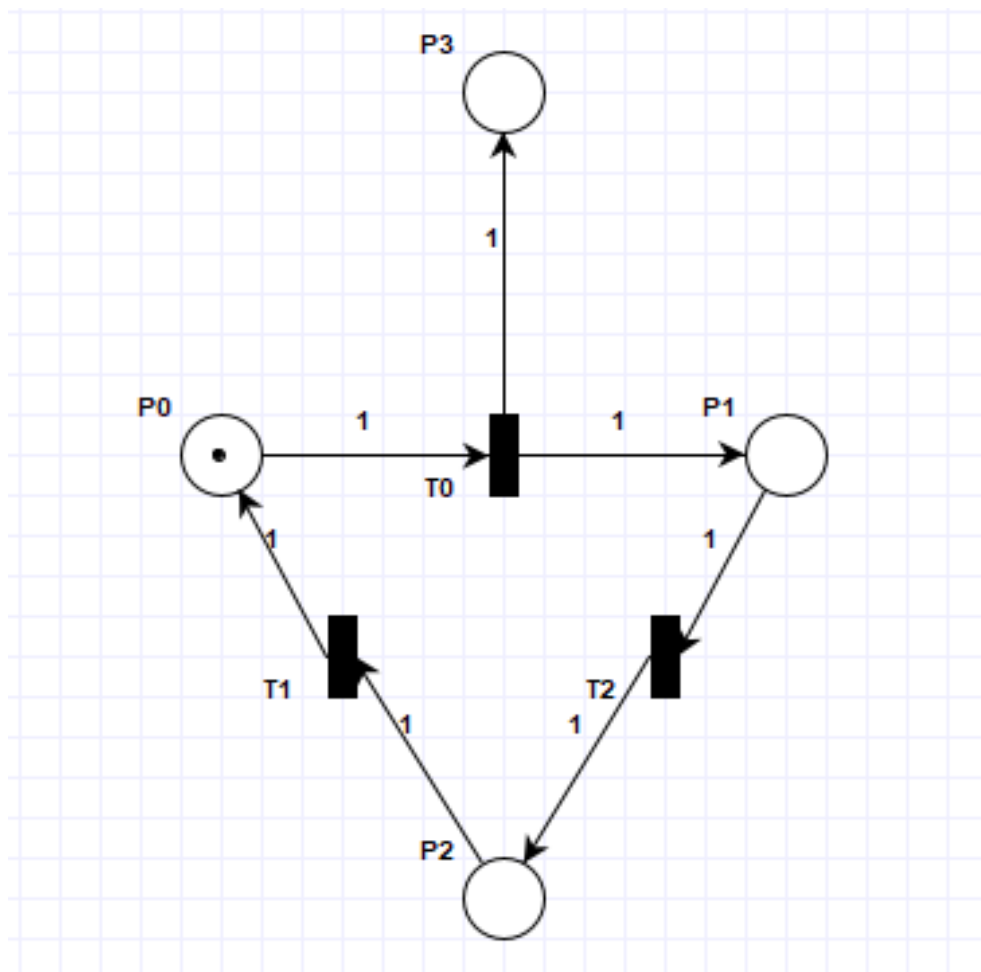
Zadanie domowe - Sieci Petri

Autor: Krzysztof Solecki

## Zadanie 1:

Wymyślić własną maszynę stanów, zasymulować przykład i dokonać analizy grafu osiągalności oraz niezmienników.

Schemat sieci:

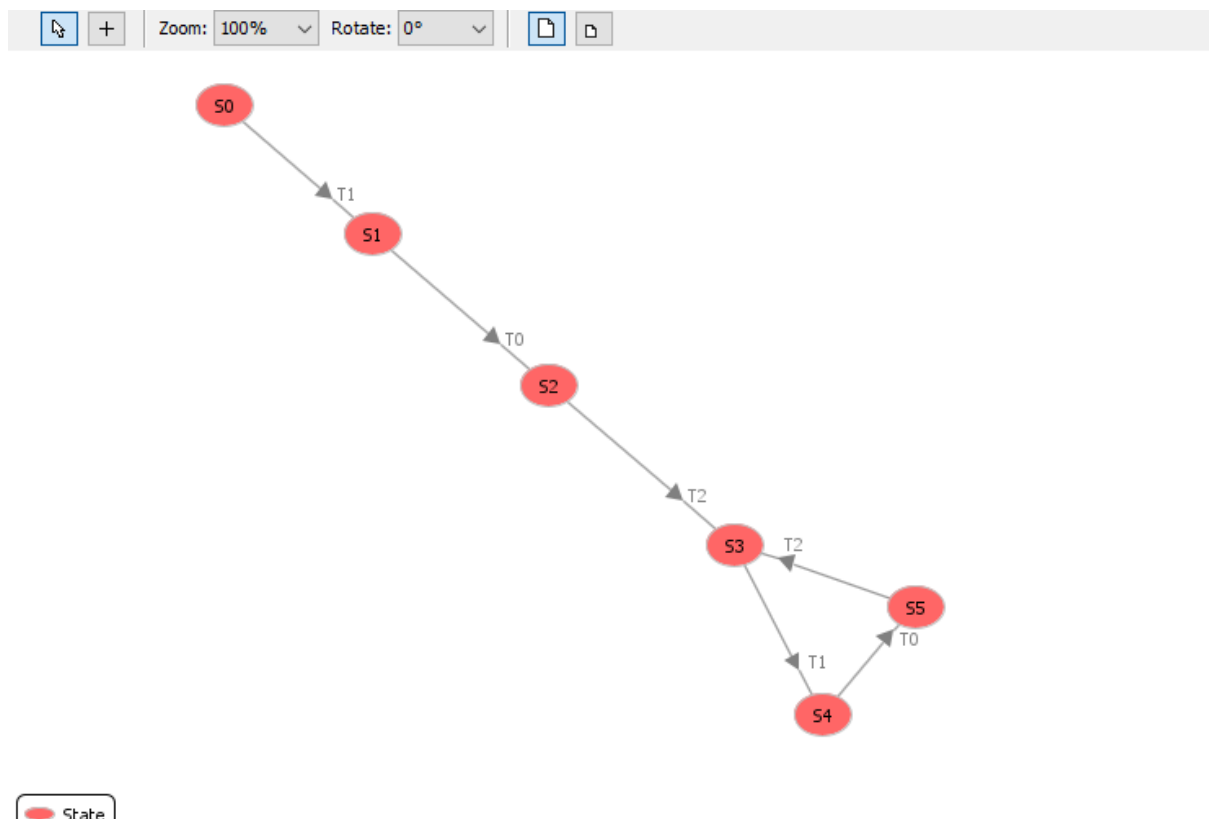


Zbudowana przeze mnie sieć to cykl wraz z licznikiem liczby przejść przez cykl. Każdy token w P3 oznacza jeden cykl który się wykonał.

Przykładowa symulacja animacji w programie:

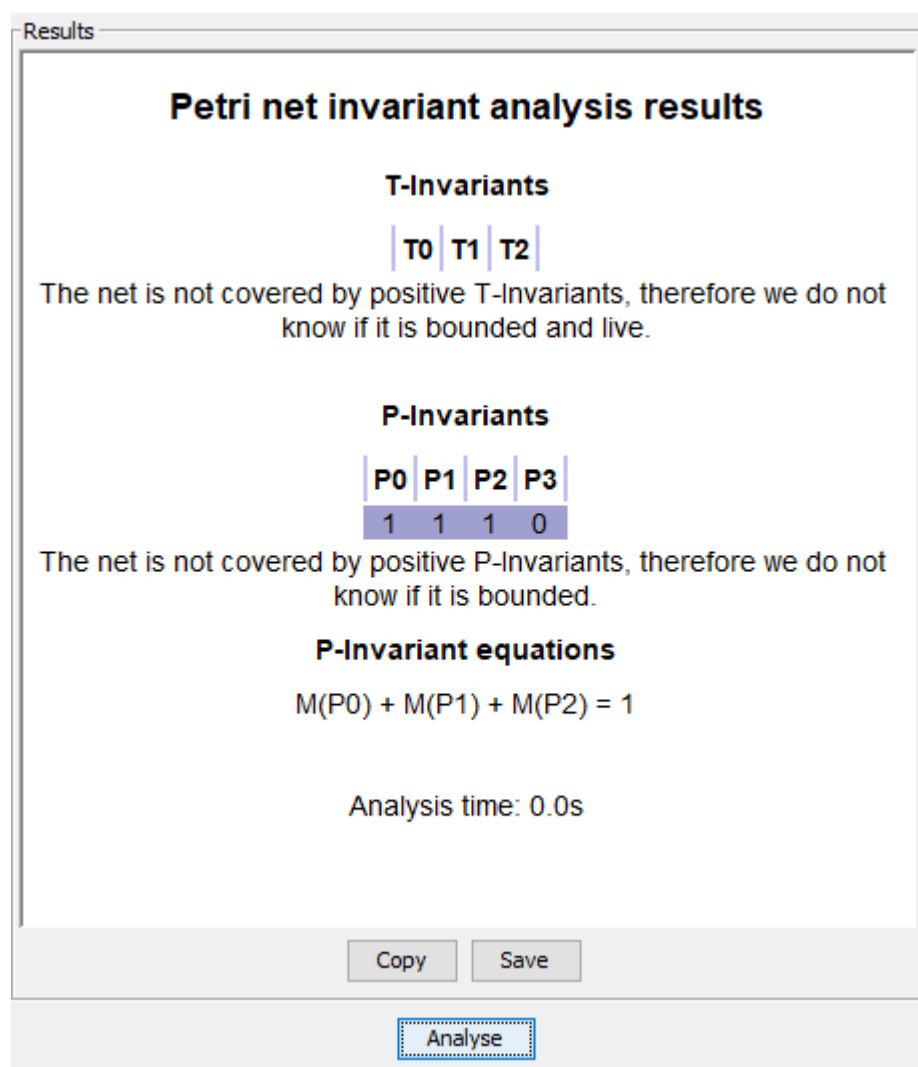
Animation history	
Initial Marking	
T0	
T2	
T1	
T0	
T2	
T1	
T0	
T2	
T1	
T0	

Graf osiągalności:



Graf pokazuje, że cykl działa prawidłowo a w P3 stale rośnie liczba tokenów.

Niezmienniki:



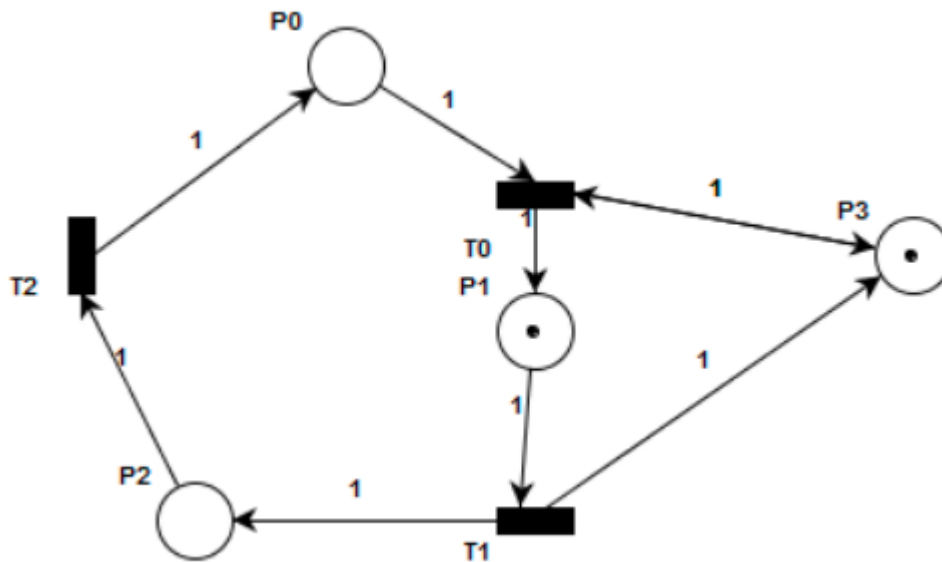
Z analizy niezmienników wynika, że sieć nie jest odwracalna. Wynika to z tego, że po rozpoczęciu pierwszego cyklu w liczniku pojawia się pierwszy token i z każdym kolejnym cyklem liczba tokenów rośnie i nie powróci do 0, które było w markowaniu początkowym.

Z tego wynika również, że sieć nie jest ograniczona, bo P3 nie jest ograniczone.

Dodatkowo z równania niezmienników miejsc widzimy, że w cyklu jest na raz tylko jeden token, czyli nie ma możliwości by dwa cykły wykonywały się na raz.

## Zadanie 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. Objasnić wniosek.

Oto przykładowa symulacja wykonana przez program:

### Animation history

Initial Marking

T1

T2

T0

T1

T2

T0

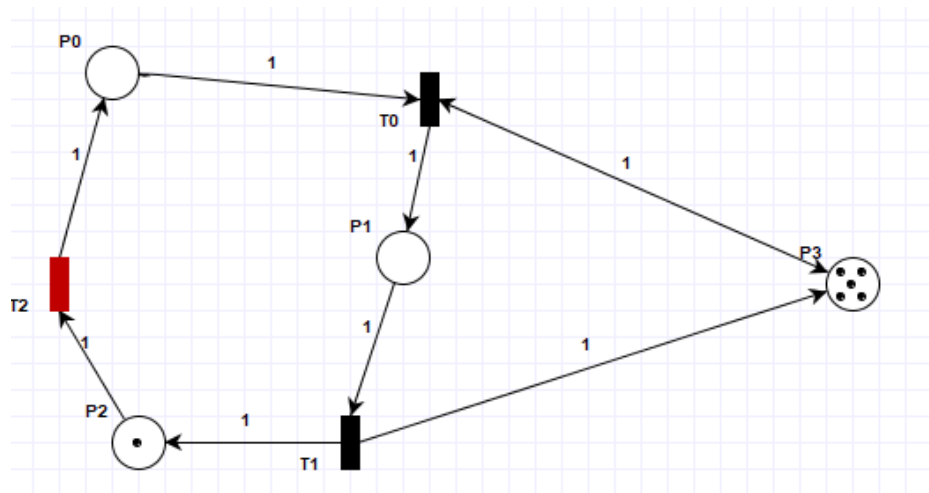
T1

T2

T0

T1

Po jej wykonaniu w miejscu P3 znajdowało się 5 znaczników.



Niezmienniki przejść:

Invariant Analysis

Source net

☒ Use current net

Filename:

Browse

Results

**Petri net invariant analysis results**

**T-Invariants**

**T0 T1 T2**

The net is not covered by positive T-Invariants, therefore we do not know if it is bounded and live.

**P-Invariants**

**P0 P1 P2 P3**

**1 1 1 0**

The net is not covered by positive P-Invariants, therefore we do not know if it is bounded.

**P-Invariant equations**

$$M(P0) + M(P1) + M(P2) = 1$$

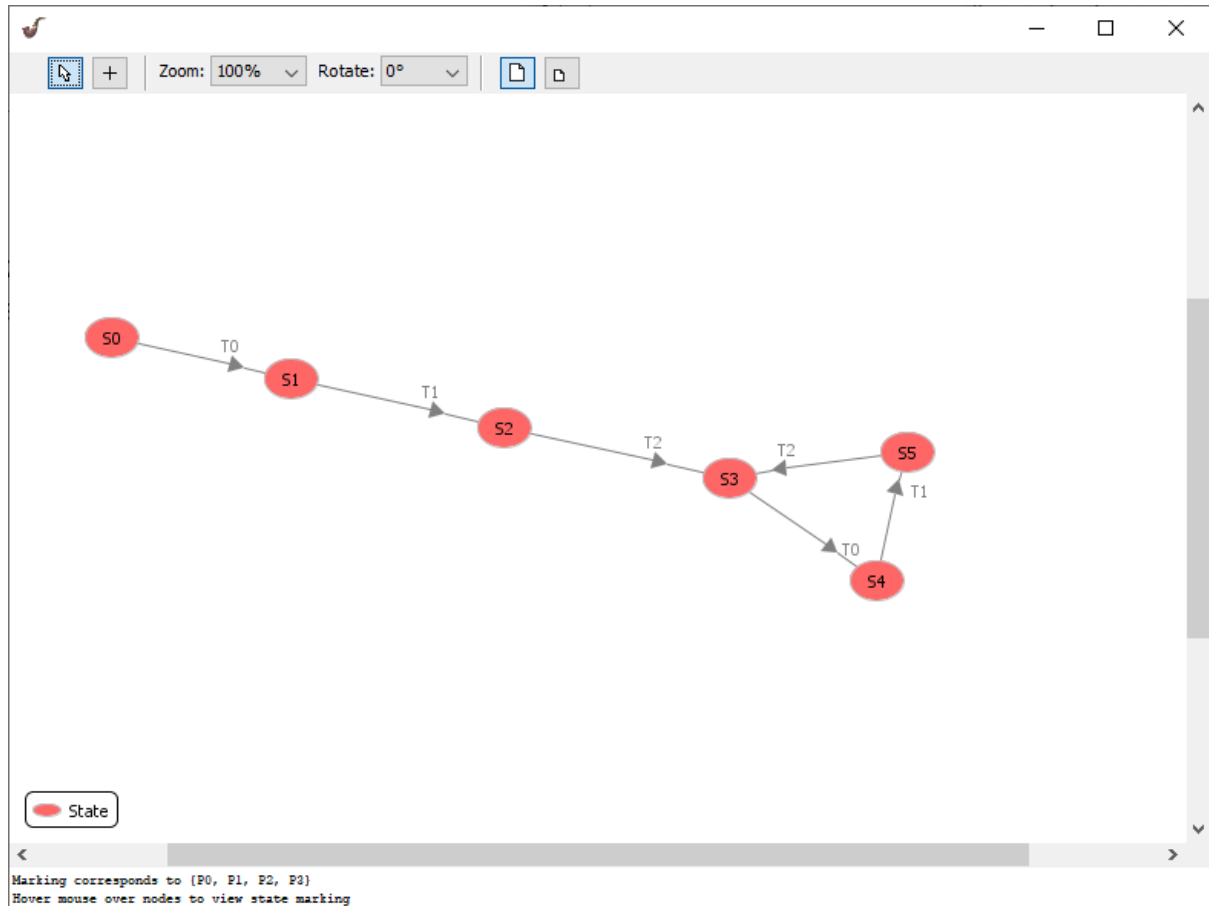
Analysis time: 0.001s

Copy Save

Analyse

Brak niezmienników przejść świadczy o tym, że sieć nie jest odwracalna, a także nie posiada żadnego znakowania własnego.

Graf osiągalności:

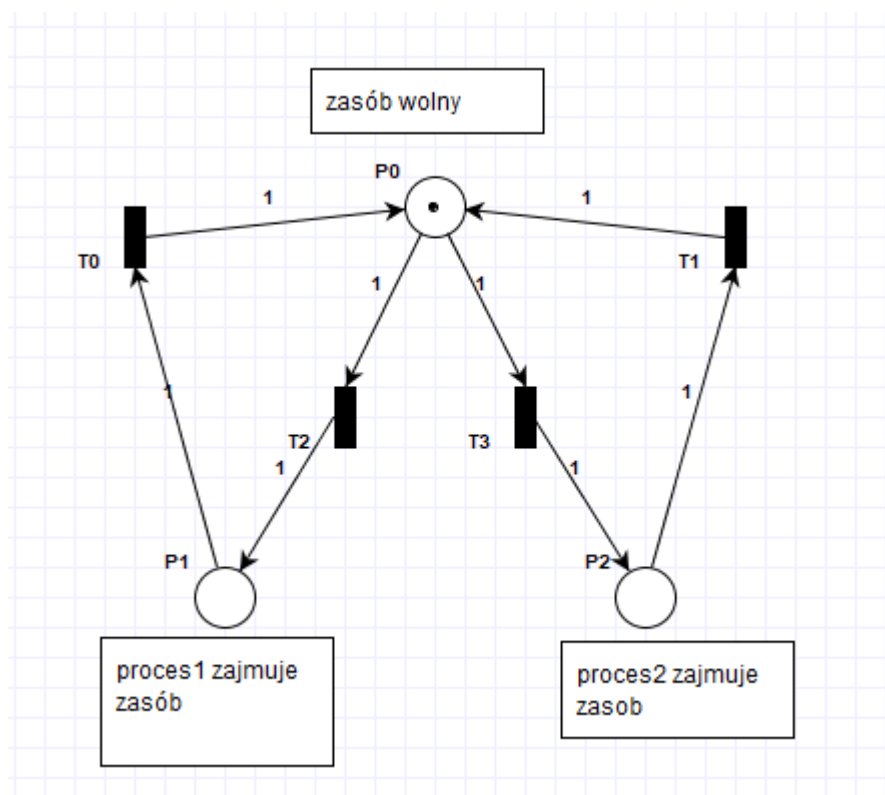


Sieć jest żywa, gdyż począwszy od stanu początkowego zostanie uruchomione każde przejście. Nie występuje żadne martwe przejście. Sieć nie jest ograniczona gdyż miejsce P3 nie jest ograniczone. Jak widać w grafie sieć wpada w nieskończony cykl między stanami S3, S4, S5 i liczba tokenów w P3 będzie rosła w nieskończoność.

### Zadanie 3:

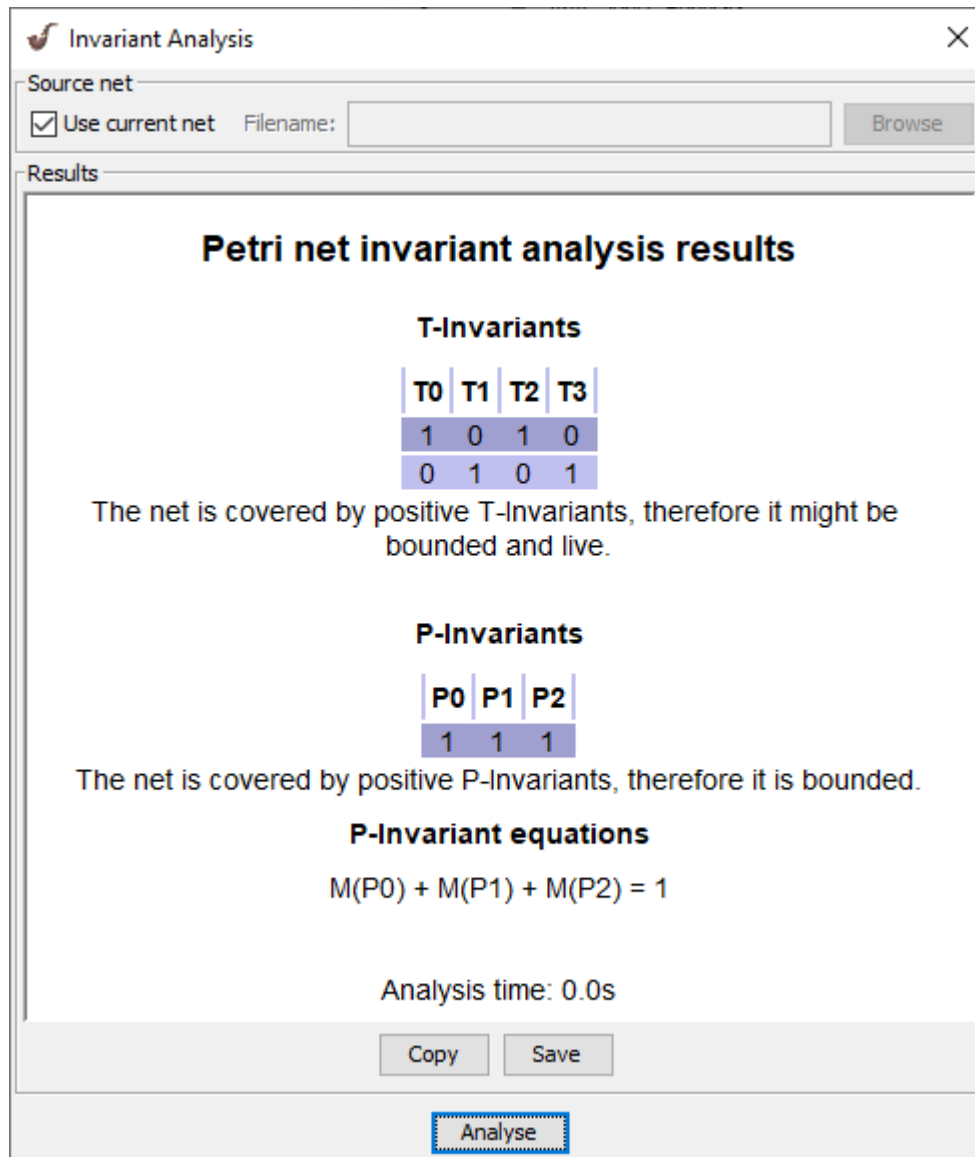
Zasymulować wzajemne wykluczanie dwóch procesów na wspólnym zasobie. Dokonać analizy niezmienników miejsc oraz wyjaśnić znaczenie równań (P-invariant equations). Które równanie pokazuje działanie ochrony sekcji krytycznej?

Schemat sieci:





Niezmienniki:

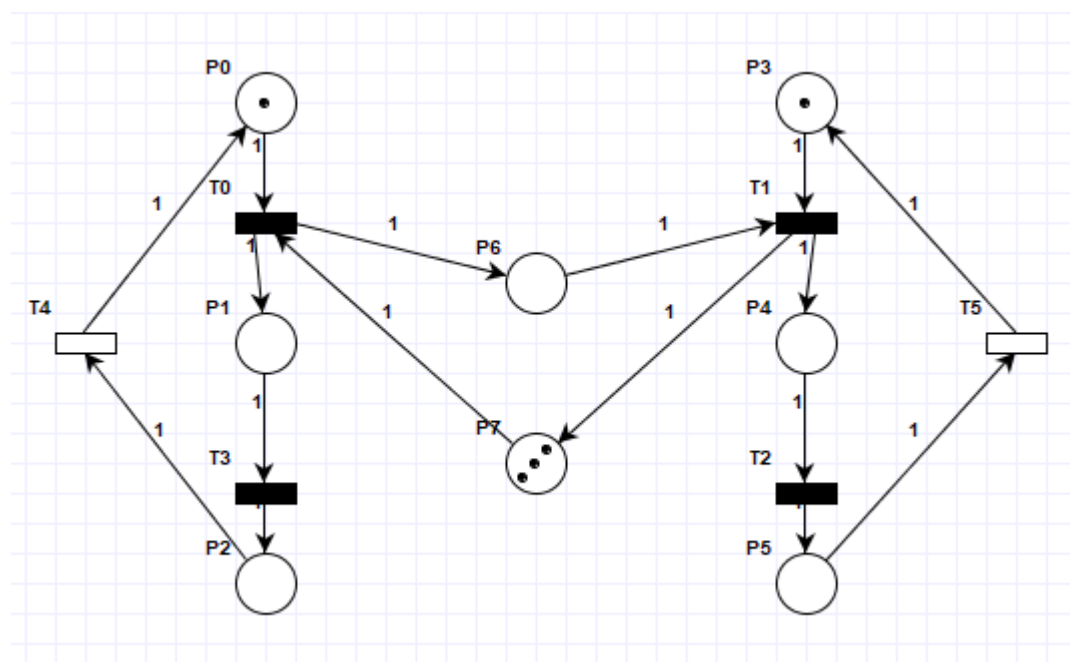


Równanie pokazuje, że suma znaczników we wszystkich trzech miejscach jest stała (i równa 1). Jako że miejsca odpowiadają stanom zasobu (wolny, zajęty przez proces 1, zajęty przez proces 2), równanie to wskazuje na niemożliwość zajęcia zasobu jednocześnie przez oba procesy, a na tym właśnie polega ochrona sekcji krytycznej.

## Zadanie 4:

Uruchomić problem producenta i konsumenta z ograniczonym buforem (można posłużyć się przykładem, menu: file, examples). Dokonać analizy niezmienników. Czy sieć jest zachowawcza? Które równanie mówi nam o rozmiarze bufora?

Schemat sieci:



Niezmienniki:

The screenshot shows a software window titled "Invariant Analysis". It has a "Source net" section with a checked "Use current net" option and a "Filename:" field with a "Browse" button. The "Results" section displays the following:

**Petri net invariant analysis results**

**T-Invariants**

T0	T1	T2	T3	T4	T5
1	1	1	1	1	1

The net is covered by positive T-Invariants, therefore it might be bounded and live.

**P-Invariants**

P0	P1	P2	P3	P4	P5	P6	P7
1	1	1	0	0	0	0	0
0	0	0	1	1	1	0	0
0	0	0	0	0	0	1	1

The net is covered by positive P-Invariants, therefore it is bounded.

**P-Invariant equations**

$$M(P0) + M(P1) + M(P2) = 1$$
$$M(P3) + M(P4) + M(P5) = 1$$
$$M(P6) + M(P7) = 3$$

Analysis time: 0.0s

At the bottom, there are "Copy" and "Save" buttons, and a highlighted "Analyse" button.

Zachowawczość sieci:

Każda tranzycja w sieci ma tyle samo miejsc wejściowych i wyjściowych, czyli liczba znaczników w sieci nie zmienia się, w związku z czym sieć jest zachowawcza.

Rozmiar bufora:

O rozmiarze bufora mówi równanie:

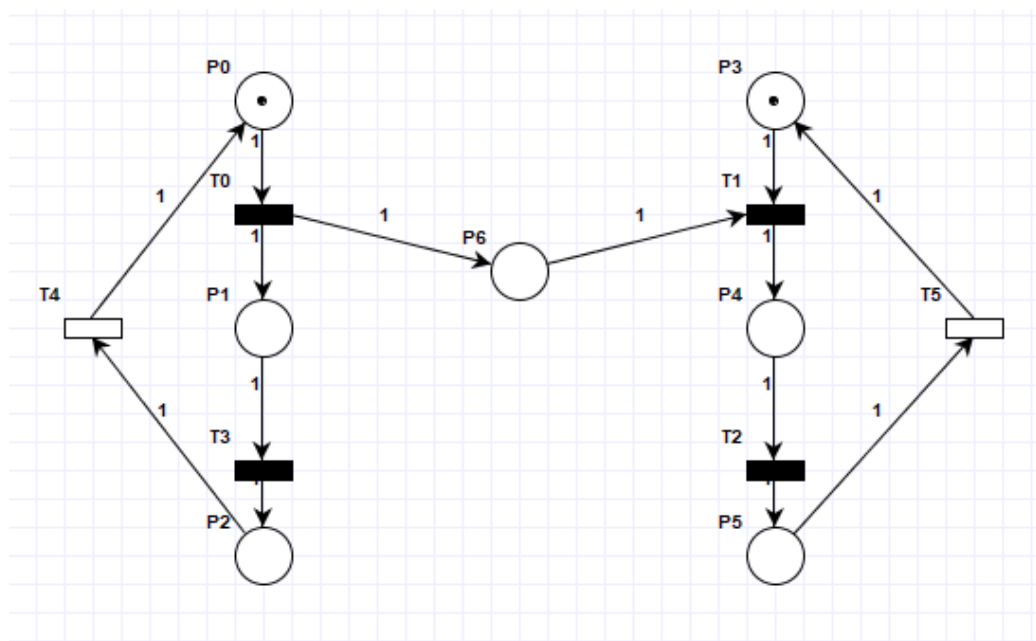
$$M(P6) + M(P7) = 3$$

Miejsce P7 reprezentuje wolne miejsca w buforze, a P6 - miejsca zajęte

## Zadanie 5:

Stworzyć symulację problemu producenta i konsumenta z nieograniczonym buforem. Dokonać analizy niezmienników. Zaobserwować brak pełnego pokrycia miejsc.

Schemat sieci:



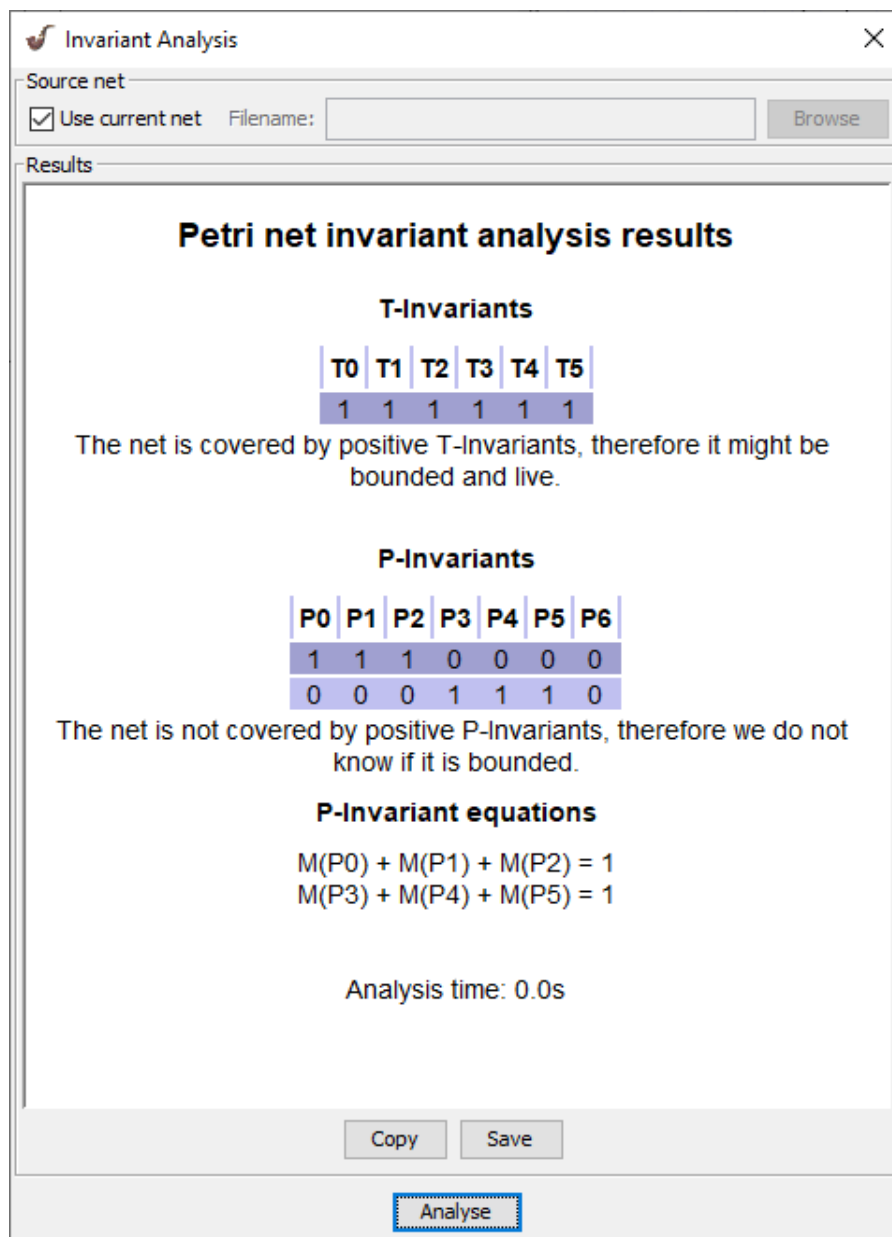
To poprzednia sieć zmodyfikowana o usunięcie miejsca P7 (i zmienienie tranzycji na natychmiastowe)

Przykładowa animacja wykonana przez program:

Animation history
Initial Marking
T3
T4
T0
T1
T3
T2
T4
T0
T3
T4
T0
T3
T4
T0
T3
T4
T0
T3
T4
T0
T3
T4
T0
T3
T5
T1
T2
T4
T0
T3

Po jej zakończeniu w buforze (P6) znajduje się 6 elementów. Ich liczba nie jest ograniczona.

Niezmienniki:



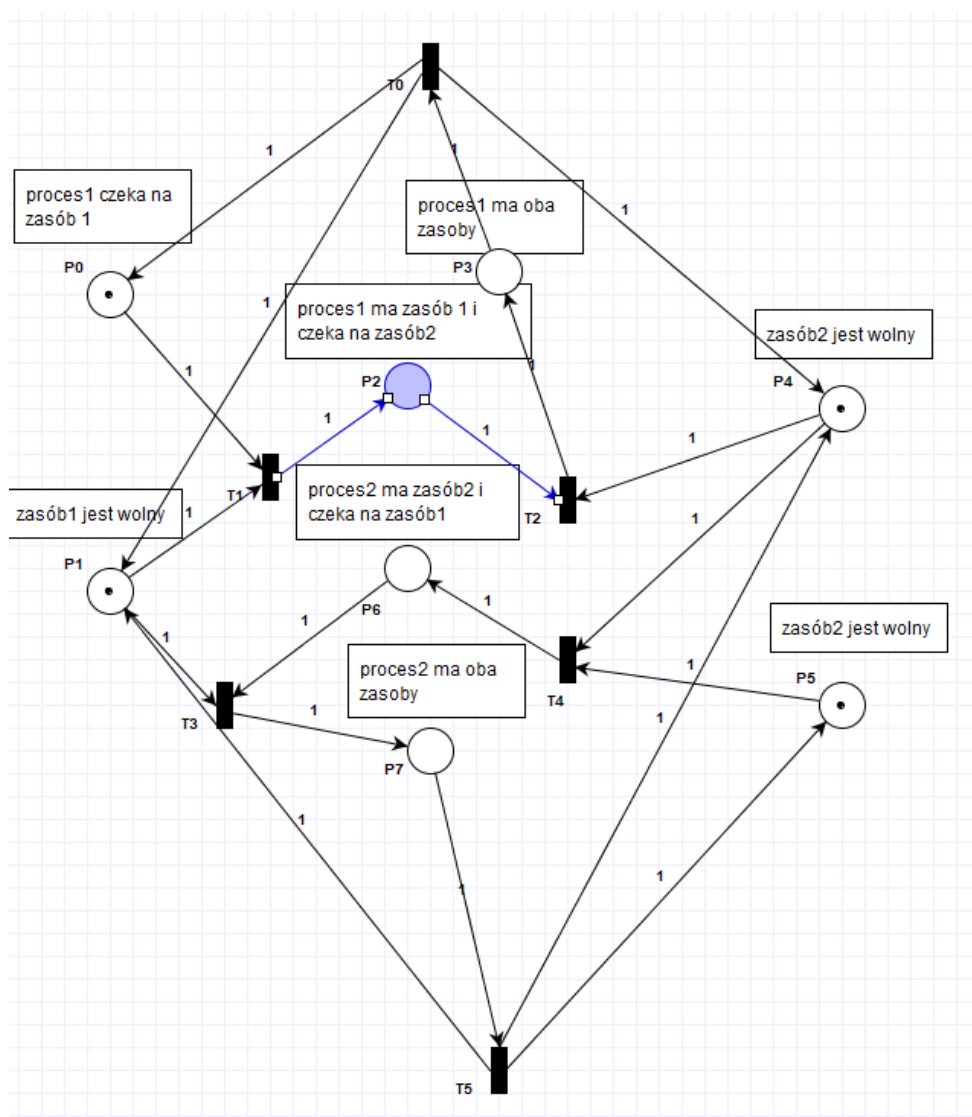
Brak pełnego pokrycia miejsc:

Jako że sieć jest nieograniczona (możliwość składowania dowolnej liczby elementów w buforze - możliwość uzyskania dowolnej liczby markowań w miejscu P6), to nie może być pokryta niezmiennikami miejsc. W szczególności nieograniczone miejsce P6 nie będzie należeć do nośnika żadnego z nich.

## Zadanie 6:

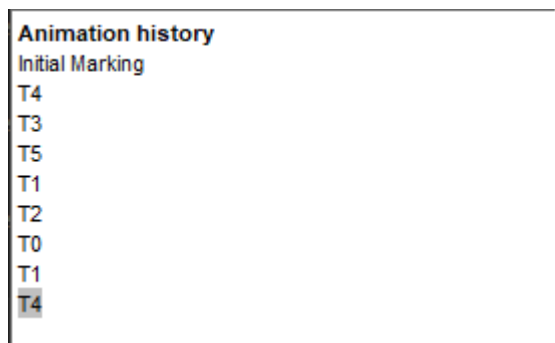
Zasymulować prosty przykład ilustrujący zakleszczenie.  
Wygenerować graf osiągalności i zaobserwować znakowania, z których nie można wykonać przejść. Zaobserwować właściwości sieci w "State Space Analysis".

Schemat sieci:

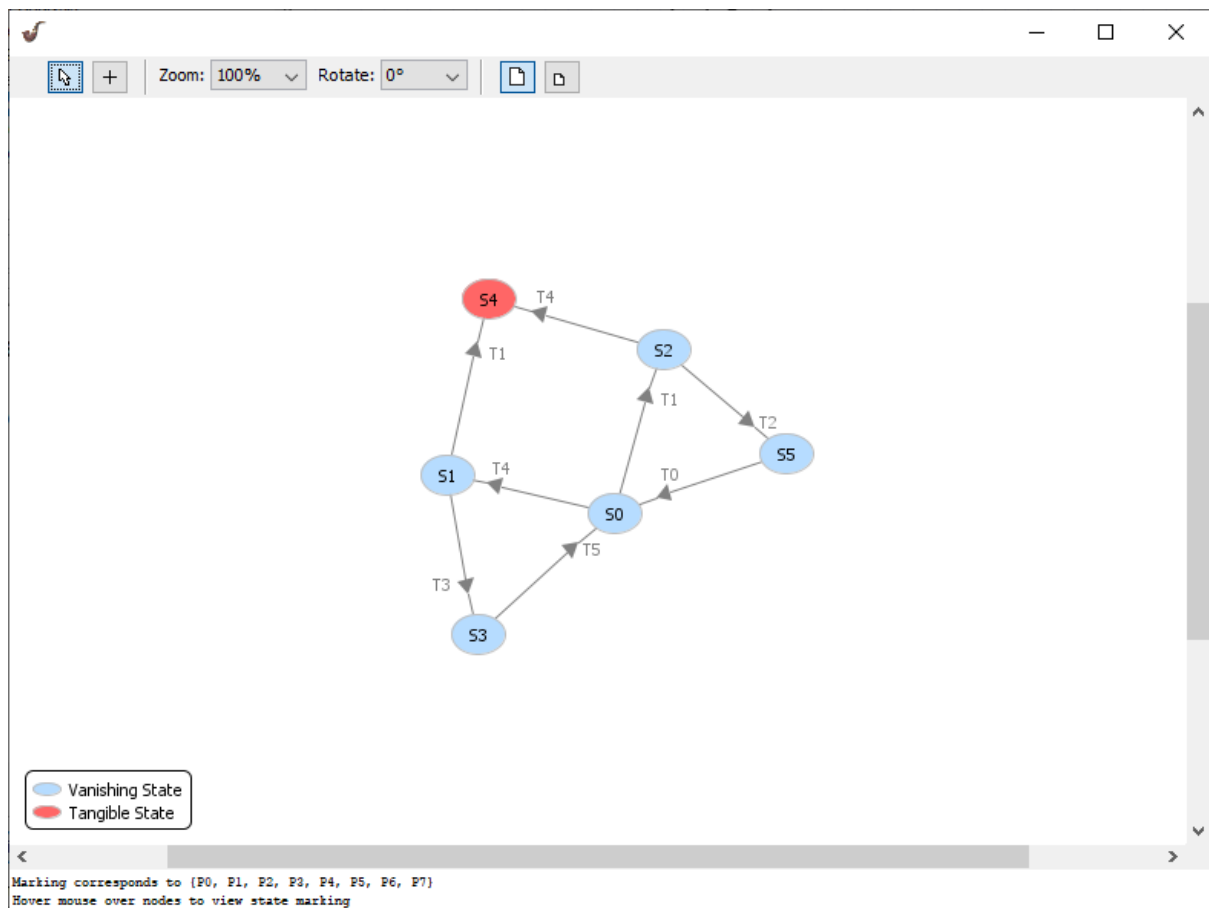


Wybrany przykład to konkurencja 2 procesów o 2 zasoby, przy czym procesy próbują je zająć w innej kolejności.

Przykładowa animacja (kończąca się deadlockiem):



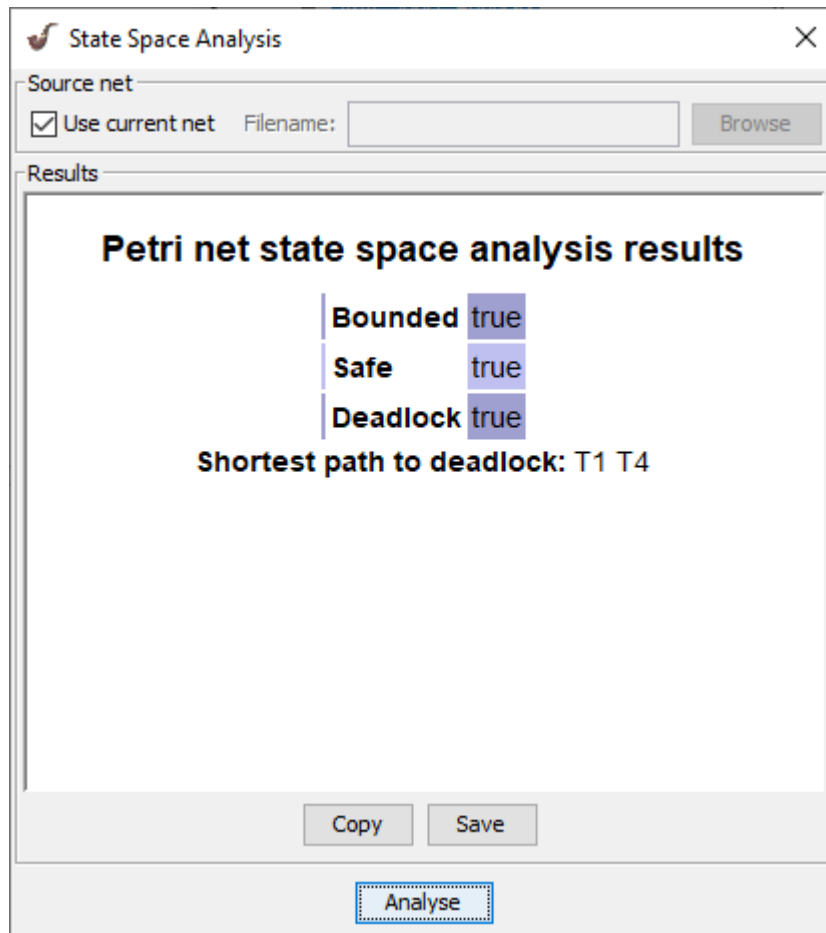
Graf osiągalności:



Znakowanie oznaczone jako S4 to to, w którym miejsca oznaczone jako 'proces1 ma zasób1 i czeka na zasób2' i 'proces2 ma zasób2 i czeka na zasób1' mają po 1 znaczniku. Z tego znakowania nie można wykonać żadnego przejścia.



## State Space Analysis:



Nie da się uzyskać więcej niż 1 znacznika w jakimkolwiek miejscu, więc sieć jest 1-ograniczona, czyli bezpieczna. Można natomiast, jak pokazano, doprowadzić do zakleszczenia.