Stanisław Denkowski
305288

Teoria Współbieżności

Zadanie domowe 7

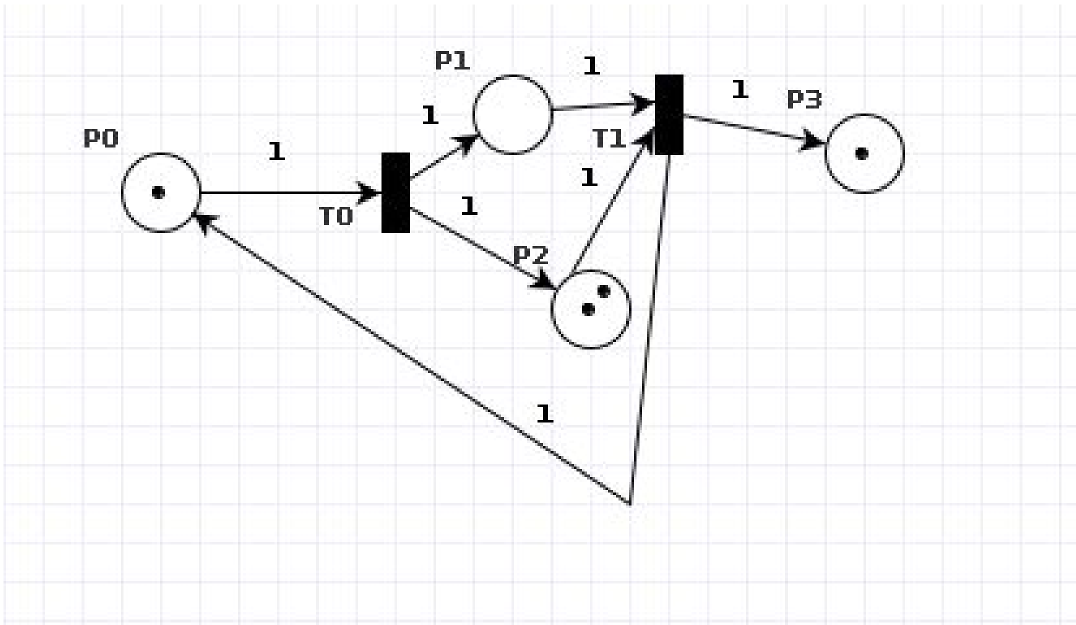
Przykłady modelowania i analizy systemów współbieżnych
z wykorzystaniem sieci Petri

Sprawozdanie

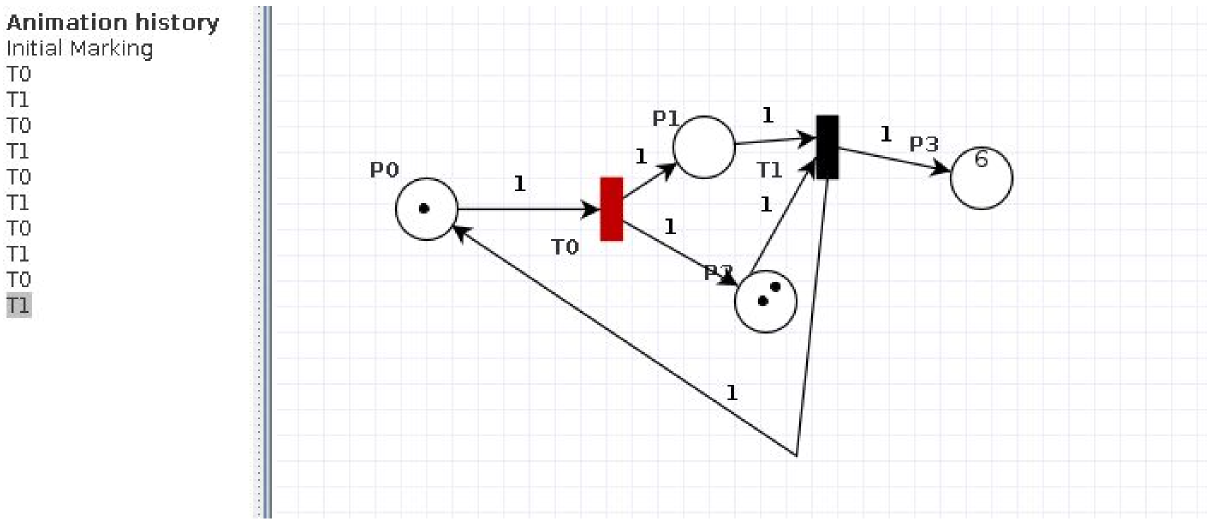
Zadanie 1

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

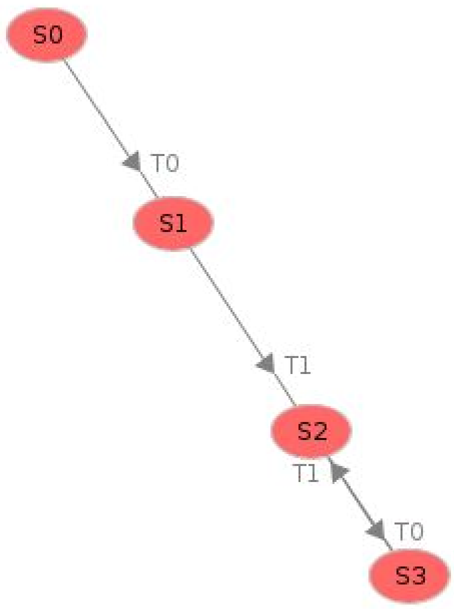
Przykład sieci:



Symulacja 10 kroków:



Graf przejść:



Graf osiągalności ma cykl, do którego prowadzi jedna ścieżka i nie ma już możliwości powrotu.

Niezmienniki:

Petri net invariant analysis results

T-Invariants

T0	T1

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	0	0
1	0	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) = 1
M(P0) + M(P2) = 3

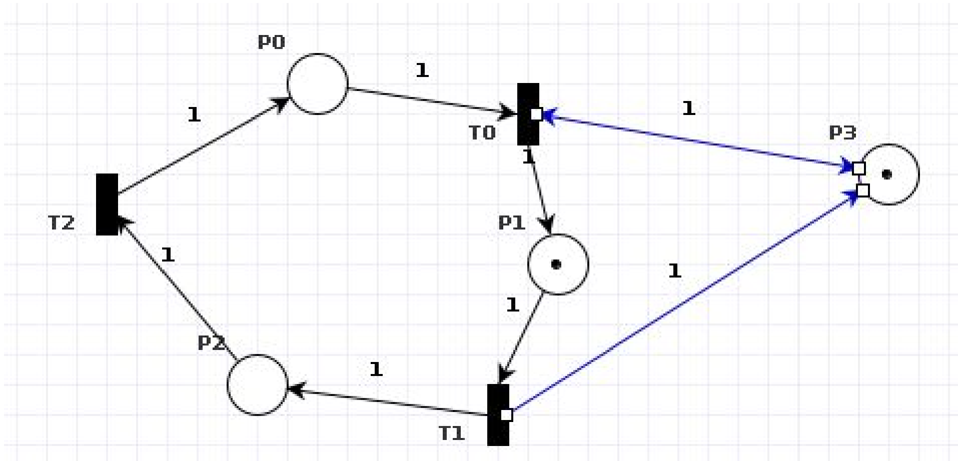
Analysis time: 0.001s

Sieć nie jest ograniczona, ale ma 2 równania niezmienników, mówiące, że w stanach P0 i P1 jest dokładnie jeden token (tylko w jednym z nich na raz, ale się zmienia) oraz w stanach P0 i P2 są w sumie dokładnie 3 tokeny (jeden jest w P0 i dwa w P2 lub 3 w P2).

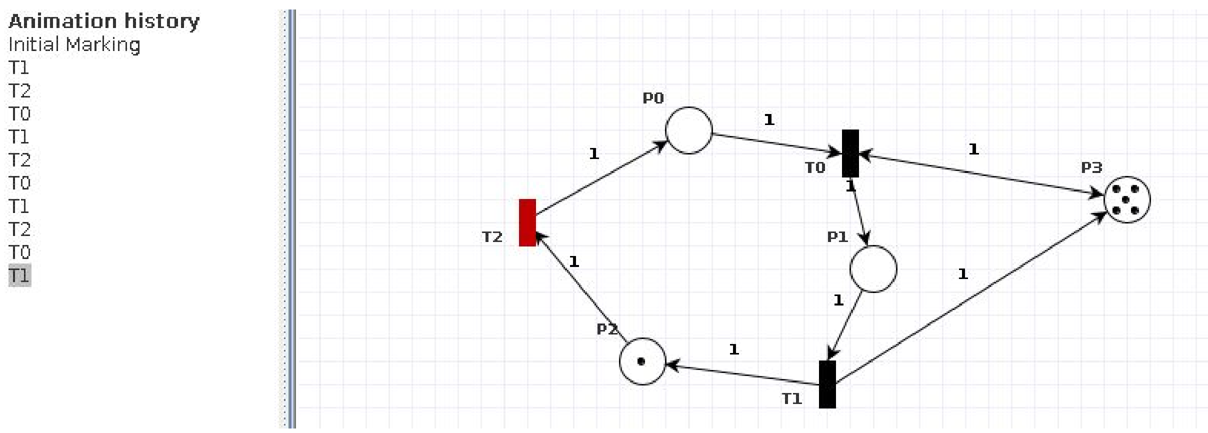
Zadanie 2

Zasymulować daną sieć, dokonać analizy niezmienników przejść, wnioski o odwracalności, graf osiągalności, wywnioskować z grafu czy sieć jest żywa lub ograniczona. Objaśnić wniosek.

Dana sieć:



Symulacja 10 ruchów w sieci:



Niezmienniki przejść:

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.

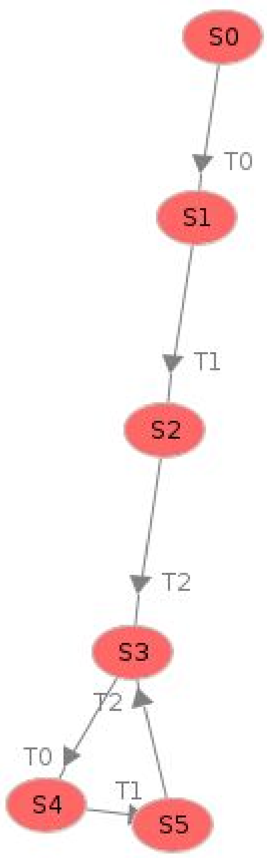
P-Invariant equations

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

Analysis time: 0.001s

Skoro nie ma niezmienników przejść, to sieć nie jest ani odwracalna, ani nie ma znakowania własnego. Jedynym niezmiennikiem jest to, że zawsze jeden token będzie się znajdować w którymś ze stanów P0, P1 lub P2.

Graf osiągalności:



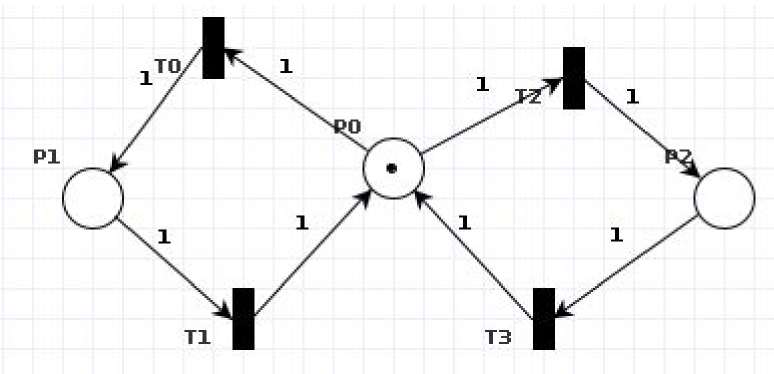
Oznacza to, że graf osiągalności jest nieskończony i został uproszczony do tej postaci. Istnieje nieskończony cykl, który za każdym razem generuje nam nowe tokeny i wykorzystuje wszystkie tranzycje, więc sieć jest żywotna.

Sieć nie jest ograniczona, ponieważ w stanie P3 liczba tokenów może wzrastać bez ograniczeń.

Zadanie 3

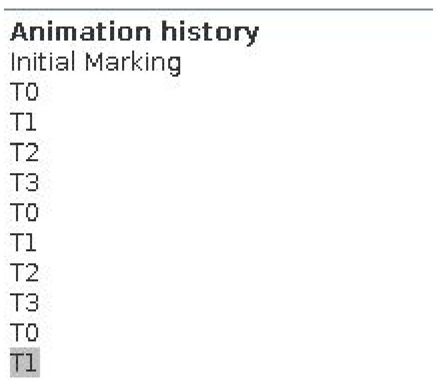
Zasymulować wzajemne wykluczanie dwóch procesów na wspólnym zasobie. Dokonać analizy niezmienników miejsc oraz wyjaśnić znaczenie równań (P-invariant equations). Które równanie pokazuje działanie ochrony sekcji krytycznej?

Przykład sieci:



P0 - Bufor (token jest wspólnym zasobem, który w tym momencie znajduje się w P0).
P1- Zasób wykorzystywany przez pierwszy proces.
P2 - Zasób wykorzystywany przez drugi proces.

Symulacja 10 kroków:


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

Niezmienniki:

Petri net invariant analysis results

T-Invariants

T0	T1	T2	T3
1	1	0	0
0	0	1	1

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

P-Invariants

P0	P1	P2
1	1	1

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

P-Invariant equations

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

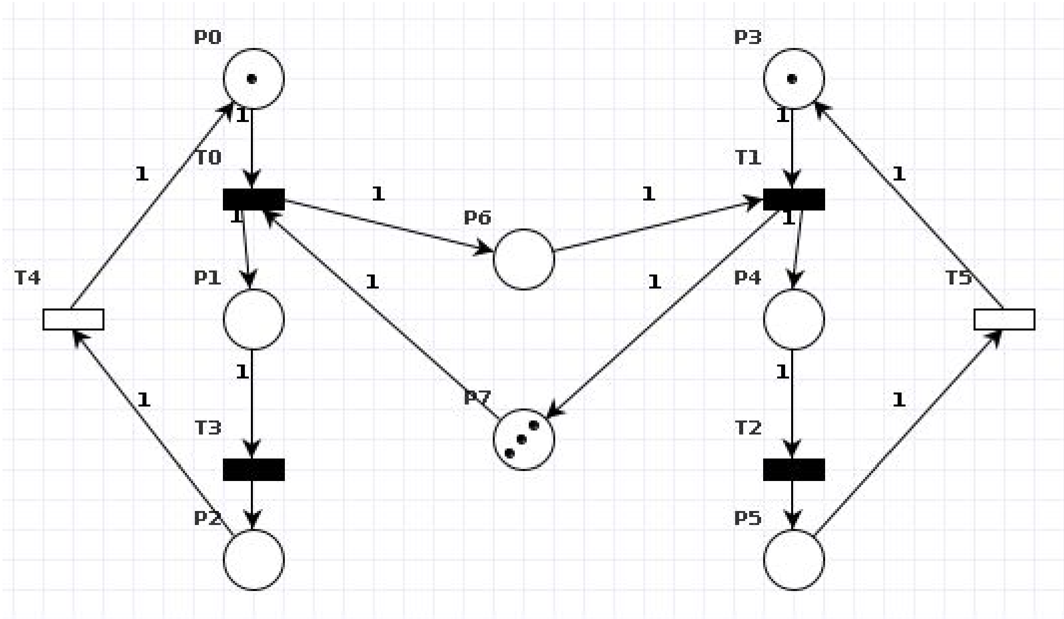
Analysis time: 0.001s

Jedyne równanie M(P0) + M(P1) + M(P2) = 1, pokazuje że w żadnych dwóch stanach (wolnym, zajętym przez pierwszy proces lub zajętym przez drugi proces) nie będzie zasobu na raz, ponieważ sumuje się to do jedynki i jest stale 1 - w jednym z tych 3 stanów jest dokładnie jeden token i pozostałe 2 nie mają tokena, zmienia się natomiast gdzie ten token się znajduje. Ochrona sekcji krytycznej polega na udostępnieniu wolnego zasobu tylko jednemu z procesów, co pokazuje to równanie.

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?

Przykład sieci:



Po bokach znajdują się producent i konsument - możemy przyjąć, że po lewej jest producent, po prawej konsument.
W środku znajduje się wspólny bufor z 3 tokenami - tokeny u dołu odpowiadają liczbie wolnych elementów, które producent może wykorzystać przygotowując zasób dla konsumenta, natomiast tokeny u góry odpowiadają liczbie przygotowanych elementów przez producenta, gotowych do wykorzystania przez konsumenta.

Symulacja 10 ruchów:

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

Niezmienniki:

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

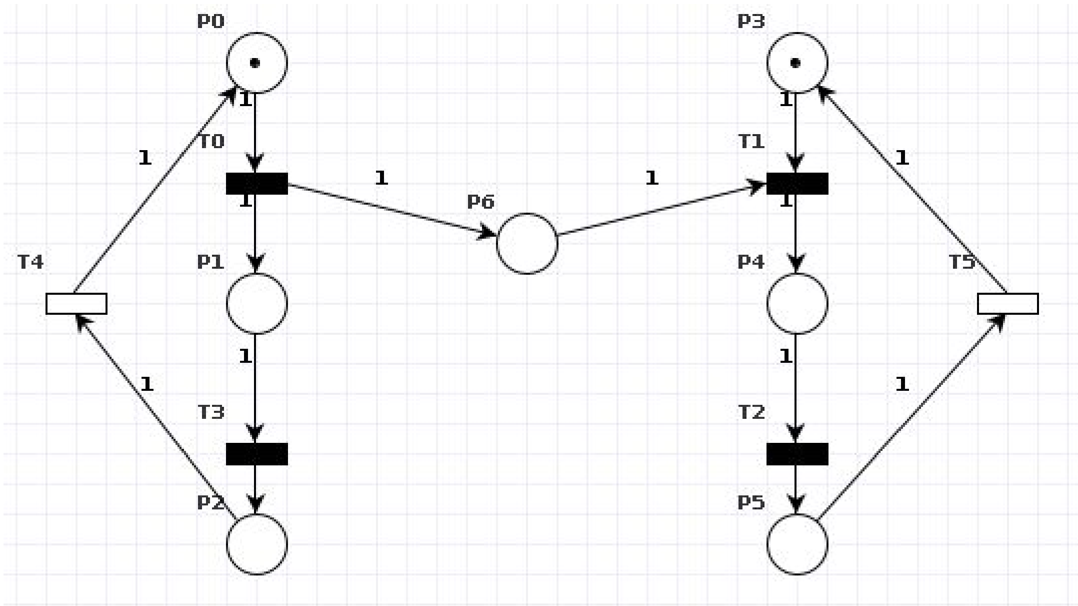
Analysis time: 0.002s

Sieć jest zachowawcza, ponieważ każda tranzycja ma taki sam rozmiar wejścia, co wyjścia i nie wzrośnie nam liczba znaczników.
Trzecie równanie M(P6) + M(P7) = 3, mówi nam o rozmiarze bufora (P6 - dla konsumenta, P7 - dla producenta).

Zadanie 5

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

Przykład sieci:



Sieć jest zmodyfikowaną wersją sieci z poprzedniego podpunktu - usunęliśmy stan P7, który odpowiadał za liczbę oczekujących na przygotowanie przez producenta elementów. Skoro teraz nie ma takiego ograniczenia, sieć teraz ma nieograniczony bufor.

Symulacja 10 kroków:

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

Niezmienniki:

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
1	1	1	0	0	0	0
0	0	0	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
M(P3) + M(P4) + M(P5) = 1

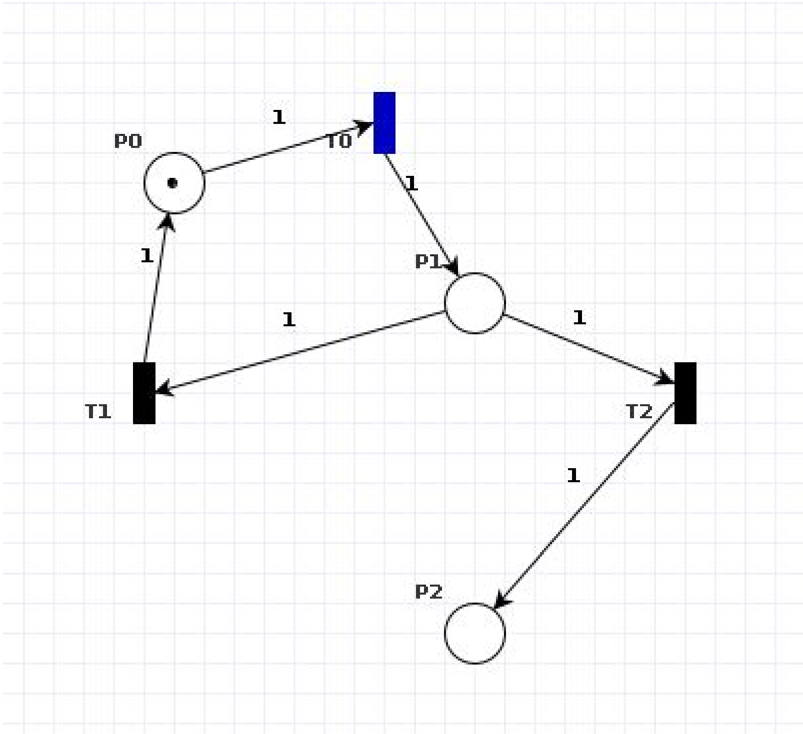
Analysis time: 0.002s

Sieć jest nieograniczona - bufor P6 jest nieograniczony.
Skoro sieć jest nieograniczona, to nie może być pokryta niezmiennikami miejsc

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”.

Przykład sieci:

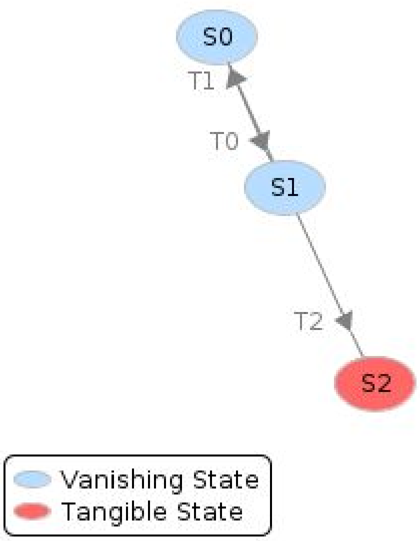


Jeśli token wejdzie do stanu P2, to już z niego nie wyjdzie, gdy inne stany na niego czekają - mamy deadlock.

Symulacja 10 kroków (po 4 już mamy deadlock):

Animation history
Initial Marking
T0
T1
T0
T2

Graf osiągalności:



Tangible state oznacza stan, z którego nie wykonamy już następnego kroku - mamy deadlock. S0 i S1 to cykl, z którego możemy przejść do S2 i już nie ma powrotu.

State space analysis:

Petri net state space analysis results

Bounded	true
Safe	true
Deadlock	true

Shortest path to deadlock: T0 T2

Sieć jest ograniczona - nigdy nie będziemy mieć więcej niż jeden token, jest 1-ograniczona, więc jest też bezpieczna. Do zakleszczenia jest łatwo doprowadzić, wykorzystując tylko 2 tranzycje.