Lab 12

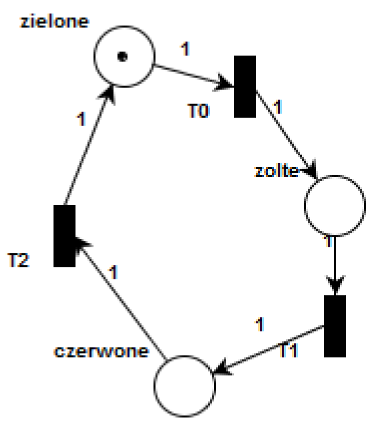
Podstawowe własności sieci Petriego

Adrian Madej 6.01.2024

1. Treść zadań

Zadanie 0

Maszyna stanów. Prosty model maszyny stanów świateł ulicznych przedstawia sieć na rysunku poniżej:



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

Ćwiczenia:

- Narysować przykład w symulatorze.
- Sprawdzić właściwości sieci (ograniczoność, bezpieczeństwo i możliwy deadlock) w symulatorze Pipe w menu "State Space Analysis".
- Wygenerować graf osiągalności "Reachability/Coverability Graph". Zaobserwować:
  - Jakie znakowania są osiągalne ?

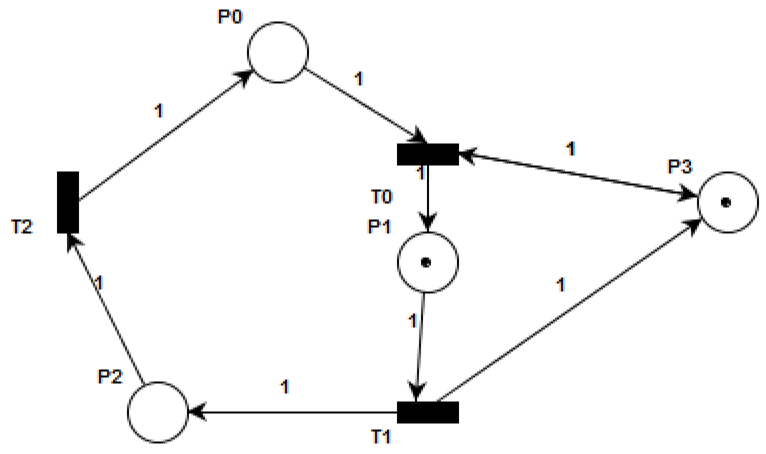
        - Ile wynosi maksymalna liczba znaczników w każdym ze znakowań ? Jakie możemy wyciągnąć z tego wnioski n.t. ograniczoności i bezpieczeństwa?
        - Czy każde przejście jest przedstawione jako krawędź w grafie ? Jaki z tego wniosek n.t. żywotności przejść ?
        - Czy wychodząc od dowolnego węzła grafu (znakowania) można wykonać dowolne przejście ? Jaki z tego wniosek n.t. żywotności sieci? Czy są możliwe zakleszczenia ?
    - Wykonać analizę niezmienników (wybrać 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 możemy 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.

Zadanie 1

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

Zadanie 2

Zasymulować siec jak poniżej.



Dokonać analizy niezmienników przejść. Jaki wniosek można wyciągnąć o odwracalności sieci ? Wygenerować graf osiągalności. Proszę wywnioskować z grafu, czy siec jest żywa. Proszę wywnioskować czy jest ograniczona. Objaśnić wniosek.

Zadanie 3

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

Zadanie 4

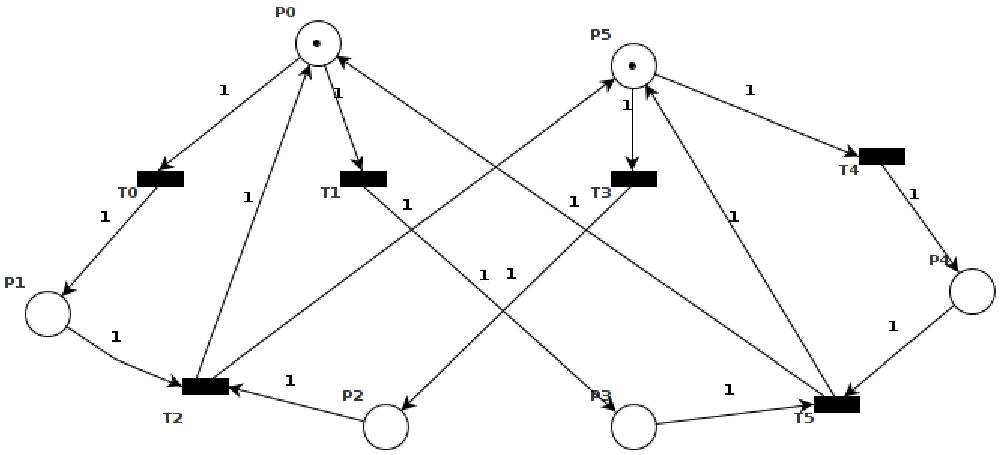
Uruchomić problem producenta i konsumenta z ograniczonym buforem (można posłużyć się przykładem, menu:file, examples). Dokonać analizy niezmienników. Czy siec jest zachowawcza ? Które równanie mówi nam o rozmiarze bufora ?

Zadanie 5

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

Zadanie 6

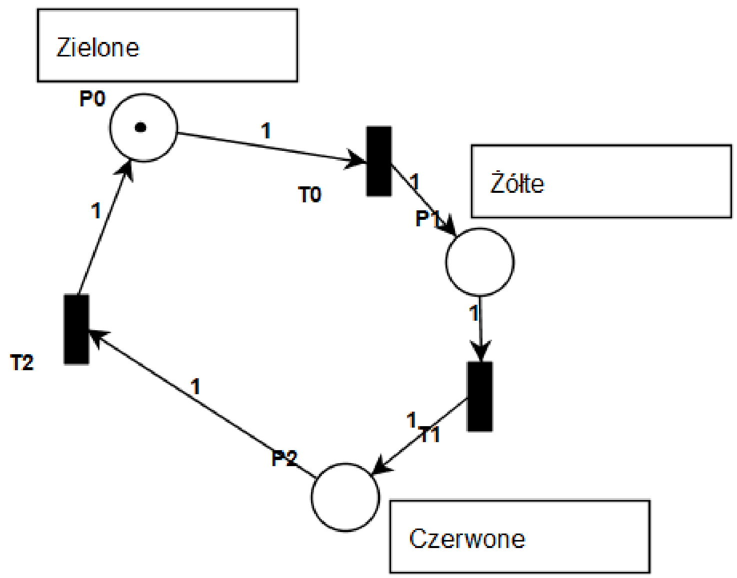
Zasymulować prosty przykład ilustrujący zakleszczenie. Wygenerować graf osiągalności i zaobserwować znakowania, z których nie można wykonać przejść. Zaobserwować właściwości sieci w "State Space Analysis". Poniżej przykład sieci z możliwością zakleszczenia (można wymyślić inny):



2. Rozwiązywanie zadań

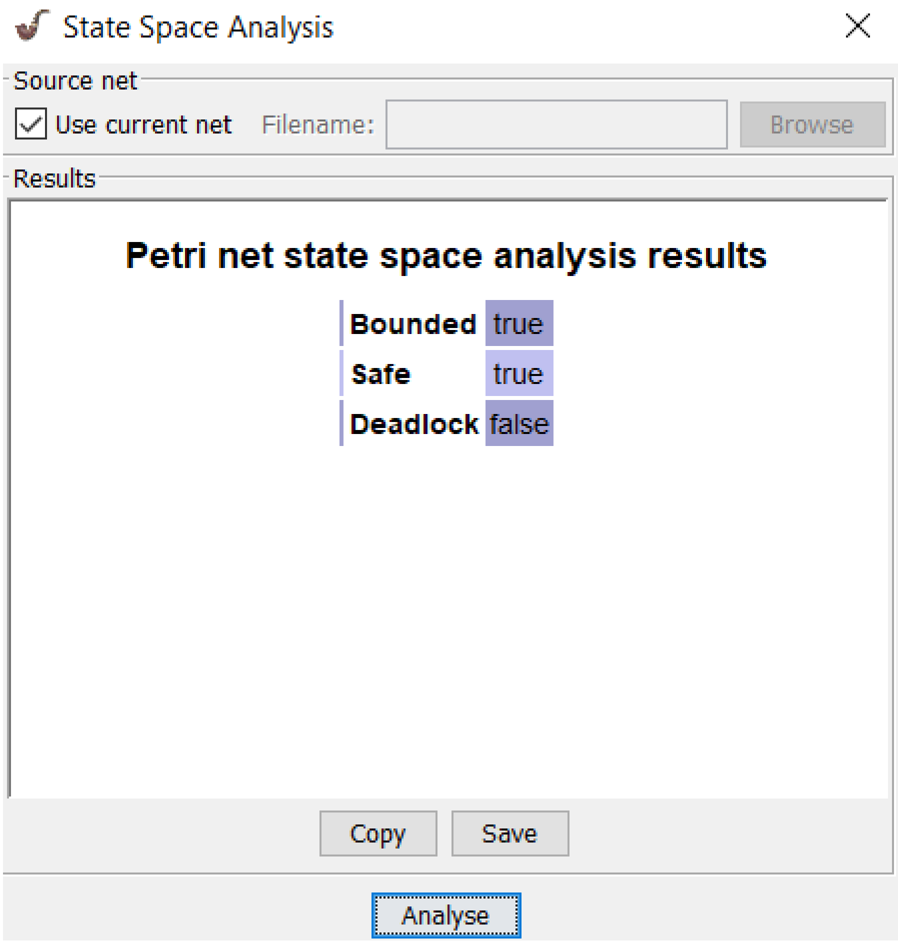
Zadanie 0

Rysujemy przykład w symulatorze



Rys. 1. Graf utworzony w programie PIPE2

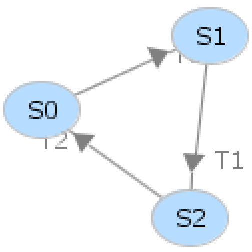
Sprawdzamy właściwości sieci



Rys. 2. Wyniki "State Space Analysis"

Jak widzimy sieć jest ograniczona (gdyż liczba tokenów wewnątrz sieci jest stale równa jeden), bezpieczna (1-ograniczona) i nie występuje w niej deadlock, czyli nie występują sytuacje w których nie moglibyśmy przejść dalej.

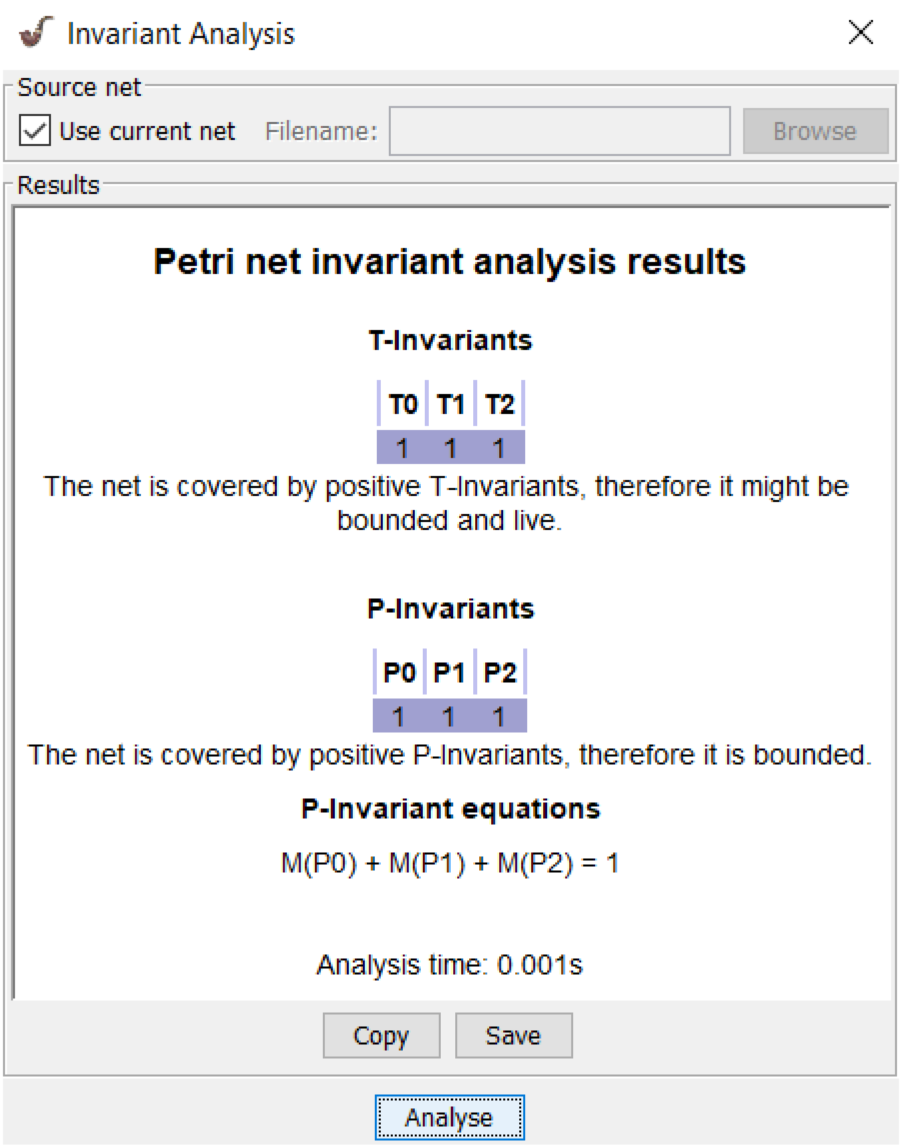
Generujemy graf osiągalności "Reachability/Coverability Graph"



Rys. 3. Graf osiągalności "Reachability/Coverability Graph"

Łatwo zauważyć, że każde znakowanie jest osiągalne, a maksymalna liczba znaczników w każdym z nich wynosi jeden. Na tej podstawie możemy wysnuć wniosek, że sieć jest ograniczona i bezpieczna. Każde z przejść przedstawione jest jako krawędź w grafie, co oznacza, że zawsze można wystartować z dowolnego stanu, zatem każde przejście jest żywe. Wychodząc od dowolnego węzła grafu, można wykonać dowolne przejście. Wynika z tego, że sieć jest żywa a zakleszczenia są niemożliwe.

Dokonujemy analizy niezmienników

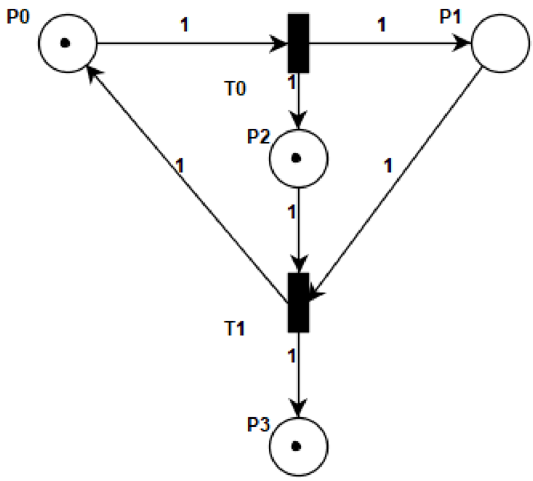


Rys. 4. Wyniki "Invariant Analysis"

Wynik analizy niezmienników przejść mówi, że sieć jest odwracalna - wracając do stanu startowego, należy przejść przez każdą tranzycję. Z kolei wynik analizy niezmienników miejsc określa miejsca, w których łączna suma znaczników się nie zmienia. Pozwala to wnioskować, że sieć jest zachowawcza i ograniczona.

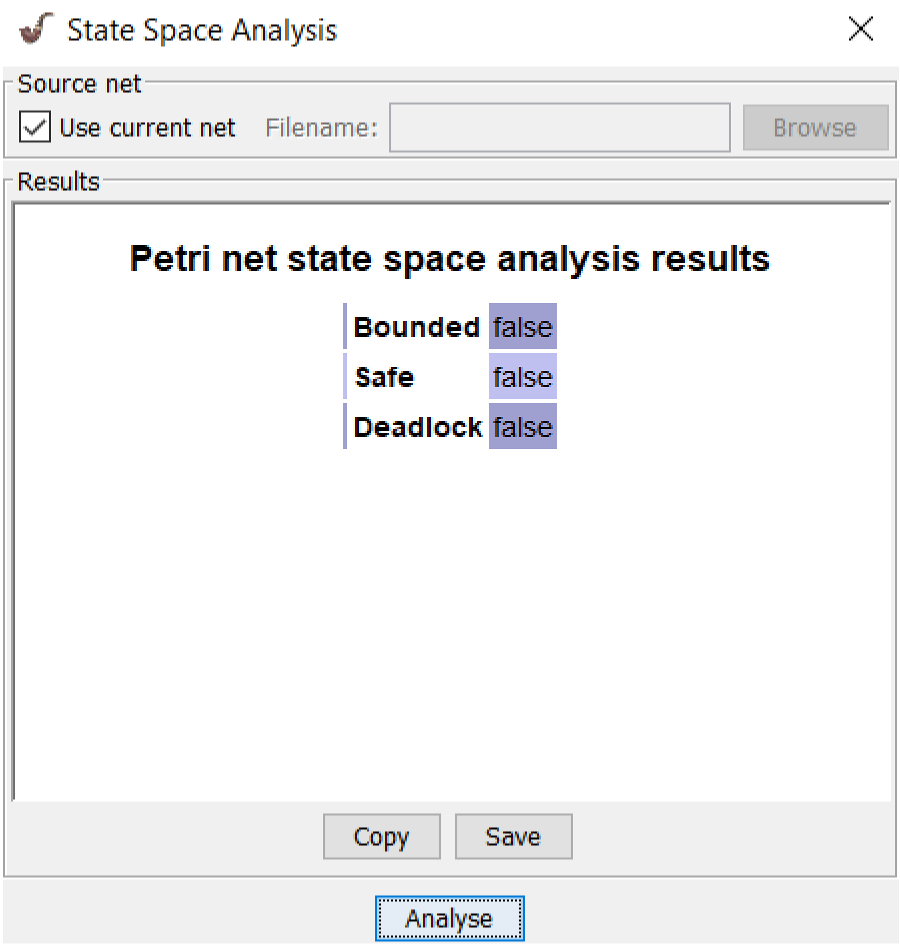
Zadanie 1

Tworzymy maszynę stanów



Rys. 5. Własny graf utworzony w programie PIPE2

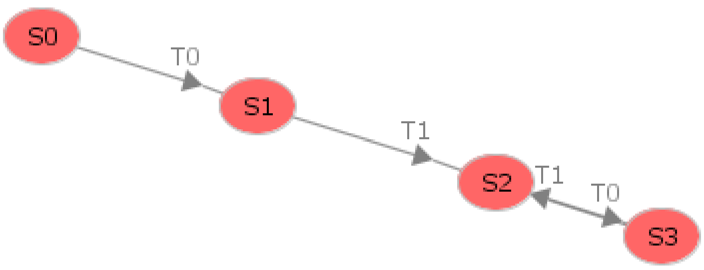
Sprawdzamy właściwości sieci



Rys. 6. Wyniki "State Space Analysis"

Zauważmy, że tranzycja T1 powoduje produkowanie dodatkowych tokenów w P3 zatem sieć nie jest ograniczona. Sieć nie jest też bezpieczna, ponieważ w P3 pojawia się więcej niż jeden token, a zatem nie jest 1-ograniczona. Sieć nie ma możliwości deadlocka, gdyż zawsze mamy możliwość przejść do innego stanu (poza przechodzeniem z P3).

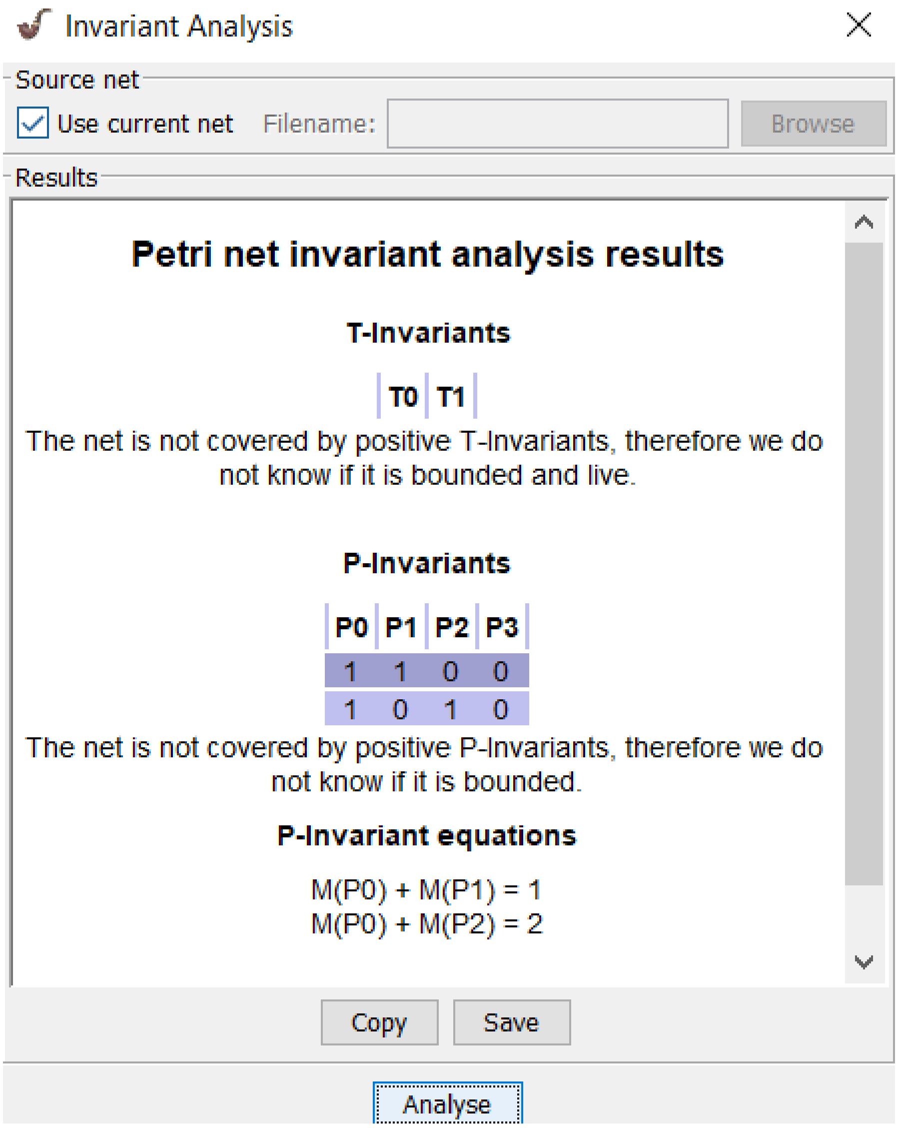
Generujemy graf osiągalności "Reachability/Coverability Graph"



Rys. 7. Graf osiągalności "Reachability/Coverability Graph"

Łatwo zauważyć, że każde znakowanie jest osiągalne a sieć też jest żywa, gdyż każde przejście jest żywe.

Dokonujemy analizy niezmienników

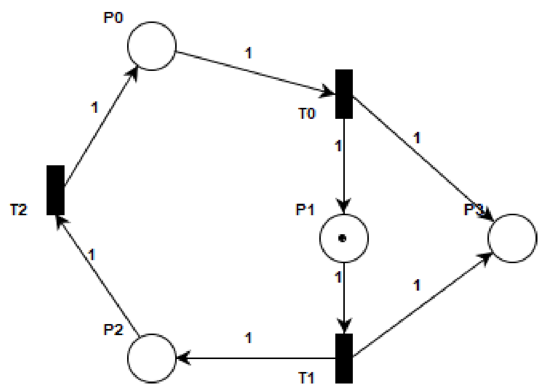


Rys. 8. Wyniki "Invariant Analysis"

Na podstawie powyższych wyników można zauważyć, że sieć jest nieodwracalna, gdyż T-invariants jest puste. Zmieniająca się suma znaczników w równaniach P-invariants equations wskazuje na to, że sieć nie jest ani bezpieczna, ani zachowawcza, ani ograniczona.

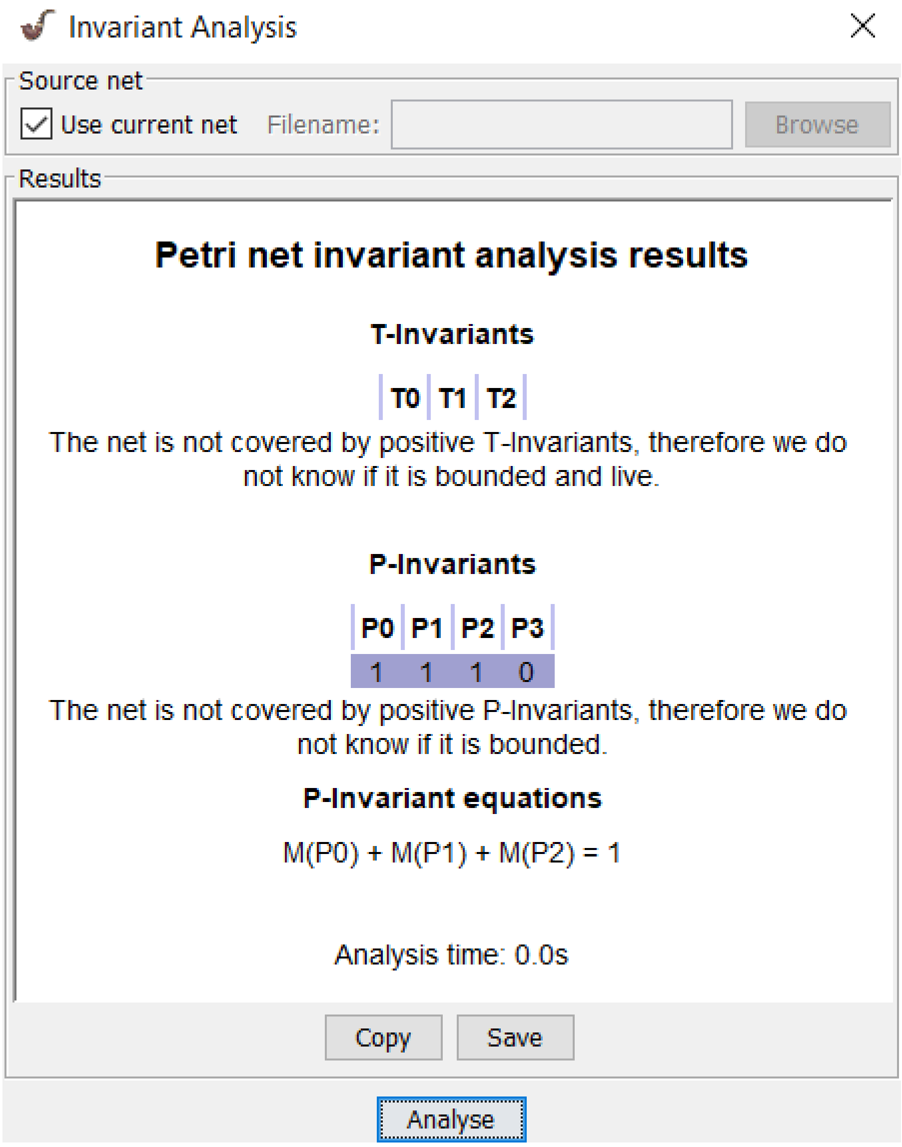
Zadanie 2

Rysujemy podany graf w programie



Rys. 9. Graf utworzony w programie PIPE2

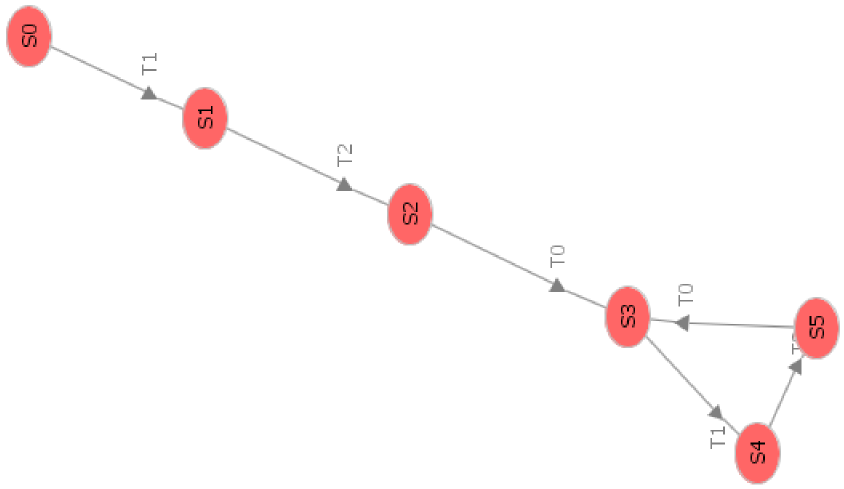
Dokonujemy analizy niezmienników przejść



Rys. 9. Wyniki "Invariant Analysis"

Jak widzimy sieć nie jest odwracalna, gdyż wektor T-invariants nie istnieje

Wygenerujemy graf osiągalności

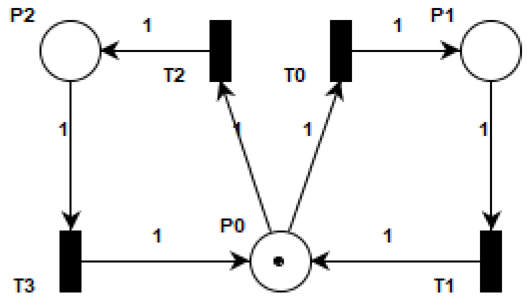


Rys. 10. Graf osiągalności "Reachability/Coverability Graph"

Jak widzimy sieć jest żywa, po wykonaniu każdego przejścia, następuje zapętlenie. Sieć nie jest ograniczona, gdyż ilość tokenów w P3 rośnie w nieskończoność.

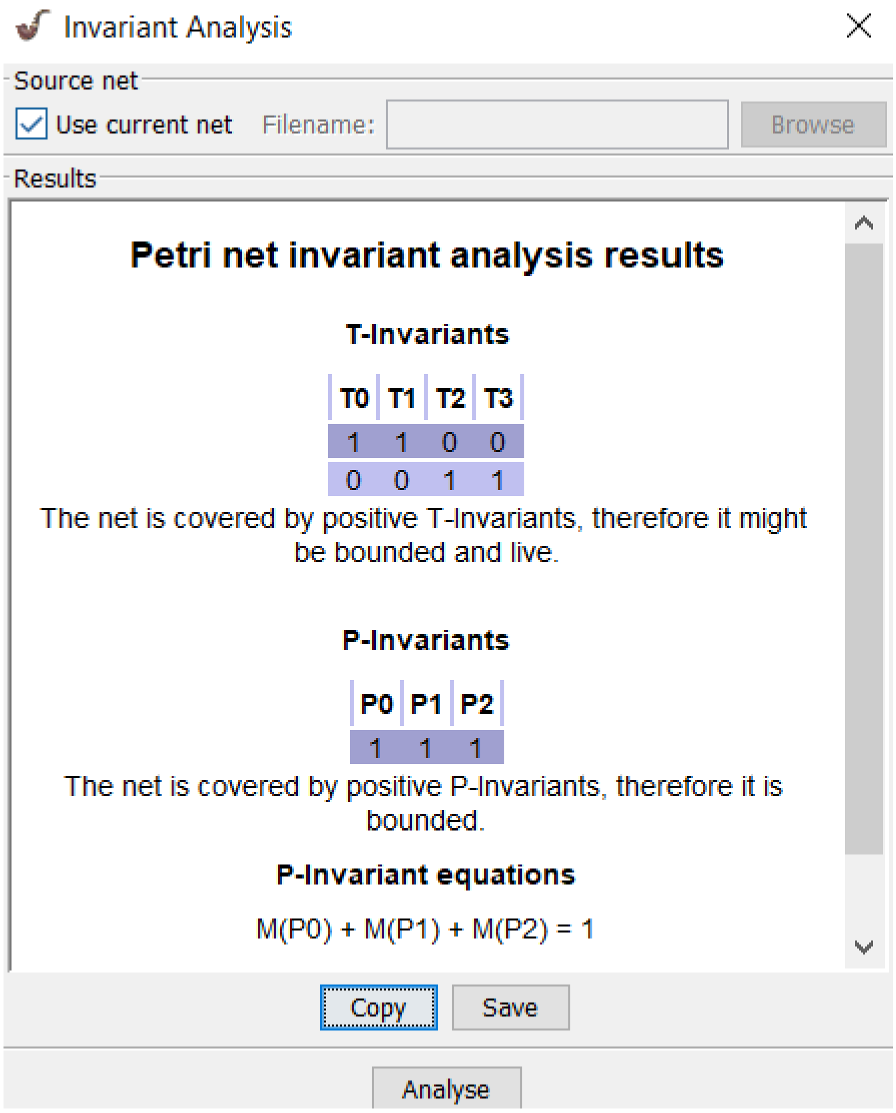
Zadanie 3

Tworzymy graf symulujący wzajemnie wykluczające się dwa procesy na wspólnym zasobie.



Rys. 11. Graf wykluczających się procesów

Dokonujemy analizy niezmienników przejść



Rys. 12. Wyniki "Invariant Analysis"

Gdy przyjrzymy się równaniom P-Invariant, możemy zauważyć, że występują w nim wszystkie trzy stany:

- stan P0 oznacza, że zasób jest wolny

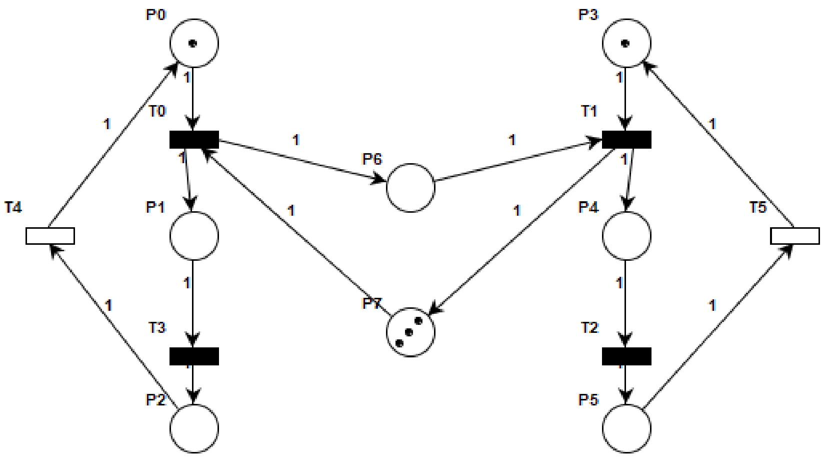
- stan P1 oznacza, że zasób jest zajęty przez pierwszy proces

- stan P2 oznacza, że zasób jest zajęty przez drugi proces.

Działanie sekcji krytycznej pokazuje równanie P-Invariant equations. Suma tokenów we wszystkich stanach zawsze wynosi jeden, czyli token zawsze znajduje się w jednym ze stanów, a na tym polega ochrona sekcji krytycznej.

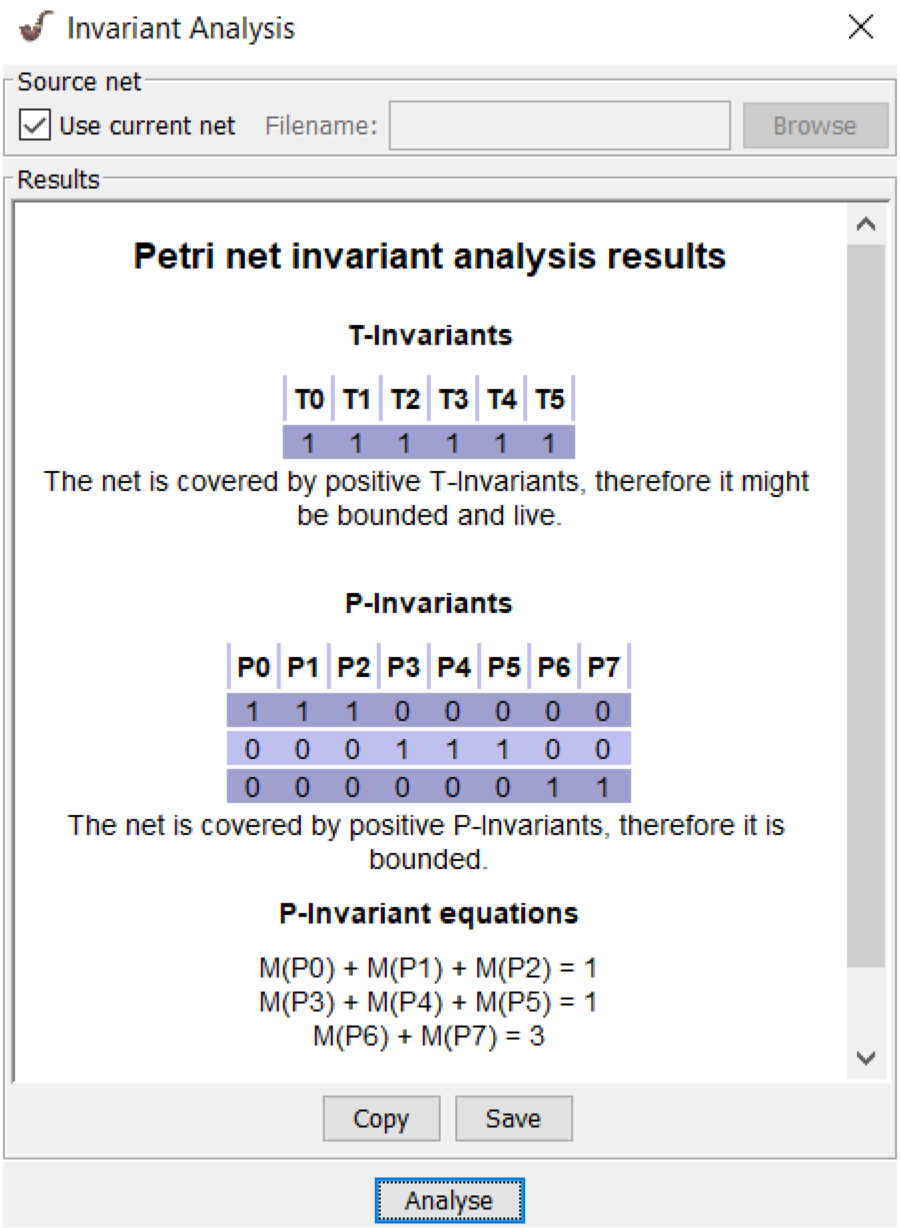
Zadanie 4

Importujemy przykład grafu producenta i konsumenta z ograniczonym buforem



Rys. 13. Graf producenta i konsumenta z ograniczonym buforem

Dokonujemy analizy niezmienników przejść



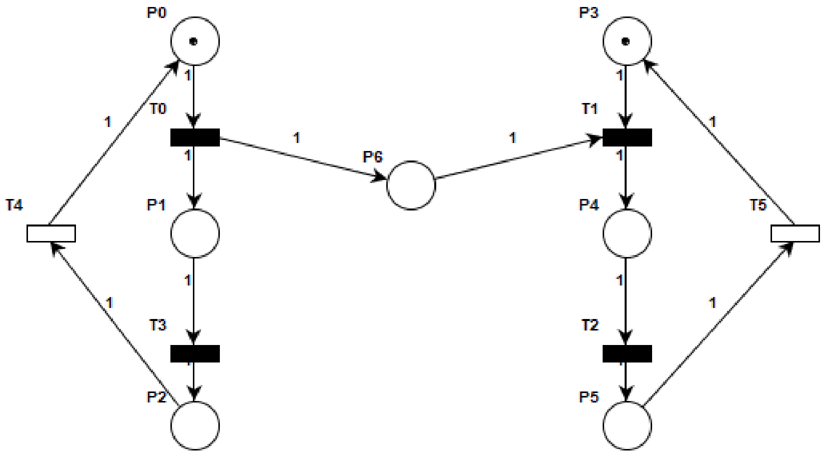
Rys. 14. Wyniki "Invariant Analysis"

Wynik analizy niezmienników przejść mówi, że sieć jest odwracalna. Wszystkie przejścia mogą być wykonane, więc sieć jest żywa. Dodatkowo każda tranzycja produkuje tyle samo tokenów ile pobiera, więc sieć jest zachowawcza.

Trzecie równanie P-Invariant equations mówi nam o wielkości bufora. Miejsca zajęte oznaczone są jako stan P6, a wolne jako P7.

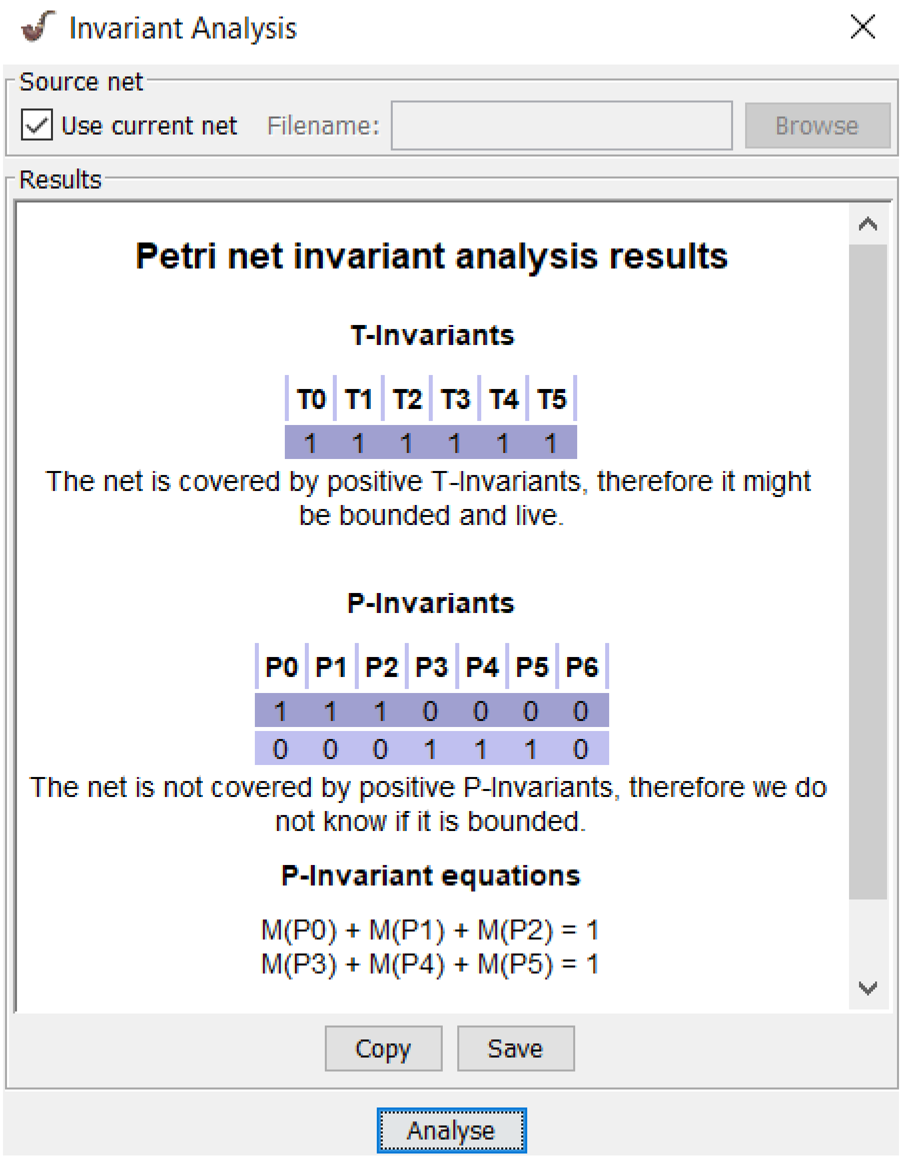
Zadanie 5

Tworzymy symulacje problemu producenta i konsumenta z nieograniczonym buforem, usuwając z poprzedniego grafu P7



Rys. 15. Graf producenta i konsumenta z nieograniczonym buforem

Dokonujemy analizy niezmienników



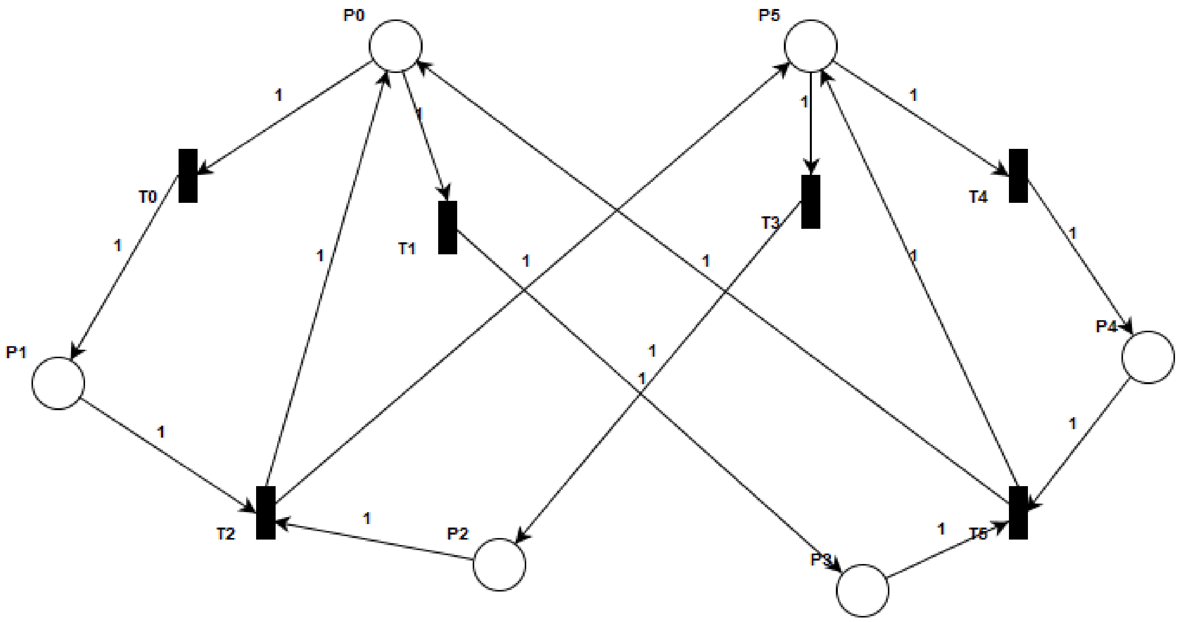
Rys. 16. Wyniki "Invariant Analysis"

Podobnie jak poprzednio sieć jest odwracalna, co wynika z T-invariants.

W P-Invariants widzimy, że przy stanie P6 (buforze) zawsze jest 0, a żadne z równań P-Invariant equations nie uwzględnia P6. Nie ma się co dziwić, w końcu jest on nieograniczony, zatem nie ma równania które go ogranicza w jakiś sposób, czyli występuje brak pełnego pokrycia miejsc.

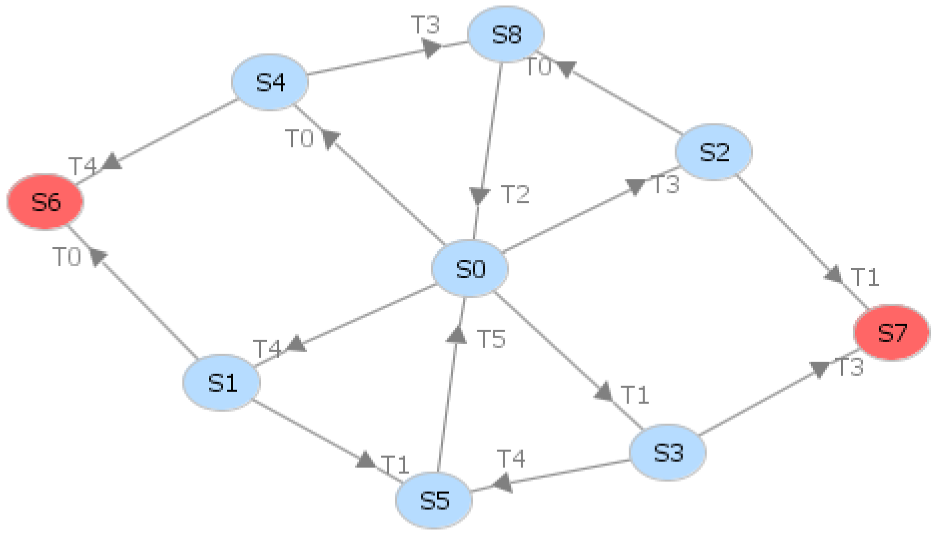
Zadanie 6

Generujemy graf z zakleszczeniem



Rys. 17. Graf z zakleszczeniem

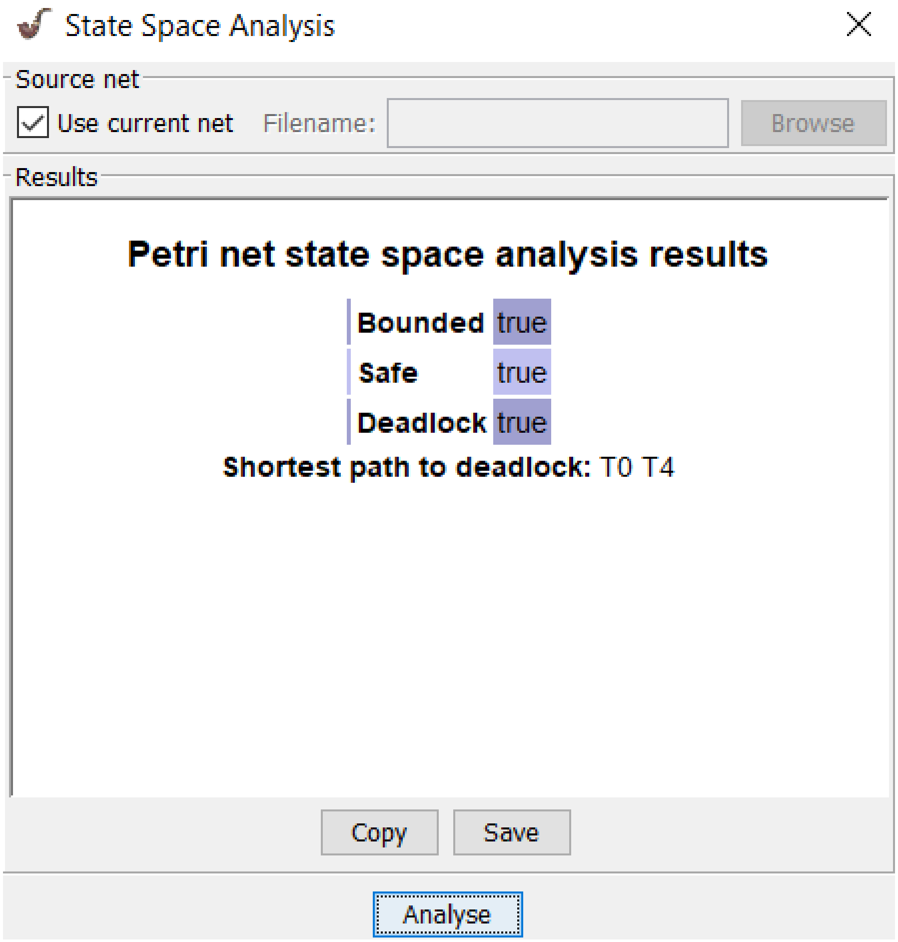
Generujemy graf osiągalności "Reachability/Coverability Graph"



Rys. 18. Graf osiągalności "Reachability/Coverability Graph"

Na podstawie powyższego grafu możemy jasno stwierdzić, że stanami, w których nastąpi deadlock są stany: S6, S7. Jak widzimy, po wejściu do nich, nie jesteśmy już w stanie z nich wyjść

Przeanalizujmy właściwości sieci w "State Space Analysis"



Rys. 17. Wyniki "State Space Analysis"

Jak widzimy sieć jest ograniczona, bezpieczna, oraz występuje w niej zakleszczenie, do którego najkrótsza ścieżka to T0 T4.

3. Wnioski

- Sieci Petriego to matematyczny formalizm używany do modelowania i analizy systemów dynamicznych, takich jak procesy produkcyjne, systemy komunikacyjne, czy procesy decyzyjne.
- Sieć Petriego może znajdować się w stanie równowagi, co oznacza, że żadne przejścia nie są aktywne, a żadne miejsca nie zmieniają swojego stanu.

- Cykl ożywienia to sytuacja, w której istnieje taka sekwencja przejść, że po jej wykonaniu system wraca do pierwotnego stanu.
- Sieć Petriego jest bezpieczna, gdy liczba żetonów we wszystkich miejscach jest ograniczona do jednego, a bezpieczeństwo gwarantuje brak konfliktów o dostęp do zasobów.
- Sieć Petriego jest odwracalna, gdy istnieje przekształcenie, które pozwala na powrót do stanu poprzedniego.
- Sieć Petriego jest bezpieczna strukturalnie, gdy dla każdego miejsca istnieje co najmniej jedno przejście wejściowe i jedno przejście wyjściowe.
- Symulator PIPE2 jest łatwym do obsługi narzędziem pozwalającym badać właściwości sieci Petriego

4. Bibliografia

(Każdy podpunkt stanowi hiperłącze)

- Strona laboratorium
- Sieci Petriego – Politechnika Warszawska
- Sieci Petriego – Politechnika Poznańska
- Wikipedia – Sieci Petriego