Sprawozdanie nr 9

Data ćwiczeń: 28.05.2015

Data oddania sprawozdania: 18.06.2015

* Tematyka

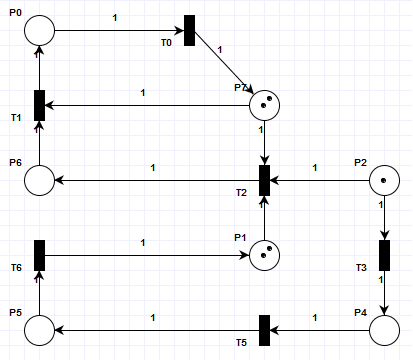
Na dziewiątych zajęciach laboratoryjnych za pomocą open-source'owego programu PIPE mieliśmy możliwość zasymulowania i zbadania pracy programów przedstawionych za pomocą sieci Petriego.

* Zadania

Na zajęciach należało zasymulować kilka sieci zgodnych z treścią zadania, w programie PIPE. Zrzut ekranu każdej z sieci oraz ich analiza znajduje się na następnych stronach.

* + Zadanie 1

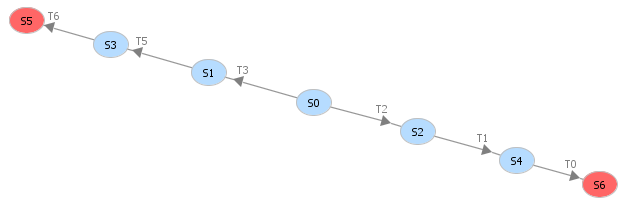
*Wymyślić własną maszynę stanów, zasymulować przykład i dokonać analizy grafu osiągalności oraz niezmienników.*



Właściwości sieci:

|  |  |
| --- | --- |
| Bounded | true |
| Safe | false |
| Deadlock | true |

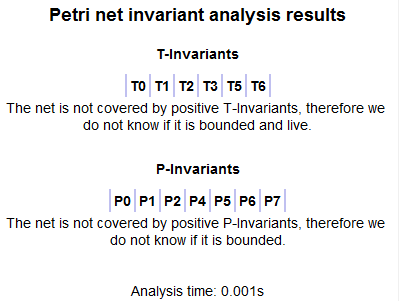
Graf osiągalności:



Komentarz: Z analizy grafu osiągalności widzimy, że każde z oznakowań jest osiągalne. Na tej podstawie możemy wysnuć wniosek, iż jest to sieć ograniczona, ale nie jest siecią bezpieczną. Mimo lekkiej pomyłce w oznaczeniu przejść (brakuje T4), widać że każde z dostępnych przejść w tej sieci jest przejściem żywym. Jesteśmy również w stanie wykonać dowolne przejście. Niestety są w tej sytuacji możliwe zakleszczenia, a jedną ze ścieżek aby się zakleszczył taki program jest:

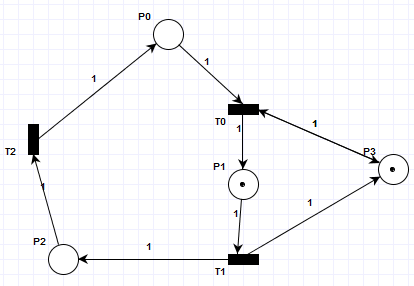
T2 →T1 →T0.

Analiza niezmienników:



* Zadanie 2

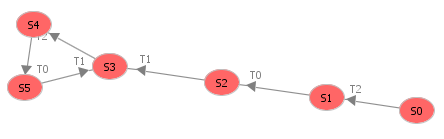
*Zasymulować sieć jak poniżej:*



Właściwości sieci:

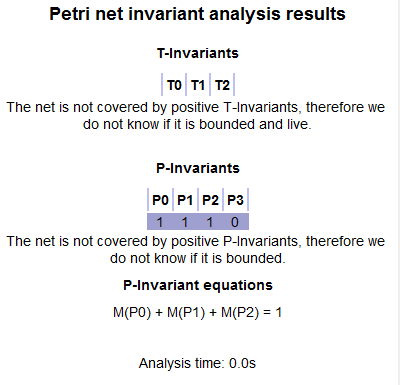
|  |  |
| --- | --- |
| Bounded | false |
| Safe | false |
| Deadlock | false |

Graf osiągalności:



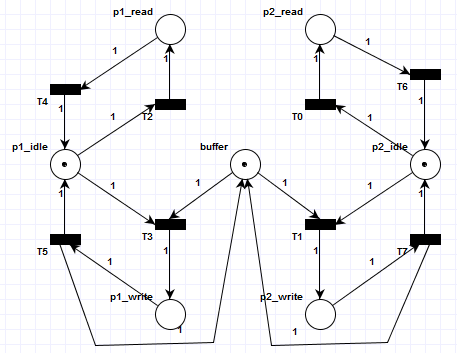
Komentarz: Z analizy grafu osiągalności widzimy, że każde z oznakowań jest osiągalne. Nie jest to sieć ograniczona, ani żywa. Widać że każde z dostępnych przejść w tej sieci jest przejściem żywym. Jesteśmy również w stanie wykonać dowolne przejście i nie jest możliwe zakleszczenie

Analiza niezmienników:

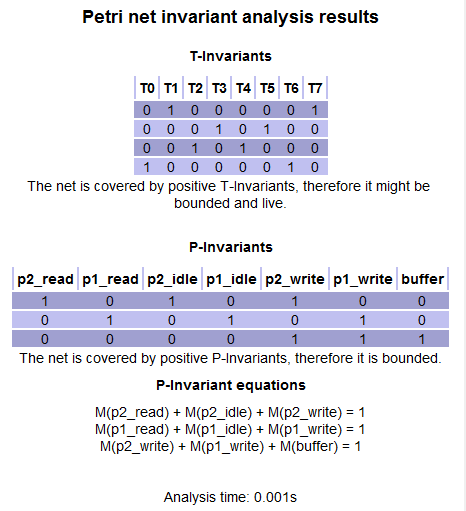


* Zadanie 3

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



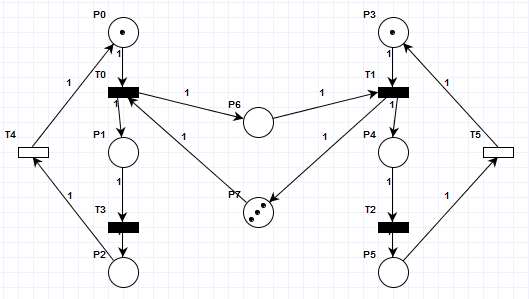
Korzystając z zasobów dostępnych na stronie belgijskiego uniwersytetu Universite Libre de Bruxelles dokonałem próby skonstrukowania sieci Petriego odpowiadającej sytuacji, w której dwa procesy wzajemnie się wykluczają. Wzajemne wykluczenie jest zapewnione przez obecność tylko jednego tokena w bufferze. Wszystkie sytuacje są połączone, a przejście "pół-żywe" (ang. *semi-live*). Po włączeniu analizy niezmienników uzyskałem następujący wynik:



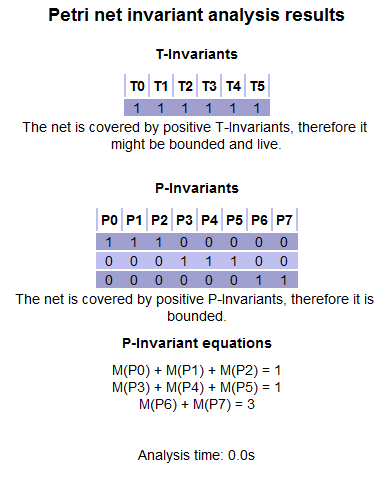
Liczba sumaryczna tokenów w sytuacjach kiedy proces 1 lub 2 jest procesem czytającym, piszącym, bądź oczekującym, jest stała. To dostarcza nam informacji, że w programie odpowiadającym takiej sieci Petriego proces może być tylko w jednym z trzech stanów. Równanie, które pokazuje ochronę sekcji krytycznej to równanie numer trzy. Widzimy na nim, że tylko jeden z obu procesów może mieć dostęp do bufora, czyli być w stanie krytycznym.

* Zadanie 4

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



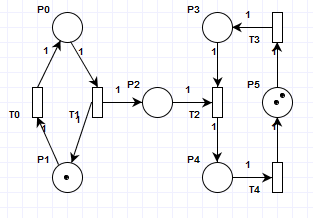
W tym zadaniu skorzystałem z sieci dostępnych w przykładach do programu PIPE. Analiza niezmienników:



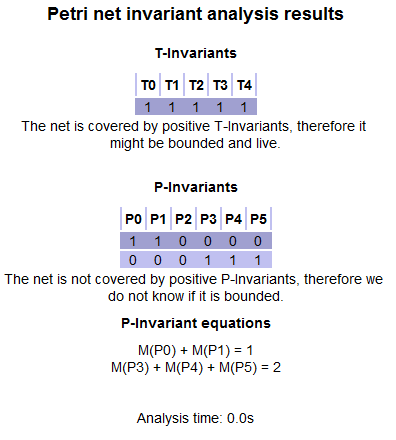
Podobnie jak w poprzednim przykładzie widać jakie znaczenie mają równania *P- invariant equations*. Równanie numer trzy pokazuje nam rozmiar bufora.

* Zadanie 5

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



Analiza niezmienników:



Bardzo łatwo na podstawie analizy niezmienników zauważyć sytuację opisaną w treści zadania. Nie wszystkie miejsca są w tej sieci Petriego pokryte, co uniemożliwia jednoznaczą analizę tej sieci.

* 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 Petriego ilustrujący zakleszczenie udało mi się już zaprezentować przy okazji zadania 1, w którym zamieściłem odpowiednie zrzuty ekranu z programu PIPE oraz komentarze do nich.