

# Teoria Współbieżności

## Ćwiczenie 11

### 1 Cel ćwiczenia

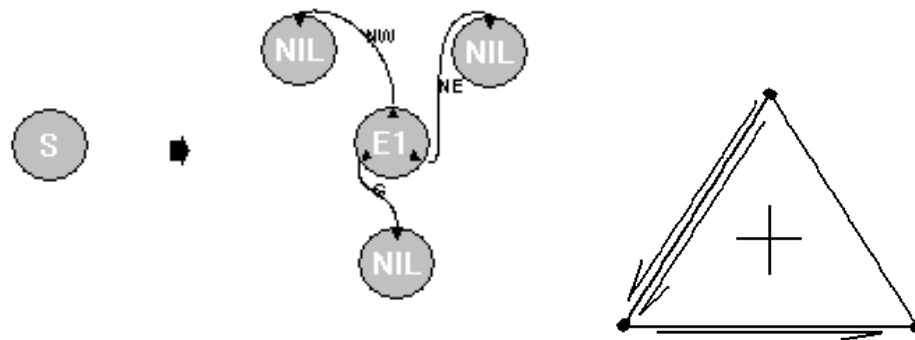
Zapoznanie z zastosowaniem teorii sieci Petrii na przykładzie modelowania algorytmu generacji siatek elementów trójkątnych. Zapoznanie ze środowiskiem programu pipe2. Zapoznanie z zastosowaniem gramatyki grafowej do modelowania wątków.

W szczególności studenci zapoznani zostaną z siecią Petrii implementowaną w programie pipe2, służącą do modelowania algorytmu generacji siatek trójkątnych, oraz z wykorzystaniem możliwości programu pipe2 do analizy własności sieci poprzez generację grafu osiągalności i badanie parametrów sieci Petrii.

### 2 Plan ćwiczenia

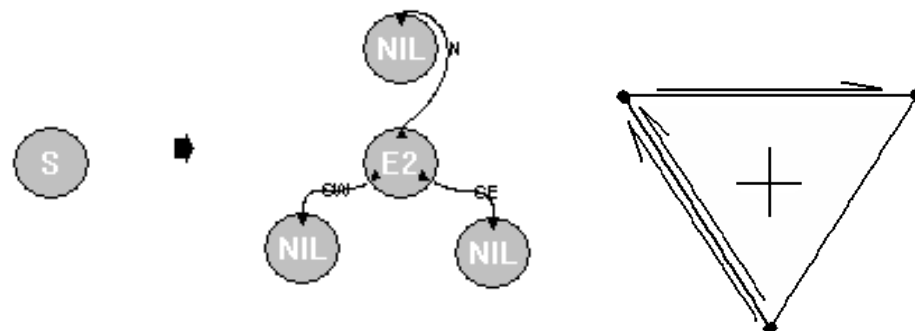
Na początku ćwiczeń przedstawiona zostanie gramatyka grafowa służąca do generacji dwuwymiarowych siatek trójkątnych. W tym celu użyta zostanie prezentacja z Załącznika 7A. Dane są następujące produkcje w gramatyce grafowej:

Produkcja generująca element trójkątny pierwszego rodzaju



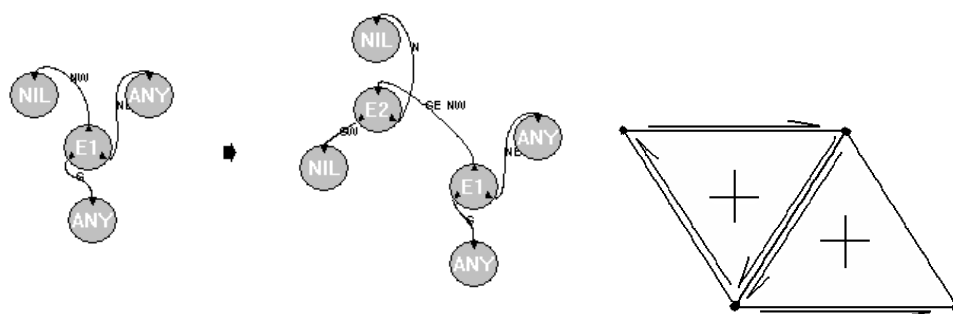
Rysunek 1: (Ptriangle1)

Produkcja generująca element trójkątny drugiego rodzaju



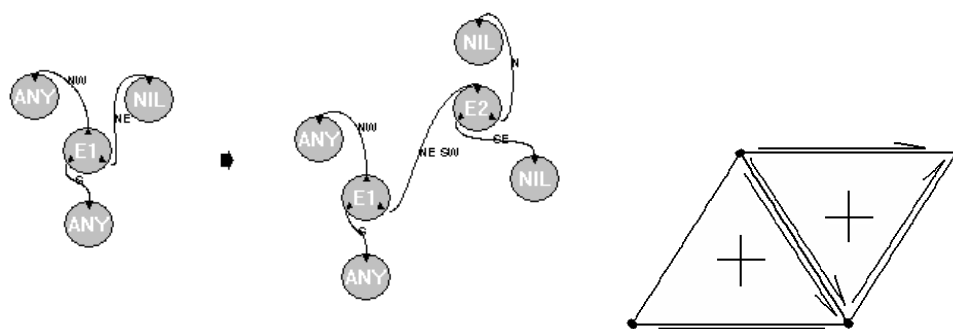
Rysunek 2: (Ptriangle2)

Produkcja generująca sąsiada elementu trójkątny pierwszego rodzaju w kierunku „w lewo”



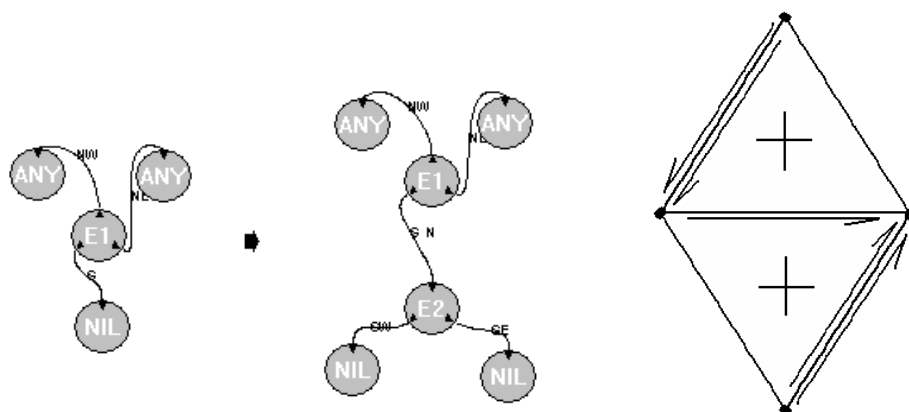
Rysunek 3: (PT1Left)

Produkcja generująca sąsiada elementu trójkątny pierwszego rodzaju w kierunku „w prawo”



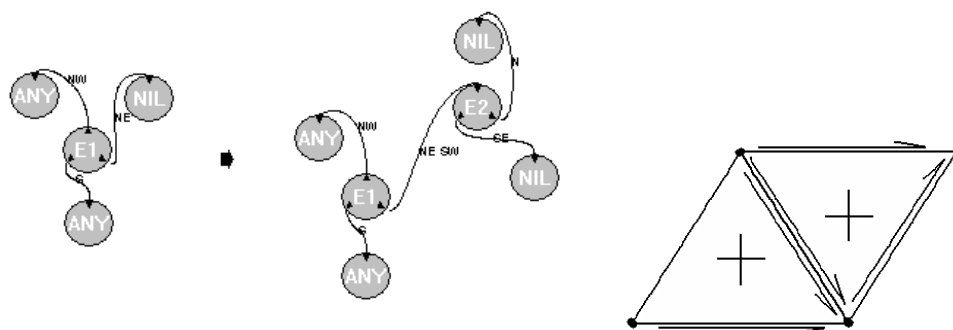
Rysunek 4: (PT1Right)

Produkcja generujące sąsiada elementu trójkątny pierwszego rodzaju w kierunku „na dół”



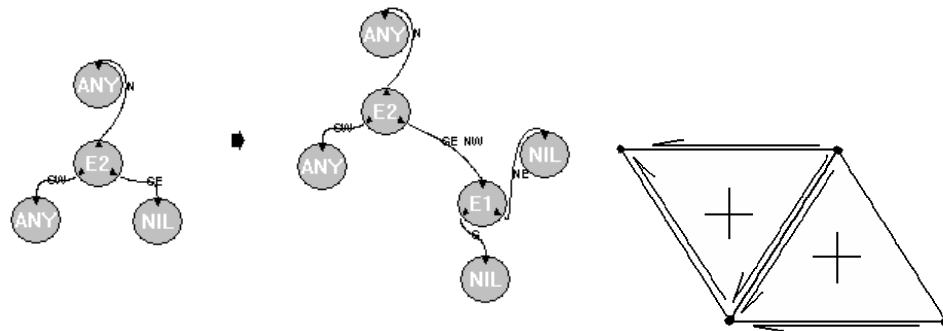
Rysunek 5: (PT1Down)

Produkcja generujące sąsiada elementu trójkątny drugiego rodzaju w kierunku „w lewo”



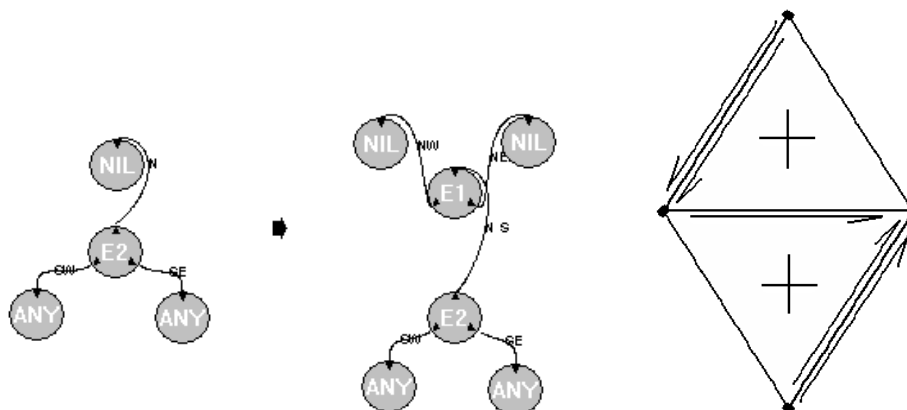
Rysunek 6: (PT2Left)

Produkcja generująca sąsiada elementu trójkątny drugiego rodzaju w kierunku „w prawo”



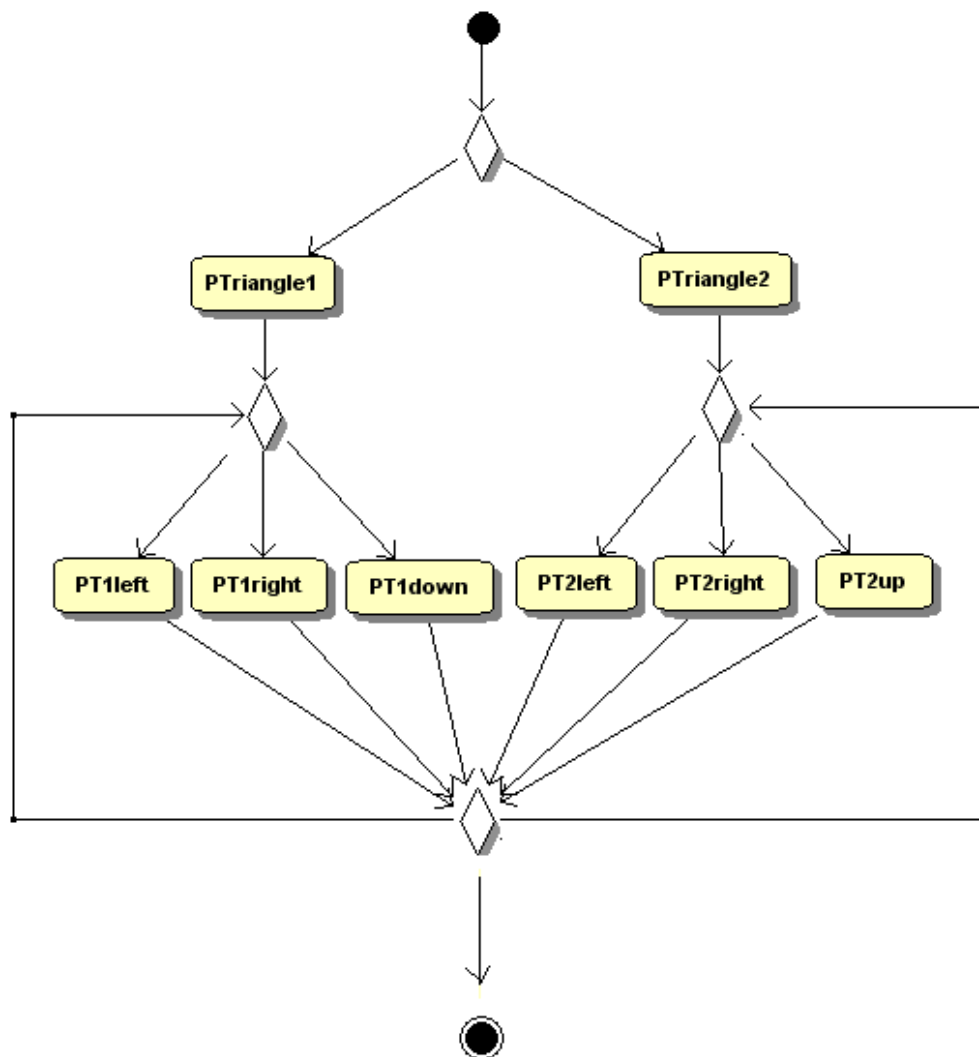
Rysunek 7: (PT2Right)

Produkcja generująca sąsiada elementu trójkątny drugiego rodzaju w kierunku „do góry”

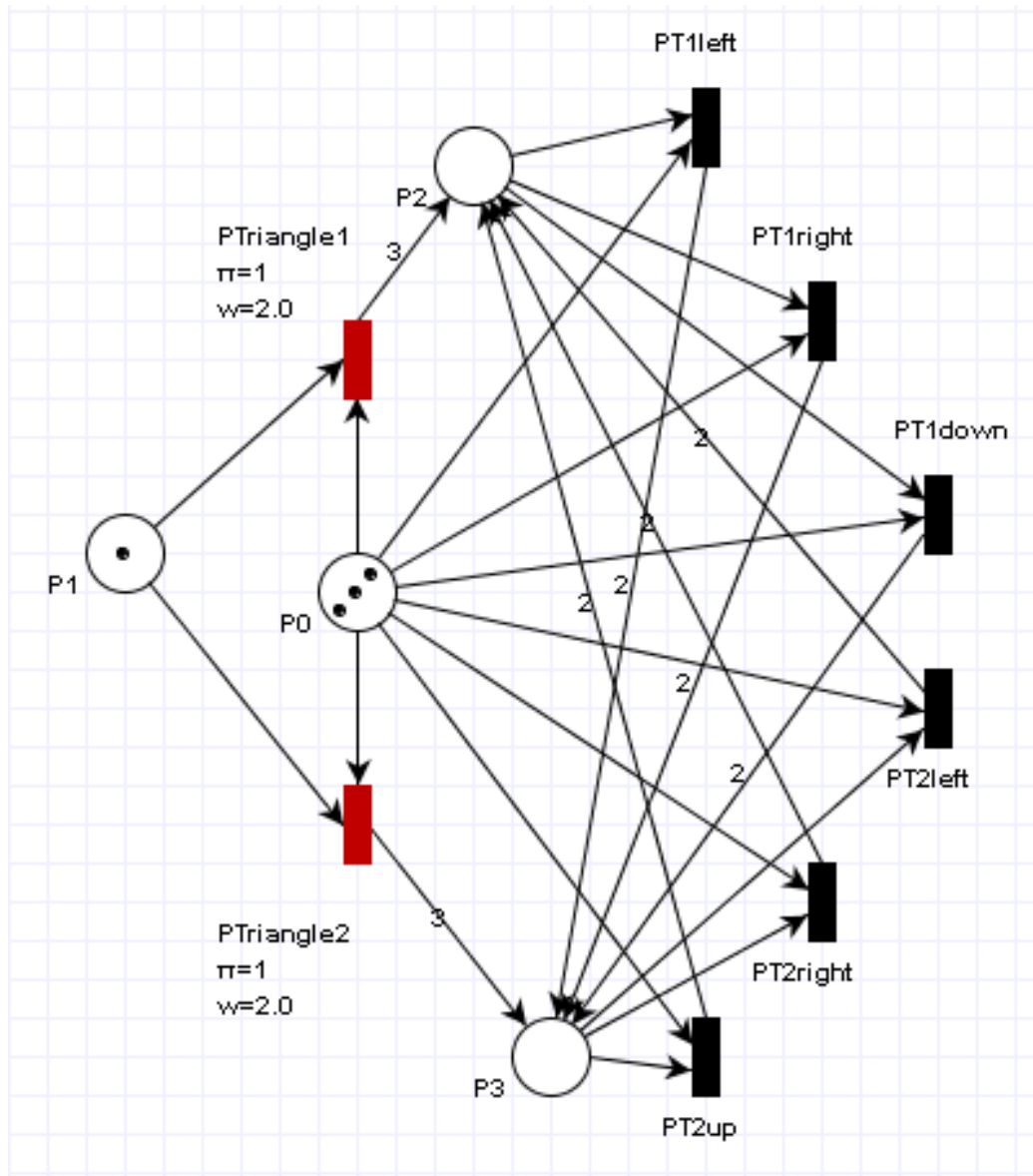


Rysunek 8: (PT2Up)

Dany jest diagram sterujący, określający kolejność wykonywania produkcji



Dana jest wreszcie sieć Petri modelująca proces generacji siatki



Ilość tokenów w stanie  $P_0$  oznacza ilość elementów które chcemy wygenerować. Jeden token w stanie  $P_1$  oznacza możliwość wyboru pomiędzy rodzajem pierwszego elementu. Sposób sterowania procesem generacji siatki przedstawiony jest w prezentacji w Załączniku 7A.

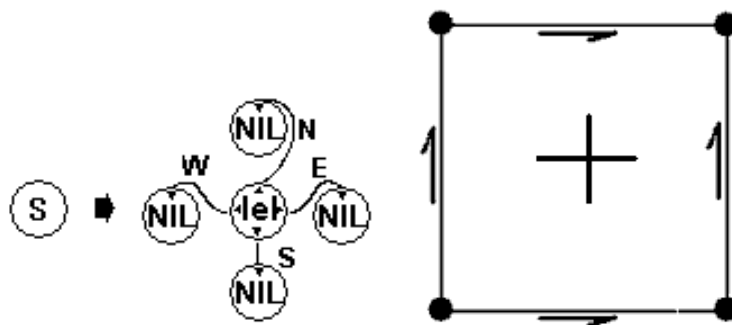
## 2.1 Ćwiczenia

Zadanie polega na wykonaniu następujących etapów:

1. Proszę narysować podaną sieć w programie pipe2
2. Proszę dla różnej ilości generowanych elementów  $N = 1, 2, 3, \dots, 10$  wygenerować graf osiągalności oraz zbadać własności sieci takie jak bezpieczeństwo, żywotność, deadlock. Proszę podać interpretację tych własności na przykładzie zagadnienia generacji siatki
3. Proszę zmodyfikować diagram sterujący i sieć Petri, w taki sposób aby sieć generowała tylko poziomy ciąg elementów.
4. Proszę ponownie dla różnej ilości generowanych elementów  $N = 1, 2, 3, \dots, 10$  wygenerować graf osiągalności oraz zbadać własności sieci takie jak bezpieczeństwo, żywotność, deadlock. Proszę podać interpretację tych własności na przykładzie zagadnienia generacji siatki
5. Proszę zmodyfikować diagram sterujący oraz sieć w taki sposób aby nowo generowane elementy były ze sobą łączone

### 3 Wprowadzenie teoretyczne

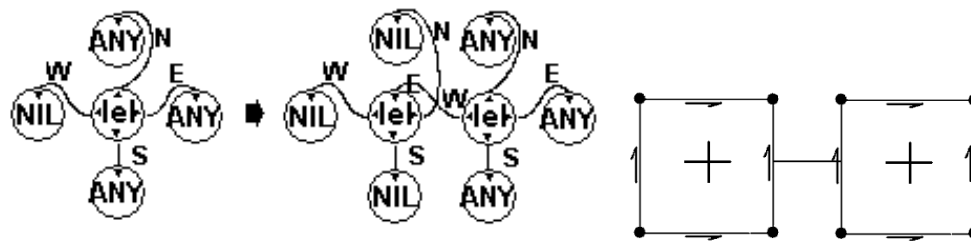
Dane są następujące produkcje w gramatyce grafowej: Produkcja generująca element prostokątny



Rysunek 9: (PInit)

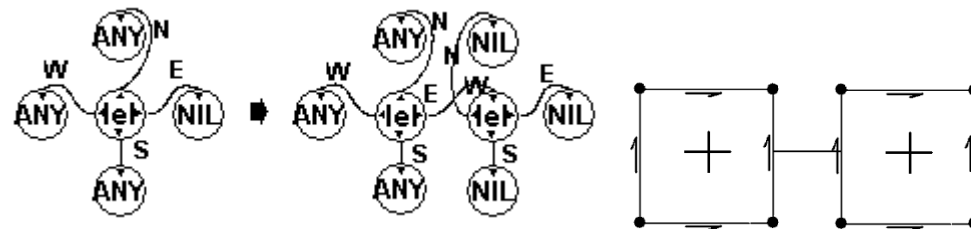


Produkcja generująca sąsiada elementu w kierunku „w lewo”



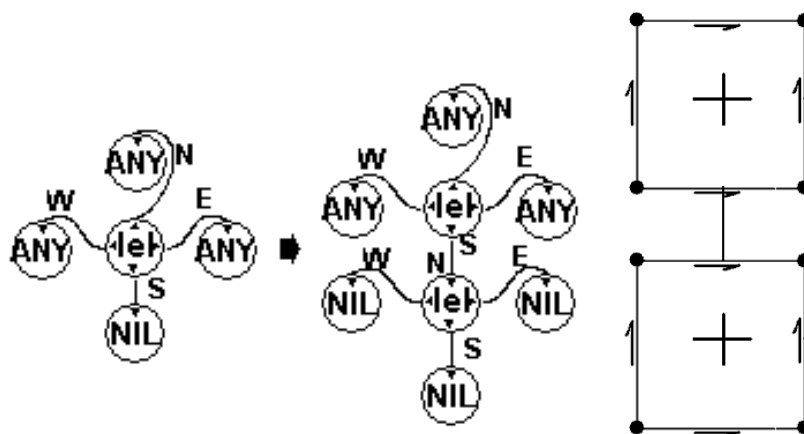
Rysunek 10: (PLeft)

Produkcja generująca sąsiada elementu w kierunku „w prawo”



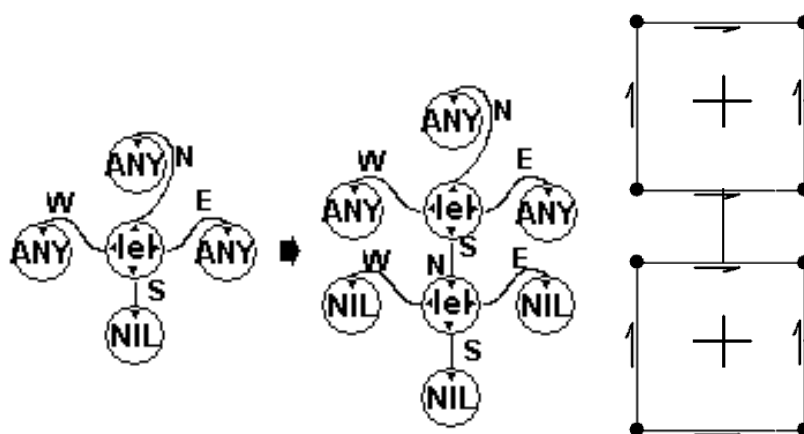
Rysunek 11: (PRight)

Produkcja generująca sąsiada elementu w kierunku „do góry”



Rysunek 12: (PUp)

Produkcja generująca sąsiada elementu w kierunku „na dół”



Rysunek 13: (PDown)

## 4 Ćwiczenie 2

Zadanie polega na wykonaniu następujących etapów:

1. Proszę narysować diagram sterujący procesem generacji siatki.

2. Proszę zaprojektować sieć Petri modelującą algorytm generacji siatki w programie pipe2. Proszę założyć że ilość tokenów w stanie  $P0$  oznacza ilość elementów które chcemy wygenerować. Proszę założyć że zewnętrzny moduł sterujący nie dopuszcza do nachodzenia elementów siatki na siebie.
3. Proszę uruchomić animację wykonania sieci.
4. Proszę dla różnej ilości generowanych elementów  $N = 1, 2, 3, \dots, 10$  wygenerować graf osiągalności oraz zbadać własności sieci takie jak bezpieczeństwo, żywotność, deadlock. Jak wygląda analiza niezmienników przejść? Czy sieć jest odwracalna? Proszę podać interpretację tych własności na przykładzie zagadnienia generacji siatki.

## Literatura

- [1] Wykład z przedmiotu „Teoria współbieżności” – rozdział dotyczący sieci Petrii
- [2] Marcin Szpyrka “Sieci Petri”
- [3] Szymczak A., Paszyński M., Graph grammar-based Petri nets model of concurrency for the self-adaptive hp-Finite Element Method with rectangular elements, Lecture Notes in Computer Science 5544 (2009)