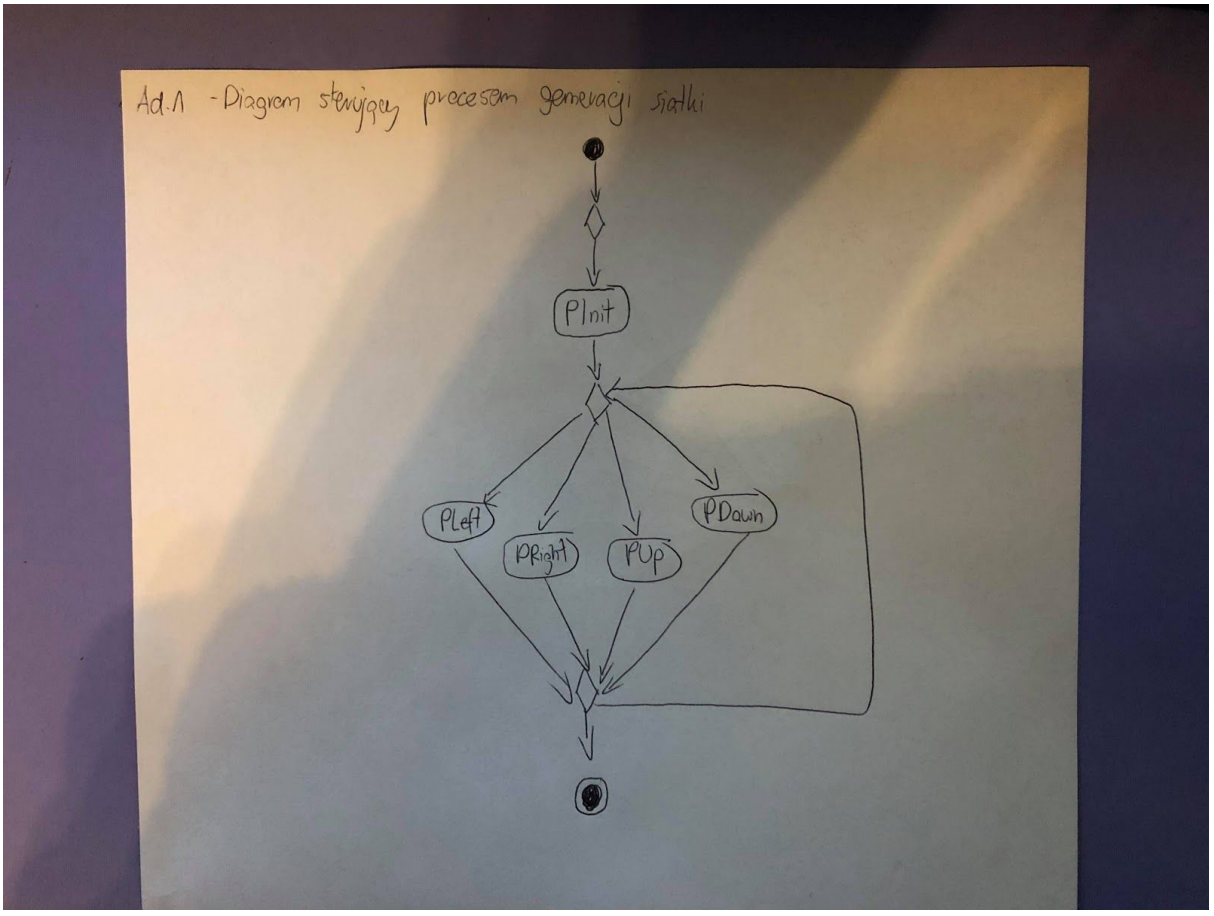
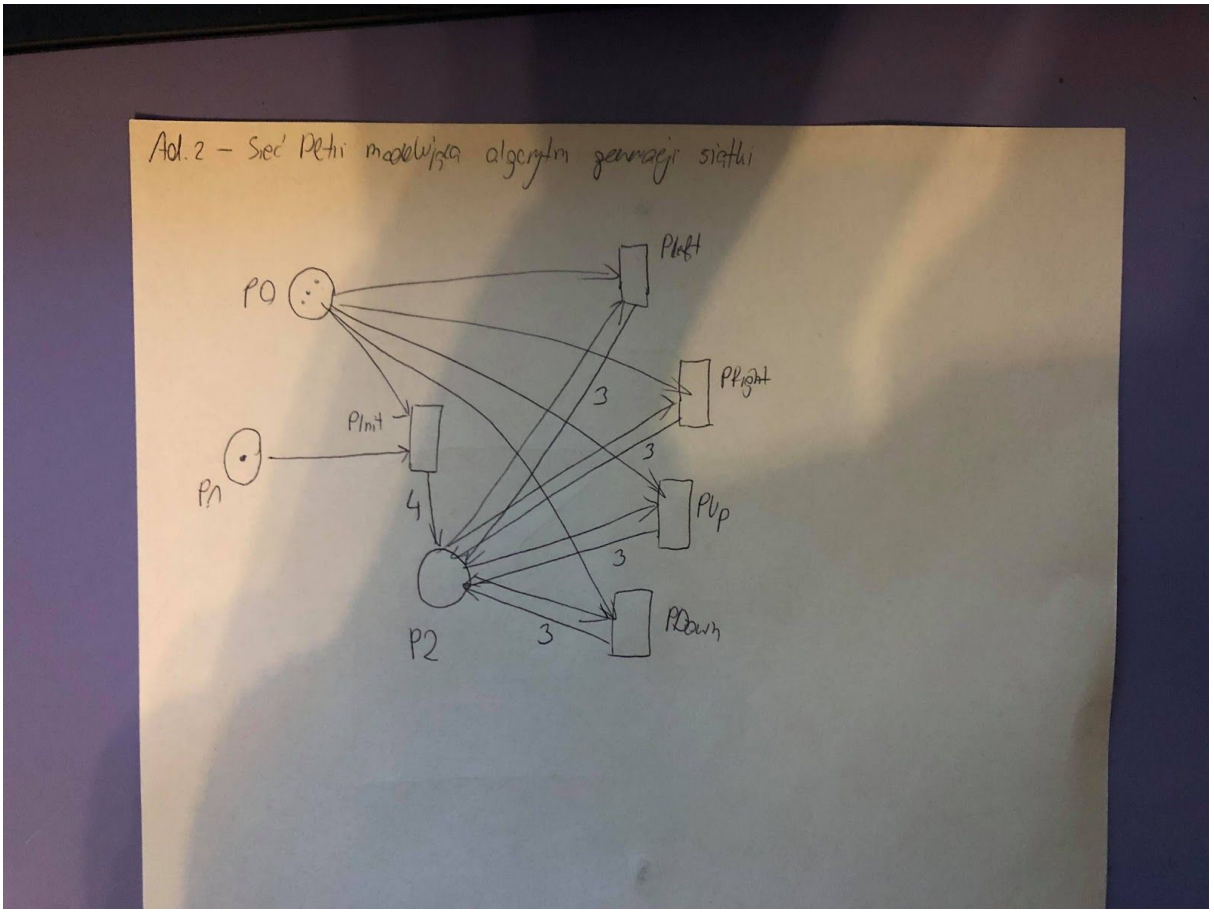
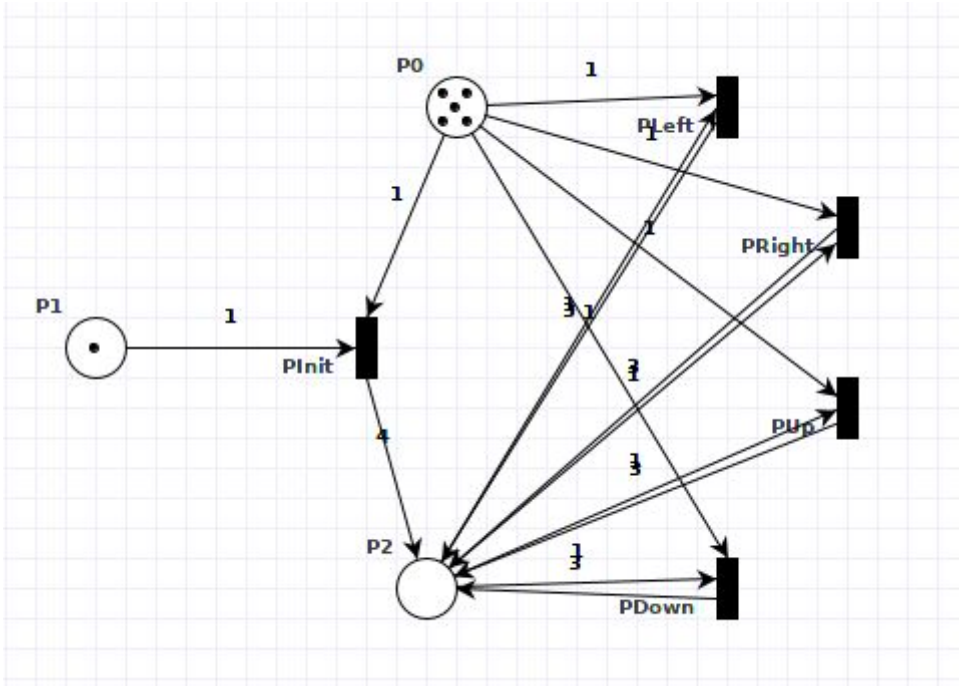
Sprawozdanie z zadania domowego z laboratorium 7. - Dawid Witek 14:40

Zadanie 1. - Diagram sterujący procesem generacji siatki



Zadanie 2. - Sieć Petri modelująca algorytm generacji siatki w programie pipe2.



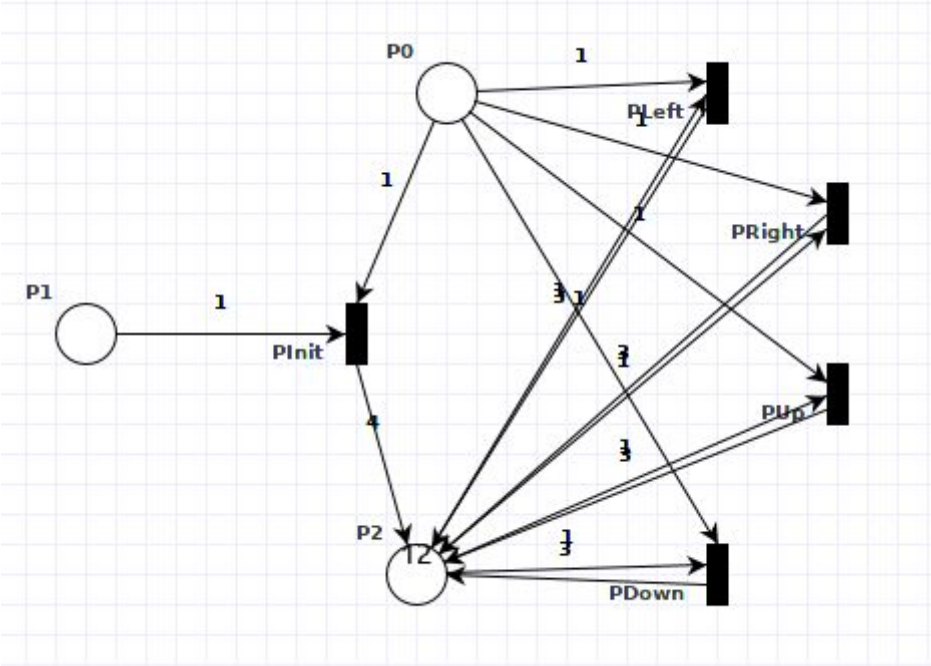


Zadanie 3. - Animacja wykonania sieci

Stan początkowy jest identyczny z tym zaprezentowanym wyżej. Historia wykonania animacji:

Animation history
Initial Marking
PInit
PRight
PDown
PUp
PUp

Stan końcowy sieci:



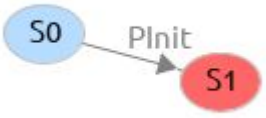
Graf osiągalności i własności sieci dla:

- 1 elementu:

Petri net state space analysis results

Bounded	true
Safe	false
Deadlock	true

Shortest path to deadlock: PInit

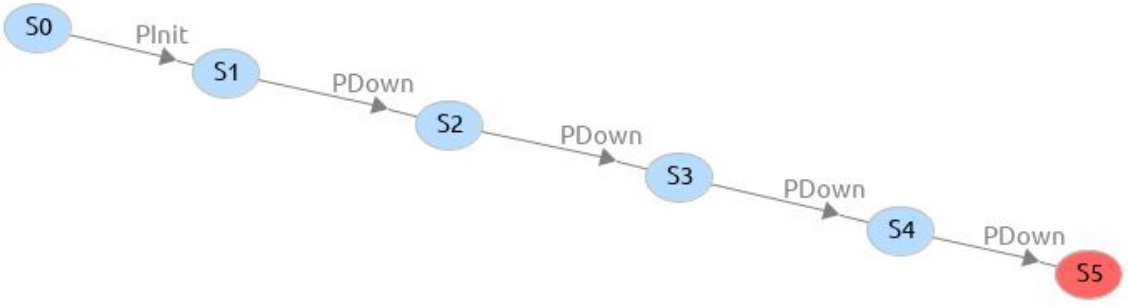


- 5 elementów:

Petri net state space analysis results

Bounded	true
Safe	false
Deadlock	true

Shortest path to deadlock: PInit PLeft PLeft PLeft PLeft

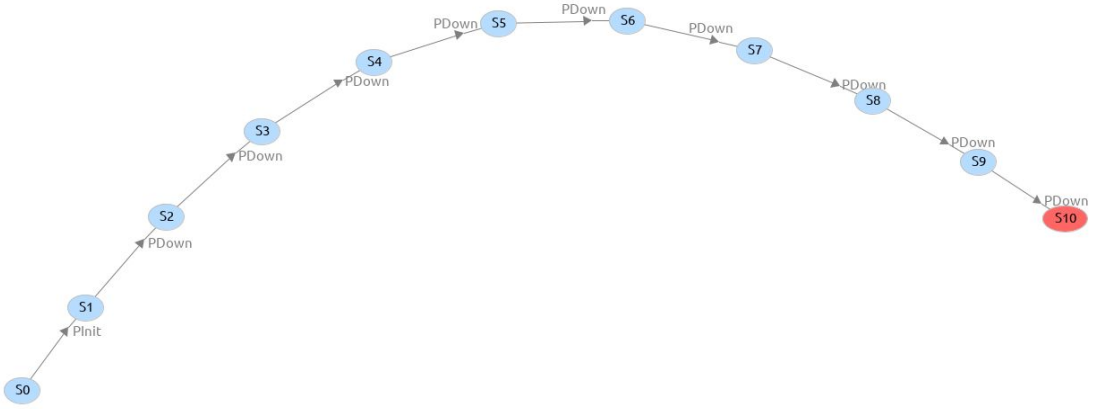


- 10 elementów:

Petri net state space analysis results

Bounded	true
Safe	false
Deadlock	true

Shortest path to deadlock: PInit PLeft PLeft PLeft PLeft PLeft PLeft PLeft PLeft PLeft



W każdym z powyższych przykładów sieć jest ograniczona i ma deadlock, jak również nie jest bezpieczna. Deadlock jest spowodowany faktem, że ilość tokenów w stanie P0 oznacza ilość elementów, które chcemy wygenerować, czyli po ilu generacjach nowych elementów sieć się zablokuje i wystąpi deadlock. Z tego powodu nie jest ona bezpieczna.