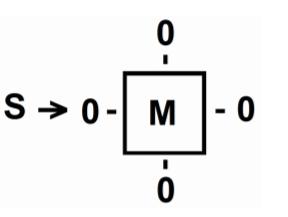


**Teoria współbieżności**

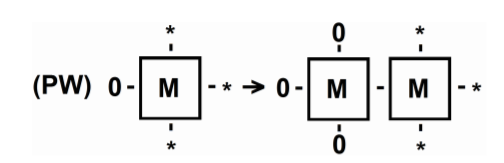
**Generowanie siatek 2D**

Kamil Koczera

**Produkcje:**

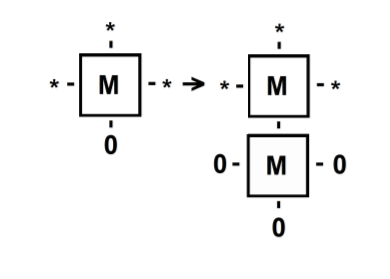
W celu wykonania zadania, do produkcji zawartych w instrukcji:

(PI) :

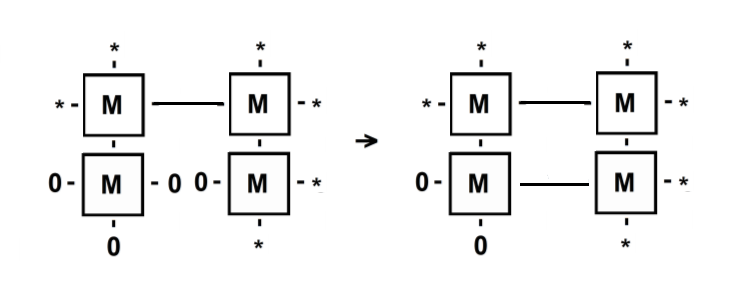


(PW):

Dodałem następujące:



(PS):



(PC):

**Generacja siatki 3x3:**

Możliwa np. za pomocą ciągu produkcji:

(PI) -> (PW) -> (PS) -> (PW) -> (PS) -> (PC) -> (PS) -> (PC) -> (PS) -> (PS) -> (PC) -> (PS) -> (PC)

Generacja przebiega następująco (uprościłem nieco zapis, ze względu na trudności związane z przedstawianiem operacji w dokumencie elektronicznym):

S -> (PI)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | M |

-> (PW)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | M | - | M |

-> (PS)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | M | - | M |
|  |  |  |  | | |
|  |  |  |  | M |

-> (PW)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| M | - | M | - | M |
|  |  |  |  | | |
|  |  |  |  | M |

-> (PS)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| M | - | M | - | M |
|  |  | | |  | | |
|  |  | M |  | M |

-> (PC)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| M | - | M | - | M |
|  |  | | |  | | |
|  |  | M | - | M |

-> (PS)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| M | - | M | - | M |
| | |  | | |  | | |
| M |  | M | - | M |

-> (PC)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| M | - | M | - | M |
| | |  | | |  | | |
| M | - | M | - | M |

-> (PS)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| M | - | M | - | M |
| | |  | | |  | | |
| M | - | M | - | M |
|  |  |  |  | | |
|  |  |  |  | M |

-> (PS)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| M | - | M | - | M |
| | |  | | |  | | |
| M | - | M | - | M |
|  |  | | |  | | |
|  |  | M |  | M |

-> (PC)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| M | - | M | - | M |
| | |  | | |  | | |
| M | - | M | - | M |
|  |  | | |  | | |
|  |  | M | - | M |

-> (PS)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| M | - | M | - | M |
| | |  | | |  | | |
| M | - | M | - | M |
| | |  | | |  | | |
| M |  | M | - | M |

-> (PC)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| M | - | M | - | M |
| | |  | | |  | | |
| M | - | M | - | M |
| | |  | | |  | | |
| M | - | M | - | M |

**Alfabet w sensie teorii śladów:**

Może zostać zdefiniowany w następujący sposób (powtarzające się produkcje rozróżnione za pomocą indeksów dolnych):

A = {PI} {PWn, n {1, 2}} {PSn, n {1, 2, 3, 4, 5, 6}} {PCn, n {1, 2, 3, 4}}

**Słowo odpowiadające generacji siatki:**

PI, PW1, PS1, PW2, PS2, PC1, PS3, PC2, PS4, PS5, PC3, PS6, PC4

**Relacja (nie)zależności dla alfabetu A:**

D = sym{{(PI, PW1), (PI, PS1), (PW1, PW2), (PW1, PS2), (PS2, PC1), (PW2, PS3), (PS3, PC2), (PS1, PS4), (PS2, PS5), (PS5, PC3), (PS3, PS6), (PS6, PC4)}+} IA

I = AxA – D

**Postać normalna Foaty:**

FNF = [PI][PW1, PS1][PW2. PS2, PS4][PS3, PC1, PS5][PC2, PS6, PC3][PC4]

Posługując się powyższymi klasami Foaty otrzymana zostaje generacja o następującym przebiegu (dla siatki 3x3):

S -> (PI)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | M |

-> (PW) -> (PS)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | M | - | M |
|  |  |  |  | | |
|  |  |  |  | M |

-> (PW) -> (PS) -> (PS)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| M | - | M | - | M |
|  |  | | |  | | |
|  |  | M |  | M |
|  |  |  |  | | |
|  |  |  |  | M |

-> (PS) -> (PC) -> (PS)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| M | - | M | - | M |
| | |  | | |  | | |
| M |  | M | - | M |
|  |  | | |  | | |
|  |  | M |  | M |

-> (PC) -> (PS) -> (PC)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| M | - | M | - | M |
| | |  | | |  | | |
| M | - | M | - | M |
| | |  | | |  | | |
| M |  | M | - | M |

-> (PC)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| M | - | M | - | M |
| | |  | | |  | | |
| M | - | M | - | M |
| | |  | | |  | | |
| M | - | M | - | M |

**Algorytm współbieżny**

Zaprojektowany został algorytm, który prezentuje się następująco (dla generowania siatki NxN):

* Do elementu wysuniętego najbardziej na zachód przyłącz nowy element z jego lewej (zachodniej) strony. Następnie do tego samego elementu (nie nowo przyłączonego) przyłącz nowy element z dołu (południa). Do każdego elementu znajdującego się na przekątnej (w kierunku południowo-wschodnim) (w stosunku do elementu, na którym wykonywaliśmy operacje przyłączania) przyłącz jego prawego (wschodniego) sąsiada oraz przyłącz nowy element z dołu (południa). Powtarzaj ten krok do osiągnięcia żądanej wysokości (szerokości).
* Do elementu wysuniętego najbardziej na zachód (niemającego sąsiada z dołu (południa)) przyłącz do niego jego prawego (wschodniego) sąsiada oraz nowy element z dołu (południa). Powtórz tę operację dla każdego elementu znajdującego się na przekątnej (w kierunku południowo-wschodnim) (w stosunku do elementu, na którym wykonywaliśmy operacje przyłączania), chyba, że jest to ostatni element (najbardziej wysunięty na południe) – wówczas pomiń operację przyłączania z dołu (południa). Powtarzaj ten krok do momentu ukończenia siatki (powinien zostać wykonany łącznie N razy).

Implementacja powyższego algorytmu znajduje się w dołączonym archiwum. Jest to zmodyfikowana wersja programu z laboratorium.

Przykład generacji siatki dla N = 5:

S -> (PI)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  | M |

-> (PW) -> (PS)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | M | - | M |
|  |  |  |  |  |  |  |  | | |
|  |  |  |  |  |  |  |  | M |

-> (PS) -> (PW) -> (PS)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | M | - | M | - | M |
|  |  |  |  |  |  | | |  | | |
|  |  |  |  |  |  | M |  | M |
|  |  |  |  |  |  |  |  | | |
|  |  |  |  |  |  |  |  | M |

-> (PW) -> (PS) -> (PS) -> (PS) -> (PC)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | M | - | M | - | M | - | M |
|  |  |  |  | | |  | | |  | | |
|  |  |  |  | M |  | M | - | M |
|  |  |  |  |  |  | | |  | | |
|  |  |  |  |  |  | M |  | M |
|  |  |  |  |  |  |  |  | | |
|  |  |  |  |  |  |  |  | M |

-> (PW) -> (PS) -> (PC) -> (PC) -> (PS) -> (PS) -> (PS)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| M | - | M | - | M | - | M | - | M |
|  |  | | |  | | |  | | |  | | |
|  |  | M |  | M | - | M | - | M |
|  |  |  |  | | |  | | |  | | |
|  |  |  |  | M |  | M | - | M |
|  |  |  |  |  |  | | |  | | |
|  |  |  |  |  |  | M |  | M |
|  |  |  |  |  |  |  |  | | |
|  |  |  |  |  |  |  |  | M |

-> (PS) -> (PS) -> (PS) -> (PS) -> (PC) -> (PC) -> (PC)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| M | - | M | - | M | - | M | - | M |
| | |  | | |  | | |  | | |  | | |
| M |  | M | - | M | - | M | - | M |
|  |  | | |  | | |  | | |  | | |
|  |  | M |  | M | - | M | - | M |
|  |  |  |  | | |  | | |  | | |
|  |  |  |  | M |  | M | - | M |
|  |  |  |  |  |  | | |  | | |
|  |  |  |  |  |  | M |  | M |

-> (PS) -> (PS) -> (PS)-> (PC) -> (PC) -> (PC) -> (PC)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| M | - | M | - | M | - | M | - | M |
| | |  | | |  | | |  | | |  | | |
| M | - | M | - | M | - | M | - | M |
| | |  | | |  | | |  | | |  | | |
| M |  | M | - | M | - | M | - | M |
|  |  | | |  | | |  | | |  | | |
|  |  | M |  | M | - | M | - | M |
|  |  |  |  | | |  | | |  | | |
|  |  |  |  | M |  | M | - | M |

-> (PS) -> (PS) -> (PC) -> (PC) -> (PC)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| M | - | M | - | M | - | M | - | M |
| | |  | | |  | | |  | | |  | | |
| M | - | M | - | M | - | M | - | M |
| | |  | | |  | | |  | | |  | | |
| M | - | M | - | M | - | M | - | M |
| | |  | | |  | | |  | | |  | | |
| M |  | M | - | M | - | M | - | M |
|  |  | | |  | | |  | | |  | | |
|  |  | M |  | M | - | M | - | M |

-> (PS) -> (PC) -> (PC)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| M | - | M | - | M | - | M | - | M |
| | |  | | |  | | |  | | |  | | |
| M | - | M | - | M | - | M | - | M |
| | |  | | |  | | |  | | |  | | |
| M | - | M | - | M | - | M | - | M |
| | |  | | |  | | |  | | |  | | |
| M | - | M | - | M | - | M | - | M |
| | |  | | |  | | |  | | |  | | |
| M |  | M | - | M | - | M | - | M |

-> (PC)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| M | - | M | - | M | - | M | - | M |
| | |  | | |  | | |  | | |  | | |
| M | - | M | - | M | - | M | - | M |
| | |  | | |  | | |  | | |  | | |
| M | - | M | - | M | - | M | - | M |
| | |  | | |  | | |  | | |  | | |
| M | - | M | - | M | - | M | - | M |
| | |  | | |  | | |  | | |  | | |
| M | - | M | - | M | - | M | - | M |