CS 580 Homework #01 Game of life. Jin Zhang 7788 Yie Sheng Chen 1303

Question 1,

Problem Definition:

Simulate Conway's Game of Life where births and deaths take place one at a time nondeterministically

Design Concept:

We used the "box" to make the nondeterministic one at a time update as the following unity program:

Unity program:

Program A1

delare

G: array [0..N+1,0..N+1] of integers

always

<[] i,j: 1<= i <= N ^ 1<= j <= N :: sum (i,j) = <+ x,y : -1<= x <= 1 ^ -1<= y <= 1 :: G(i+x,j+y)> - G[i,j] >

< [] i,j : 1<= i <= N ^ 1<= j <= N :: birth(i,j) := true if sum(i, j) - G[i, j] ==3 ~ false

initially

assign

```
< [] i,j : 1<= i <= N ^ 1<= j <= N ::
G[i,j] := 1 if birth(I,j) ~
```

0

end{A1}

Question 2,

Problem Definition:

Simulate Conway's Game of Life where births and deaths take place simultaneously over the entire community

Design Concept:

We used the "parallel bars" to make simultaneous update as the following unity program:

Unity Program:

Program A2

delare

G: array [0..N+1,0..N+1] of integers

Always

< [] i,j : 1<= i <= N ^ 1<= j <= N :: birth(i,j) := true if sum(i, j) - G[i, j] == 3 ~

false

initially

< [] i,j : 0<= i <= N+1 ^ j== 0 :: G[i,j] := 0 [] G[j,i] := 0 > [] < [] i,j : 0<= i <= N+1 ^ j== N+1 :: G[i,j] := 0 [] G[j,i] := 0 >

Assign

```
< || i,j : 1<= i <= N ^ 1<= j <= N ::
G[i,j] := 1 if birth(I,j) ~
0
```

end{A2}

Question 3,

Problem Definition:

Simulate Conway's Game of Life where births and deaths take place simultaneously within a single colony at a time.

Design Concept:

We construct neighborhood mask for each point using the concept of fix points: an synchronous group of operations updates the neighborhood mask for a particular point, (x,y), and after N such synchronous group of operations the neighborhood mask of (x,y) will reach fixed point and can be used to update births and deaths of a single colony. And such idea is implemented as the following unity program:

Unity Program:

Program A3

delare

G: array [0..N+1,0..N+1] of integers

NM: array [1..N,1..N,1..N,1..N] of boolean

Iter: array [1..N,1..N]

Always

```
<[] i,j: 1<= i <= N ∧ 1<= j <= N :: sum (i,j) = <+ x,y: -1<= x <= 1 ∧ -1<= y <= 1 :: G(i+x,j+y)> - G[i,j] >
```

< [] i,j : 1<= i <= N \land 1<= j <= N :: birth(i,j) := true if sum(i, j) - G[i, j] ==3 ~ false

```
<[] i,j,x,y : 1<= i <= N ∧ 1<= j <= N ∧ 1<= x <= N ∧ 1<= y <= N:: colony(x,y,i,j) := true if NM[x,y,i,j] ==1 ∨
```

>

initially

[] <| | i,j : 1<= i <= N ^ 1<= j <= N :: NM[I,j,I,j] := 1 >

assign

>end{A3}