

CSC466 Project Task 7

Single Point Evaluation Rule

Abstract

In this task I created a rule which evaluates each tile individually. If the number of unrevealed tiles adjacent to the tile is equal to the value of the tile, then all of the adjacent tiles are mines and will be flagged. Additionally, if the number of flagged adjacent tiles is equal to the value of the tile, then all other adjacent tiles which are not flagged can be safely revealed.

Demo

```
[1]> ( load "demos.l" )  
;; Loading file demos.l ...  
;; Loading file hms.l ...  
;; Loading file ms.l ...  
;; Loading file util.l ...  
;; Loaded file util.l  
;; Loaded file ms.l  
;; Loaded file hms.l  
;; Loaded file demos.l  
T
```

```
[2]> ( demo--single-point )
>>> Testing single point function
```

Board configuration

```
0 0 0 0 0 0 1 1 1 0
0 1 1 0 0 0 1 0 1 0
0 0 1 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0 0
A B C D E F G H I J
```

```
+-----+
0 | # # # # # # # # # # |
1 | # # # # # # # # # # |
2 | 1 3 # # # # # # # # |
3 | 0 1 1 # # # # # # # |
4 | 0 0 1 # # # # # # # |
5 | 0 0 1 # # # # # # # |
6 | 0 0 1 2 4 # # # # # |
7 | 0 0 0 0 1 # # # # # |
8 | 0 0 0 0 1 1 2 # # # |
9 | 0 0 0 0 0 0 1 # # # |
+-----+
```

Running single-point function

	A	B	C	D	E	F	G	H	I	J
0	#	#	#	#	#	#	#	#	#	#
1	#	#	#	#	#	#	#	#	#	#
2	1	3	>	#	#	#	#	#	#	#
3	0	1	1	1	#	#	#	#	#	#
4	0	0	1	2	#	#	#	#	#	#
5	0	0	1	>	>	>	#	#	#	#
6	0	0	1	2	4	4	#	#	#	#
7	0	0	0	0	1	>	#	#	#	#
8	0	0	0	0	1	1	2	#	#	#
9	0	0	0	0	0	0	1	#	#	#

Running single-point function

	A	B	C	D	E	F	G	H	I	J
0	#	#	#	1	0	2	#	#	#	#
1	#	#	#	2	0	2	#	#	#	#
2	1	3	>	2	0	1	1	3	#	#
3	0	1	1	1	0	0	0	1	#	#
4	0	0	1	2	3	3	2	1	#	#
5	0	0	1	>	>	>	#	#	#	#
6	0	0	1	2	4	4	#	#	#	#
7	0	0	0	0	1	>	2	#	#	#
8	0	0	0	0	1	1	2	#	#	#
9	0	0	0	0	0	0	1	#	#	#

Running single-point function

	A	B	C	D	E	F	G	H	I	J		
	+-----+											
0		#	#	2	1	0	2	>	#	#	#	
1		#	#	>	2	0	2	>	#	#	#	
2		1	3	>	2	0	1	1	3	#	#	
3		0	1	1	1	0	0	0	1	#	#	
4		0	0	1	2	3	3	2	1	#	#	
5		0	0	1	>	>	>	>	#	#	#	
6		0	0	1	2	4	4	#	#	#	#	
7		0	0	0	0	1	>	2	#	#	#	
8		0	0	0	0	1	1	2	#	#	#	
9		0	0	0	0	0	0	1	#	#	#	
	+-----+											

Running single-point function

	A	B	C	D	E	F	G	H	I	J
0	#	#	2	1	0	2	>	#	#	#
1	#	#	>	2	0	2	>	6	#	#
2	1	3	>	2	0	1	1	3	>	#
3	0	1	1	1	0	0	0	1	1	1
4	0	0	1	2	3	3	2	1	0	0
5	0	0	1	>	>	>	>	1	0	0
6	0	0	1	2	4	4	3	1	0	0
7	0	0	0	0	1	>	2	1	1	0
8	0	0	0	0	1	1	2	#	1	0
9	0	0	0	0	0	0	1	#	1	0

Running single-point function

	A	B	C	D	E	F	G	H	I	J
0	#	#	2	1	0	2	>	>	>	#
1	#	#	>	2	0	2	>	6	>	#
2	1	3	>	2	0	1	1	3	>	2
3	0	1	1	1	0	0	0	1	1	1
4	0	0	1	2	3	3	2	1	0	0
5	0	0	1	>	>	>	>	1	0	0
6	0	0	1	2	4	4	3	1	0	0
7	0	0	0	0	1	>	2	1	1	0
8	0	0	0	0	1	1	2	>	1	0
9	0	0	0	0	0	0	1	#	1	0

Running single-point function

	A	B	C	D	E	F	G	H	I	J
0	#	#	2	1	0	2	>	>	>	#
1	#	#	>	2	0	2	>	6	>	3
2	1	3	>	2	0	1	1	3	>	2
3	0	1	1	1	0	0	0	1	1	1
4	0	0	1	2	3	3	2	1	0	0
5	0	0	1	>	>	>	>	1	0	0
6	0	0	1	2	4	4	3	1	0	0
7	0	0	0	0	1	>	2	1	1	0
8	0	0	0	0	1	1	2	>	1	0
9	0	0	0	0	0	0	1	1	1	0

Running single-point function

	A	B	C	D	E	F	G	H	I	J		
	+-----+											
0		#	#	2	1	0	2	>	>	>	2	
1		#	#	>	2	0	2	>	6	>	3	
2		1	3	>	2	0	1	1	3	>	2	
3		0	1	1	1	0	0	0	1	1	1	
4		0	0	1	2	3	3	2	1	0	0	
5		0	0	1	>	>	>	>	1	0	0	
6		0	0	1	2	4	4	3	1	0	0	
7		0	0	0	0	1	>	2	1	1	0	
8		0	0	0	0	1	1	2	>	1	0	
9		0	0	0	0	0	0	1	1	1	0	
	+-----+											

NIL

```
[3]> ( demo--heuristic-game )
>>> Testing heuristic game player
Playing game on easy with display option
```

```

      A B C D E F G H I J
+-----+
0 | 0 1 1 1 0 0 0 0 1 > |
1 | 0 1 > 1 0 0 0 0 1 1 |
2 | 0 1 1 2 1 1 0 0 0 0 |
3 | 0 0 0 1 > 1 1 1 1 0 |
4 | 0 0 0 1 2 2 2 > 1 0 |
5 | 0 0 0 0 1 > 2 1 1 0 |
6 | 1 1 1 0 1 1 1 0 0 0 |
7 | 1 > 1 0 0 0 0 1 1 1 |
8 | 2 3 3 2 1 0 0 1 > 1 |
9 | > 2 > > 1 0 0 1 1 1 |
+-----+
```

```
Playing game on easy with stats option
(90 100)
```

```
Playing 100 games on easy
Average number of tiles revealed: 75
Number of wins: 76
```

```
Playing 100 games on medium
Average number of tiles revealed: 147
Number of wins: 35
```

Playing 100 games on hard

Average number of tiles revealed: 169

Number of wins: 1

NIL

[4]>

Stats

Stats on 100 games played						
	Easy		Medium		Hard	
Iteration	Avg Tiles	Wins	Avg Tiles	Wins	Avg Tiles	Wins
Random	58	2	85	0	116	0
Corner 1st	66	10	86	0	124	0
Single Point	75	76	147	35	169	1

Main Code

```
( defun rules ()
  ( list
    'single-point
    'corner-r
    'reveal-random
  )
)

( defun single-point ( &aux tiles u f res )
  ( setf tiles ( edge-tiles ) )
  ( dolist ( ti tiles )
    ( setf u ( adj-unrevealed ti ) )
    ( setf f ( adj-flags ti ) )
    ( cond
      ( ( = ( length u ) ( tile-value ti ) )
        ( dolist ( n u ) ( setf ( tile-flag n ) T ) )
      )
      ( ( = ( length f ) ( tile-value ti ) )
        ( dolist ( n ( set-difference u f ) )
          ( if ( reveal-tiles ( list n ) )
            ( return T ) )
          )
      )
    )
  )
  ( setf res ( edge-tiles ) )
  ( if ( equal res tiles ) 'na )
)
```

Util Functions

```
( defun nth-tile ( n )
  ( nth n ( board-tiles *board* ) )
)

( defun adj-tiles ( ti &aux l )
  ( setf l ( list
    ( tile-n ti )
    ( tile-ne ti )
    ( tile-e ti )
    ( tile-se ti )
    ( tile-s ti )
    ( tile-sw ti )
    ( tile-w ti )
  )
)
```

```

        ( tile-nw ti )
      ) )
      ( remove nil l )
    )
  ( defun revealed-tiles ( )
    ( remove-if-not #'tile-revealed ( board-tiles *board* ) )
  )
  ( defun unrevealed-tile ( ti )
    ( or
      ( not ( tile-revealed ti ) )
      ( tile-flag ti )
    )
  )
  ( defun unrevealed-tiles ( )
    ( remove-if-not #'unrevealed-tile ( board-tiles *board* ) )
  )
  ( defun adj-unrevealed ( ti &aux adj )
    ( setf adj ( adj-tiles ti ) )
    ( remove-if #'tile-revealed adj )
  )
  ( defun adj-flags ( ti &aux adh )
    ( setf adh ( adj-tiles ti ) )
    ( remove-if-not #'tile-flag adh )
  )

  ( defun edge-tiles ( &aux tiles l )
    ( setf tiles ( board-tiles *board* ) )
    ( setf l '() )
    ( dolist ( ti tiles )
      ( if
        ( and
          ( tile-revealed ti )
          ( /= ( length ( adj-unrevealed ti ) ) ( length ( adj-flags ti ) ) )
        )
        ( push ti l )
      )
    )
  )
  l
)

```