DSL: Autómatas Celulares

Damián Ariel Marotte

18 de febrero de 2019

Índice

1. Idea general

Definir un lenguaje de dominio especifico que permita indicar el comportamiento de automatas celulares, junto con una función de observación que dado un estado inicial, represente el comportamiento del autómata.

2. Alcances

Deberán poder especificarse autómatas en términos de vencidad de Moore y vecindad de von Neumann, con funciones de transición que puedan contar y comparar elementos de su vecindad.

3. Gramática

```
AUTOMATA   ::= STATES
               RULES
STATES     ::= Lambda
             | 'CHAR': 'CHAR' COLOR COLOR STATES
COLOR      ::= Black | Red | Green | Yellow | Blue | Magenta | Cyan | White
RULES      ::= Lambda
             | State == 'CHAR' && COMPARISON: 'CHAR' RULES
             | State == 'CHAR': 'CHAR' RULES
COMPARISON ::= DISTANCE('CHAR',INT) CMP INT && COMPARISON
             | CARDINAL(INT) CMP 'CHAR' && COMPARISON
             | DISTANCE('CHAR',INT) CMP INT
             | CARDINAL(INT) CMP 'CHAR'
CMP        ::= == | <= | >= | < | > | !=
DISTANCE   ::= Chebyshev | Manhattan
CARDINAL   ::= North | South | East | West | NE | NW | SE | SW
```

4. Ejemplos

4.1. Game of Life

```
' ': ' ' Black Black
'*': ' ' White White

State == '*' && Chebyshev('*',1) <  2: ' '
State == '*' && Chebyshev('*',1) == 2: '*'
State == '*' && Chebyshev('*',1) == 3: '*'
State == '*' && Chebyshev('*',1) >  3: ' '
State == ' ' && Chebyshev('*',1) == 3: '*'
```

4.2. Brian's Brain

```
' ': ' ' Black Black
'*': ' ' White White
'-': ' ' Blue  Blue

State == ' ' && Chebyshev('*',1) == 2: '*'
State == '*': '-'
State == '-': ' '
```

4.3. Seeds

```
'*': ' ' White White
' ': ' ' Black Black

State == ' ' && Chebyshev('*',1) == 2: '*'
State == '*': ' '
```

4.4. Wireworld

```
' ': ' ' Black  Black
'*': ' ' Blue   Blue
'-': ' ' Red    Red
'+': ' ' Yellow Yellow

State == '*': '-'
State == '-': '+'
State == '+' && Chebyshev('*',1) == 1: '*'
State == '+' && Chebyshev('*',1) == 2: '*'
```

4.5. Langton's Ant

```
'2': 'v' Red   Black
'4': '<' Red   Black
'6': '>' Red   Black
'8': '^' Red   Black
'3': 'v' Red   White
'5': '<' Red   White
'7': '>' Red   White
'9': '^' Red   White
' ': ' ' Black Black
'*': ' ' White White

State == '2': '*'
State == '4': '*'
State == '6': '*'
State == '8': '*'
State == '3': ' '
State == '5': ' '
State == '7': ' '
State == '9': ' '
```

```
State == ' ' && North(1) == '6': '2'
State == ' ' && West(1)  == '8': '6'
State == ' ' && East(1)  == '2': '4'
State == ' ' && South(1) == '4': '8'
State == '*' && North(1) == '6': '3'
State == '*' && West(1)  == '8': '7'
State == '*' && East(1)  == '2': '5'
State == '*' && South(1) == '4': '9'
State == ' ' && South(1) == '7': '8'
State == ' ' && East(1)  == '9': '4'
State == ' ' && West(1)  == '3': '6'
State == ' ' && North(1) == '5': '2'
State == '*' && South(1) == '7': '9'
State == '*' && East(1)  == '9': '5'
State == '*' && West(1)  == '3': '7'
State == '*' && North(1) == '5': '3'
```