Torre de babilonia

Instituto Tecnológico de Costa Rica

Ingeniería en Computación

Inteligencia Artificial – IC6200

Segundo Semestre

Edisson Lopez

Alonso Rivas Solano 2014079916

2018

Contenido

[Gramática 2](#_Toc525131446)

[Explicación 2](#_Toc525131447)

[Ejemplo 2](#_Toc525131448)

[Expresiones regulares 2](#_Toc525131449)

[Plan de pruebas 3](#_Toc525131450)

[Tabla 4](#_Toc525131451)

[Clase tabla 4](#_Toc525131452)

# Gramática

## Explicación

La gramática debe ir con únicamente cuatro elementos por fila separados por comas, sin espacio y deben ir 5 filas, no se pueden más ni menos. Luego se cuenta con 6 distintos elementos para usar, sin embargo, solo se tienen 4 elementos de las letras (R, G, B, Y), tres elementos de la letra X y solo un elemento de la letra O. La sumatoria total de elementos es 20. En caso de que se utilice un elemento repetido más veces de las cantidades permitidas, el algoritmo no funcionara.

Ejemplo

X,R,X,X

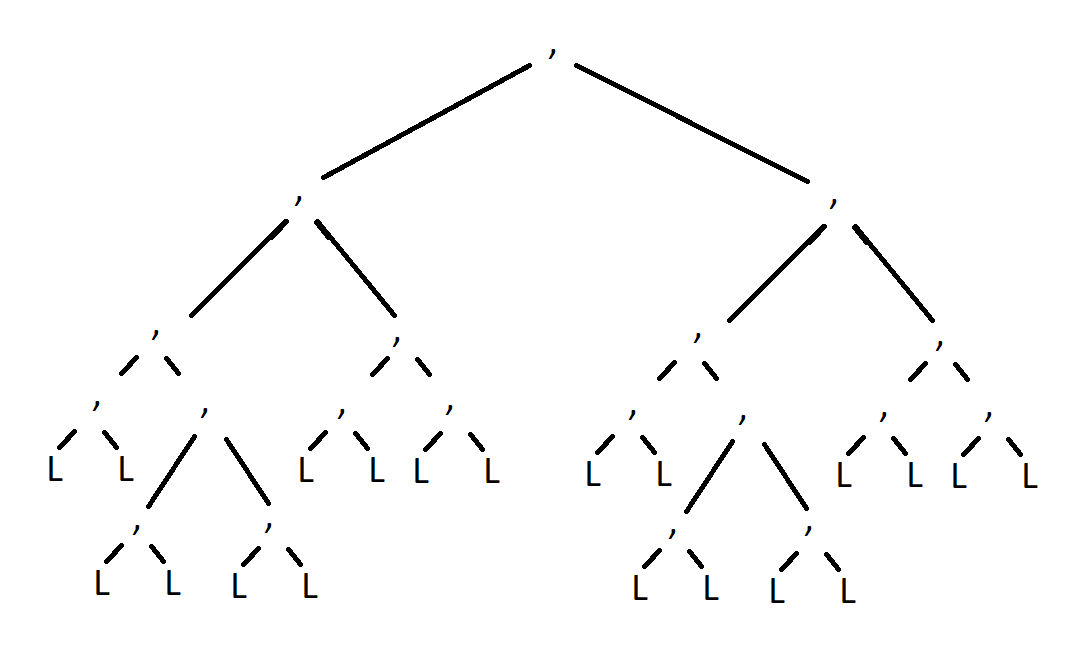
G,O,B,Y

Y,Y,G,B

Y,G,B,G

B,R,R,R

## Expresiones regulares



# Plan de pruebas

## Tabla

### Clase tabla

#### CalcularDistancia()

##### a. Una descripción del universo de entradas

La entrada de este módulo va a ser las posiciones i,j dentro de la tabla a la cual se le quiere buscar un color igual al contenido y otras posiciones i,j de la tabla meta a la que se quiere llegar, para así buscar el color igual a el más cercano. Se mide por filas y columnas, entonces se toman las coordenadas actuales, las meta y se restan, a la diferencia se le aplica el valor absoluto. En caso de que la segunda entrada, o sea la j, sea tres, se cambia por un 1.

##### b. los casos mínimos, máximos (si los hubiere), típicos, excepcionales

El caso mínimo que se puede insertar es el (1,0) y el máximo seria (4,3)

##### c. entradas no pertenecientes al universo (que el módulo debe detectar y rechazar)

cualquiera en que la primera entrada de la tupla fuere (1 < i < 4, 0 < j < 3). No hay típicos ni excepcionales.

##### d. las pruebas, esto es, los ejemplos de cada caso con sus correspondientes resultados.

En esta depende de la posición actual de la tabla que ser está usando y la tabla meta.

(1,2) , (3,4)

j = abs(jactual-jdestino)

if (j==3):

j=1

D = abs(iacutal-idestino) + j

D = 4

### Clase celda

#### SetColor()

##### a. Una descripción del universo de entradas

De entrada, se tiene que elegir uno entre los 6 colores disponibles para colocarle el color en la celda, siempre y cuando no rompa ninguna de las reglas establecidas anteriormente.

##### b. los casos mínimos, máximos (si los hubiere), típicos, excepcionales

El caso mínimo y máximo solo es 1. No hay típicos ni excepcionales.

##### c. entradas no pertenecientes al universo (que el módulo debe detectar y rechazar)

Cualquiera que no sea el color ya definidos.

##### d. las pruebas, esto es, los ejemplos de cada caso con sus correspondientes resultados.

En esta depende de la posición actual de la tabla que ser está usando y la tabla meta.

(‘R’)

self.color = color

### PrimerProyectoLogica

#### Algoritmo

##### a. Una descripción del universo de entradas

La entrada de este módulo va a ser la cantidad de movimientos que se ha hecho desde la tabla inicial para llegar hasta esa tabla y la tabla actual en si.

##### b. los casos mínimos, máximos (si los hubiere), típicos, excepcionales

El caso mínimo puede ser 0 movimientos por parte de la cantidad de movimientos que se ha hecho desde la tabla inicial y el máximo no tendría. No hay típicos ni excepcionales.

##### c. entradas no pertenecientes al universo (que el módulo debe detectar y rechazar)

Cualquiera numero negativo para la cantidad de movimientos.

##### d. las pruebas, esto es, los ejemplos de cada caso con sus correspondientes resultados.

def Algoritmo(g, tabla):

Sum=0

for i in range(Columnas):

for j in range(Filas):

#color que queremos sacarle la distancia

color = tabla.ObtenerColor(i,j)

#if (color == "R" or color == "G" or color == "B" or color == "Y" ):

(idestino,jdestino) = tabla.CalcularIJdelColorMasCercano(i,j)

DistanciaCalculada = CalcularDistancia(i, j, idestino, jdestino)

Sum = Sum + DistanciaCalculada

Tabla.GuardarPeso(self, (g + 1/20 \* Sum))

return

#### Calculardistancia

##### a. Una descripción del universo de entradas

De entrada, se tiene que elegir uno entre los 6 colores disponibles para colocarle el color en la celda, siempre y cuando no rompa ninguna de las reglas establecidas anteriormente.

##### b. los casos mínimos, máximos (si los hubiere), típicos, excepcionales

El caso mínimo y máximo solo es 1. No hay típicos ni excepcionales.

##### c. entradas no pertenecientes al universo (que el módulo debe detectar y rechazar)

Cualquiera que no sea el color ya definidos.

##### d. las pruebas, esto es, los ejemplos de cada caso con sus correspondientes resultados.

En esta depende de la posición actual de la tabla que ser está usando y la tabla meta.

(‘R’)

self.color = color