

PROCÉS DE LECTURA EN UN AF

Documentació del codi

MARC CERVERA ROSELL – 47980320C

EPS - UDL

Llenguatges, autòmats i gramàtiques

Josep Maria Miret Biosca

Índex

Introducció:	1
Codi:	1
Variables d'instància:	1
Mètodes:	1
run():	1
wordReading():	2
wordAcceptation():	2
index():	3
transitionFunctionMethod():	3
dataViewing():	3
alphabetIntroduction():	4
finalStatesIntroduction():	4

Introducció:

En aquest cas pràctic, el professorat ha donat a escollir diverses opcions d'implementació en relació a la lliçó referent a autòmats i també ha donat l'opció d'implementar temes en relació a la lliçó de gramàtiques. En aquesta ocasió, s'ha escollit implementar un programa que llegeix un autòmat amb tota la seva informació; nombre d'estats, estats finals, alfabet i funció de transició. Un cop introduïda aquesta informació per part de l'usuari, aquest ha d'introduir una paraula, la qual serà llegida per l'autòmat introduït prèviament per l'usuari i finalment, el programa retornarà un missatge tot especificant si la paraula ha estat acceptada o no.

L'equip docent va especificar que la implementació d'aquest cas pràctic es podia fer en el llenguatge de programació que l'alumnat considerés més adient en cada cas. En aquest cas, s'ha realitzat en el llenguatge Java.

Codi:

En aquest apartat s'explicaran els trets més importants de cada part del codi implementat, en aquesta ocasió en el llenguatge Java.

Variables d'instància:

Les variables d'instància de la classe "LecturaAutomat", són variables públiques, estàtiques i finals que contenen els codis ANSI dels colors que es faran servir per treure missatges per pantalla. S'ha optat per utilitzar diferents colors, atès que en cas contrari, tot el text s'imprimeix en color blanc i pot resultar una mica dificultós la seva lectura.

Per tant, aquestes susdites variables, tenen el següent patró:

```
public static final String ANSI_NOM-COLOR = "codi-color";
```

Mètodes:

En aquest apartat es descriuran els trets més importants dels mètodes emprats per a la implementació de l'algorisme final.

run():

Aquesta, és la funció principal i la que determina l'ordre d'execució del programa.

Primerament, és demana per teclat diferents dades, necessàries, per definir l'autòmat. Aquestes dades inclouen el nombre d'estats, el nombre d'estats finals, els estats finals, el nombre de símbols de l'alfabet, l'alfabet, la funció de transició, i la paraula a llegir.

Abans de demanar la paraula a llegir, però, es fa una crida a un mètode que, senzillament escriu les dades per pantalla.

Un cop introduïdes les dades, la paraula i fets els *prints* necessaris, és moment de tractar la paraula, amb el mètode "*wordReading()*".

Finalment, segons l'estat en el qual acabi el procés de lectura, es determinarà si la paraula és acceptada o no.

wordReading():

Aquest, és el mètode més important de tots, si més no, sense ell, aquesta pràctica no podria ser possible atès que no podríem llegir la paraula.

Aquest mètode, rebrà com a paràmetres la paraula, la funció de transició i l'alfabet.

El que farà aquesta funció, és per cada símbol de la paraula, provar de obtenir l'índex que el símbol que s'està tractant en aquell moment ocupa en l'array de l'alfabet. Si no s'obtingués aquest índex, no és podria realitzar el següent pas, que es busca en la matriu de la funció de transició quin és el següent estat. La matriu de la funció de transició és una matriu de dimensions "nombre d'estats X nombre de símbols de l'alfabet". Un cop conegut quin és aquest estat, es guarda en una variable que s'anomenarà "*nextState*".

A continuació, es treu per pantalla la informació referent a la transició (estat actual, símbol llegit i estat següent).

Finalment, s'actualitza l'estat actual.

En cas que aquestes operacions no siguin possibles, per exemple, perquè s'ha introduït un AFN i la paraula necessita d'una transició que no ha estat definida, el programa tallarà la seva execució tot llençant una excepció amb el missatge "*Transition not defined*".

Per acabar la implementació d'aquest mètode de lectura de la paraula (en cas que no s'hagi llençat la excepció), es treu un missatge per pantalla especificant quin ha estat l'estat on ha acabat la màquina la lectura i es retorna aquest mateix estat per a posteriorment decidir si la paraula ha d'ésser acceptada o no.

wordAcceptation():

Aquest mètode, és el que decideix si una paraula ha d'ésser acceptada o no.

La manera en que actua és senzilla. Rep com a paràmetres l'estat en el qual acaba la lectura i l'array d'estats finals i senzillament busca si l'estat on ha acabat l'autòmat està en l'array de estats finals. Si hi és es retorna *true* (paraula acceptada) i si no hi és, es retorna *false* (paraula no acceptada).

index():

Aquest mètode retorna l'índex que ocupa el símbol que s'està tractant en l'array on està comprès l'alfabet, per tant, aquest mètode rep com a paràmetres l'alfabet i el símbol.

La manera de buscar aquest índex, es basa en recórrer l'array de l'alfabet fins a comprovar que el símbol i la posició de l'array coincideixen tot guardant l'índex de la posició de l'array.

Finalment, es retorna aquest índex.

NOTA: com ja s'ha esmentat anteriorment, aquest índex s'usa per accedir a la matriu de la funció de transició.

transitionFunctionMethod():

Aquesta funció, emmagatzema i retorna una matriu de dimensions "nombre d'estats X nombre de símbols de l'alfabet".

Aquesta matriu, contindrà la funció de transició, on tindrem, "nombre d'estats" files i "nombre de símbols de l'alfabet" columnes (dades rebudes com a paràmetre del mètode).

Cal remarcar que en aquest programa, els estats es numeren de 0 a $n - 1$, és a dir, si l'usuari introdueix un total de 5 estats, la numeració dels estats serà des de 0 fins a 4.

El procés, es basarà en anar introduint les transicions corresponents a cada estat i a cada símbol llegit. En cas de que l'autòmat sigui AF i una transició no estigui definida, el valor a introduir en la funció de transició serà -1.

Finalment, es retorna aquesta matriu.

dataViewing():

Aquest mètode senzillament treu per pantalla totes les dades introduïdes per l'usuari, és a dir, treu per pantalla:

- Nombre d'estats de la màquina.
- Nombre d'estats finals.
- Array d'estats finals.
- Nombre de símbols de l'alfabet.
- Array de símbols de l'alfabet.
- Matriu de la funció de transició.

Totes aquestes dades són rebudes pel mètode com a paràmetre del mateix.

alphabetIntroduction():

Aquest, és el mètode d'introducció de l'alfabet, és a dir, segons el nombre de símbols que hagi especificat l'usuari (el qual rebrà el mètode com a paràmetre), es generarà un vector amb aquell nombre de posicions i seguidament es demanarà per teclat els símbols de l'alfabet a l'usuari fins a emplenar el vector.

Finalment, es retornarà el vector el qual contindrà l'alfabet.

finalStatesIntroduction():

La tasca d'aquest últim mètode auxiliar, és retornar un vector que contindrà els estats finals de la màquina. És a dir, segons el nombre d'estats finals especificat per l'usuari (aquest nombre és rebut com a paràmetre), aquest mètode crearà un vector amb aquest nombre de posicions i seguidament demanarà per teclat els estats finals sempre tenint en compte que els estats es numeren de 0 a $n - 1$.

Finalment, es retorna aquest vector.