

Construcción de un AFD no mínimo de una ER

Que para coadyuvar en la calificación del primer parcial

Alumno

Instituto Politécnico Nacional

Agosto de 2018

Trabajo: transformar una ER en un autómata AFD no mínimo

Objetivo

Utilizar las funciones `anulable()`, `primerapos()`, `ultimapos()` y `siguientepos()`. Para construir el autómata AFD no mínimo.

DEFINICIÓN (Función `primerapos()`.)

La que proporciona el conjunto de posiciones que pueden concordar con el primer símbolo de una cadena generada por la subexpresión con raíz en n $[?]$.

Las funciones, continuación

DEFINICIÓN (Función últimapos().)

La que proporciona el conjunto de posiciones que pueden concordar con el último símbolo en esa cadena [?].

DEFINICIÓN (Función anulable().)

Es necesario conocer qué nodos son las raíces de las subexpresiones que generan lenguajes que incluyen la cadena vacía. A dichos nodos se les denomina anulables, y la función anulable(n) se define como verdadera si el nodo n es anulable, y falso en caso contrario [?].

DEFINICIÓN (Función siguientespos().)

*Es el conjunto de posiciones j tales que hay alguna cadena de entrada **cd** tal que **i** corresponde a la aparición de **c** y **j** a la aparición de **d** [?].*

DEFINICIÓN (Construcción de un AFD de una expresión regular.)

Entrada:

Una expresión regular r .

Salida:

Un AFD **D** que reconoce a $L(r)$.

Método:

- 1 Constrúyase un árbol sintáctico para la expresión regular aumentada $(r)\#$, donde $\#$ es un marcador de final único que se añade a (r) .
- 2 Constrúyanse las funciones `anulable()`, `primerapos()`, `últimapos()` y `siguientepos()` haciendo recorridos en profundidad en el árbol T .
- 3 Constrúyanse los **estadosD**, el conjunto de estados **D**, y **tranD**, la tabla de transiciones para **D**. Los estados dentro de **estadosD** son conjuntos de posiciones; al principio, cada estado esta “no marcado”, y un estado se convierte en “marcado” justo antes de considerar sus transiciones de salida. El estado de inicio de **D** es `primerapos(raíz)`, y los estados de aceptación son todos los que contienen la posición asociada con el marcador de final $\#$.

Listing 1: Construcción de la tabla de transiciones del estado D.

```
al principio , el unico estado no marcado en estadosD
es primerapos(raiz), donde raiz es la raiz del arbol
de sintaxis para (r)#;

while hay un estado sin marcar T en estadosD do begin
    marcar T;
    for cada simbolo de entrada a do begin
        sea U el conjunto de posiciones que estan en
            siguiente(p) para alguna posicion p
            en T, tal
                que el simbolo en la posicion p es a;
        if U no esta vacio y no esta en estadosD then
            aniadir U como estado no marcado a
                estadosD;
        tranD[T, a] := U
    end
end
```