

PRÁCTICA 1: Programar un simulador de un autómeta con pila

Objetivo

El objetivo de la práctica consiste en programar un simulador de un autómeta con pila, realizando un diseño orientado a objetos.

Entrega

La ejecución de la práctica será revisada en la sesión de entrega en el laboratorio.

Se debe entregar el código fuente en la tarea del aula virtual.

Fecha límite de entrega: 22 de octubre de 2021.

Notas de implementación

Se puede implementar el autómeta de pila con finalización por vaciado de pila (APv) o por estado final (APf). Únicamente es necesario implementar uno de los dos tipos.

Los elementos del autómeta con pila se introducirán en tiempo de ejecución del programa utilizando un fichero de texto con el siguiente formato:

Comentarios

q1 q2 q3 ... # conjunto Q

a1 a2 a3 ... # conjunto Σ

A1 A2 A3 ... # conjunto Γ

q1 # estado inicial

A1 # símbolo inicial de la pila

q2 q3 # conjunto F

q1 a^(*) A1 q2 A^(*) # función de transición: $(q2, A) \in \delta(q1, a, A1)$

... # cada una de las transiciones en una línea distinta

^(*) a es un símbolo de $(\Sigma \cup \epsilon)$. En el fichero el símbolo ϵ se representará por un punto (.)

^(*) A puede ser ϵ (que se escribirá también en el fichero mediante un punto) o estar formado por uno o más símbolos de Γ , que se escribirán separados por espacios en blanco. Por ejemplo: $A = A1 A1 A1$

El programa deberá funcionar con los ejemplos descargados del aula virtual.

El programa debe verificar que la información proporcionada en el fichero cumple con las restricciones de la definición formal de un autómata, por ejemplo: $s \in Q$.

Una vez cargada la definición del autómata, debe ser posible comprobar varias cadenas de entrada e indicar si pertenecen al lenguaje reconocido por el autómata. Las diferentes cadenas de entrada podrán ser introducidas por teclado o por fichero.

Como salida, el programa debe indicar si la palabra pertenece al lenguaje que reconoce el autómata de pila.

Se puede realizar el código utilizando los lenguajes C++ o Java.

Se debe utilizar un diseño orientado a objetos.

De forma adicional (no obligatoria), el programa podrá tener la opción de ejecutar en *modo traza*, donde se mostrará la información del autómata después de aplicar cada transición.