

Traductores de Lenguajes de Programación

Evaluación continua – Práctica 1

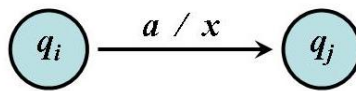
Un autómata finito traductor (AFDT) es un autómata finito determinista al que se asocia una función de traducción, que se utiliza para producir una salida. La finalidad de un AFDT es, por tanto, reconocer y al mismo tiempo “traducir” un determinado lenguaje regular. Formalmente, un AFDT es

$$M = (Q, \Sigma, \delta, q_0, F, \Delta, \psi),$$

donde

- Q es un conjunto finito de estados.
- Σ es el alfabeto de símbolos de entrada.
- $\delta: Q \times \Sigma \rightarrow Q$ es la función de transición.
- $q_0 \in Q$ es el estado inicial.
- $F \subseteq Q$ es el conjunto de estados finales.
- Δ es el alfabeto de símbolos de salida.
- $\psi: Q \times \Sigma \rightarrow \Delta^*$ es la función de traducción.

Diremos que la cadena $x \in \Delta^*$ traduce el símbolo $a \in \Sigma$ cuando se transita del estado q_i al estado q_j si y sólo si $\delta(q_i, a) = q_j$ y $\psi(q_i, a) = x$. Gráficamente se representa por el diagrama de transición / traducción



Se define la función de transición extendida $\hat{\delta}: Q \times \Sigma^* \rightarrow Q$ inductivamente por

- $\hat{\delta}(q, \varepsilon) = q$ para todo $q \in Q$.
- $\hat{\delta}(q, aw) = \hat{\delta}(\delta(q, a), w)$ para todo $q \in Q, a \in \Sigma$ y $w \in \Sigma^*$.

Asimismo, definimos la función de traducción extendida $\hat{\psi}: Q \times \Sigma^* \rightarrow \Delta^*$ por

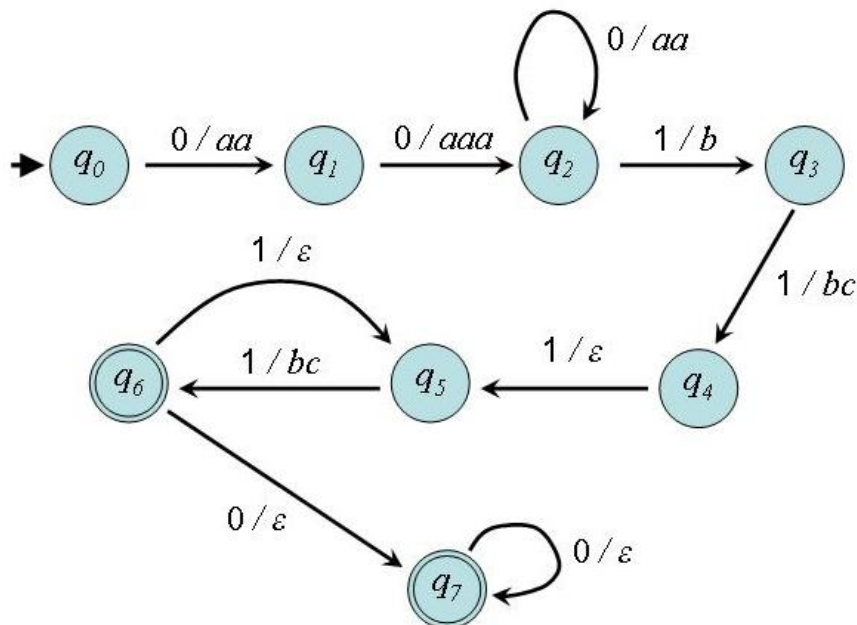
- $\hat{\psi}(q, \varepsilon) = \varepsilon$ para todo $q \in Q$.
- $\hat{\psi}(q, aw) = \psi(q, a) \cdot \hat{\psi}(\delta(q, a), w)$ para todo $q \in Q, a \in \Sigma$ y $w \in \Sigma^*$.

Finalmente, el lenguaje de traducción $T(M) \subseteq \Delta^*$ generado por un autómata finito traductor M es, por definición, el conjunto

$$T(M) = \bigcup_{w \in \Sigma^*} \{ \hat{\psi}(q_0, w) : \hat{\delta}(q_0, w) \in F \}.$$

Ejemplo. Sea el lenguaje regular $L = \{0^n 1^{2m} 0^k : n \geq 2, m \geq 2, k \geq 0\}$. El diagrama de la figura representa un autómata finito traductor M , que genera a partir de L el lenguaje de traducción

$$T(M) = \{a^{2n+1}b(bc)^m : n \geq 2, m \geq 2\}.$$



Se pide:

Programar un AFDT genérico M tal que a partir de un lenguaje regular L dado, genere el lenguaje de traducción asociado. En particular, se considerarán los siguientes casos:

- $L = \{ab^n a : n \geq 0\}$, $T(M) = \{c^{n+3} : n \geq 1\}$.
- $L = \{a^n b(ca)^{2k} w : n, k \geq 0, w \in \{a, b\}^*\}$, $T(M) = \{10^{k+1}1^m : k \geq 0, m = |w|\}$.
- $L = \{w \in \{a, b\}^* : w \text{ no termina en } b\}$,
 $T(M) = \{x \in \{0,1\}^* : \text{el valor en binario de } x \text{ es par}\}$.

Documentación

El programa se implementará en C de forma genérica, obteniendo por tanto un AFDT configurable. Los datos necesarios para el correcto funcionamiento del AFDT se obtendrán de un almacenamiento externo. La práctica se presentará acompañada de una documentación por escrito en la que se expongan los aspectos que se consideren más relevantes en el desarrollo realizado, en particular, debe haber una justificación razonada de la construcción y configuración del AFDT. No se admiten documentaciones manuscritas.

Evaluación

Esta práctica representa el 10% de la nota final dentro de un proceso de evaluación continua, y debe realizarse obligatoriamente y con suficiencia para que un alumno pueda ser evaluado positivamente en la asignatura. Para alcanzar esta evaluación positiva es requisito imprescindible presentar la documentación solicitada en el apartado anterior, que será valorada atendiendo a su contenido y calidad.