CENTRO SUPERIOR DE INFORMÁTICA
Departamento de Estadística, I.O. y Computación
Teoría de Autómatas y Lenguajes Formales

PRACTICA 10: Autómatas con Pila (PDAs)

10.1. Introducción

Un autómata a pila no determinista se define como una 7-upla:
$M \equiv (Q, \Sigma, \Gamma, q_0, F, Z, \delta)$
donde cada uno de estos elementos tiene el siguiente significado:

- Q es el conjunto finito de los estados del autómata. El autómata siempre se encontrará en uno de los estados de este conjunto.
- Σ es el alfabeto de entrada del autómata. Se trata del conjunto de símbolos que el autómata acepta como entradas.
- Γ es un conjunto finito de símbolos que constituye lo que se denomina el alfabeto de la pila.
- $q_0 \in Q$ es el estado inicial o de arranque del autómata. Se trata de un estado distinguido. El autómata se encuentra en este estado al comienzo de la ejecución.
- $Z \in \Gamma$ es el símbolo inicial de la pila.
- $F \subseteq Q$ es el conjunto de estados finales o de aceptación del autómata.
- δ es la función de transición. $\delta : Q$ x $(\Sigma \cup \{\epsilon\})$ x $\Gamma \rightarrow \mathfrak{P}(Q$ x $\Gamma^*)$

El término Autómata a pila no Determinista lo abreviaremos PDA por su traducción del inglés, (Pushdown Automaton).

El PDA comienza su ejecución estando en el estado de arranque y a la vista de un determinado símbolo en la entrada y otro símbolo en la pila (un elemento del alfabeto de pila) efectúa una transición por la que cambia su estado y coloca en la pila una secuencia de símbolos (un elemento de Γ^*)). El no determinismo del autómata viene dado por el hecho de que para una determinada terna de entrada (q, a, b) ($q \in Q, a \in \Sigma, b \in \Gamma$) el autómata puede realizar diferentes transiciones.

Por ejemplo podría ocurrir $\delta(q, a, b) = \{(q_1, abc), (q_2, abcde), (q_3, e)\}$, es decir el autómata podría elegir de forma no determinista por cambiar a uno de los estados q_1, q_2 o q_3 y en cada uno de estos casos colocar en la pila las cadenas “abc”, “abcde” o la cadena vacía, respectivamente.

Asípues, estrictamente hablando, la función δ no es realmente una función puesto que no tiene porqué estar definida para todos los triples posibles y también porque su resultado no es único, sino que en el caso general produce un conjunto de transiciones posibles.

Veamos un ejemplo de PDA definido como:
$Q = \{q_1, q_2, q_3, q_4\}$
$S = \{a, b\}$
$G = \{A, B\}$
$Z = A$
$q_0 = q_1$
$F = \{q_4\}$

La función de transición, δ destá definida por la tabla 10.1:

δ	(a, A)	(b, A)	(ϵ, A)	(a, B)	(b, B)	(ϵ, B)
q_1	$\{(q_2, BA), (q_4, A)\}$		$\{(q_4, \epsilon)\}$			
q_2				$\{(q_2, BB)\}$	$\{(q_3, \epsilon)\}$	
q_3			$\{(q_4, A)\}$		$\{(q_3, \epsilon)\}$	

Cuadro 10.1: La función de transición del PDA del ejemplo

En la tabla, las filas corresponden a los estados y las columnas a todos los posibles pares de símbolo de entrada y de la pila. Se observa en la tabla que no hay transiciones definidas para todos los posibles triples de entrada. Si el PDA pasa a un estado para el cual no se especifica un estado siguiente y una acción en la pila para la configuración actual, ello significa que el autómata detiene su actividad y no realizará ningún movimiento a partir de ese instante. Esto ocurre en el ejemplo si el autómata alcanza el estado q_4 para el que no hay transiciones definidas.

Durante el proceso de una cadena de símbolos de entrada se puede describir las sucesivas configuraciones por las que pasa el PDA utilizando lo que se llaman descripciones instantáneas. Una descripción instantánea para un PDA es una terna $(q, w, u) \in Q$ x Σ^* x Γ^* en la que q es el estado en el que se encuentra el autómata, w es la parte de la cadena de entrada que está pendiente de leer y u es el contenido de la pila (convendremos que el primer símbolo de esta cadena es el símbolo que se encuentra en el tope de la pila).

Un cambio de descripciones instantáneas se define por:
$(q, au, b\alpha) \vdash (p, u, \beta\alpha) \Leftrightarrow (p, \beta) \in \delta(q, a, b)$
donde el símbolo $\vdash$ se utiliza para representar el paso de una descripción instantánea a otra. Como es habitual se utilizan los símbolos $\vdash^+$ y $\vdash^*$ para indicar uno o más movimientos y cero o más movimientos respectivamente por parte del PDA.

Veamos un ejemplo de cómputo con el PDA del ejemplo utilizando la notación introducida. El autómata comienza su ejecución en el estado de arranque, teniendo en el tope de la pila el símbolo inicial de la pila y teniendo pendiente de leer toda la entrada, que supondremos que es la cadena “aabb”:
$(q_1, aabb, A) \vdash (q_2, abb, BA) \vdash (q_2, bb, BBA) \vdash (q_3, b, BA) \vdash (q_3, e, A) \vdash (q_4, e, A)$

Cuando el autómata alcanza el estado q_4 ya no hay más transiciones posibles y por ello detiene su ejecución. Obsérvese también que el no determinismo del PDA podría haberle llevado en la primera transición de esta secuencia a elegir una alternativa diferente.

Sea $M \equiv (Q, \Sigma, \Gamma, q_0, F, Z, \delta)$ un PDA. Se define el lenguaje aceptado por M como:
$L(M) = \{w \in \Sigma^* | (q_0, w, Z) \vdash (p, \epsilon, u)$ para $p \in F$ y $u \in \Gamma^*\}$

Es decir, según esta definición las cadenas que pertenecen a $L(M)$ son aquellas que hacen que el PDA pase de su descripción instantánea inicial a una descripción en la que alcanza un estado de aceptación habiendo consumido todos los símbolos de la entrada, y pudiendo terminar la ejecución con la pila vacía o no.

Una definición alternativa para el lenguaje aceptado por un PDA es la siguiente:
$N(M) = \{w \in \Sigma^* | (q_0, w, Z) \vdash (p, \epsilon, \epsilon)$ para $p \in Q\}$

En este caso se habla del lenguaje aceptado por el autómata “por pila vacía”. En general tendremos que $N(M) \neq L(M)$. No obstante se puede demostrar que ambas definiciones son equivalentes en el sentido de que dado un PDA M siempre se puede hallar otro M' tal que $N(M) = L(M')$ y viceversa.

Se dice que un PDA es determinista si ante cada descripción instantánea sólo hay un movimiento posible. Para que un PDA sea determinista se ha de dar una de las dos condiciones siguientes:

1. $Card(\delta(q, a, x)) \leq 1 \forall a \in \Sigma$ y $Card(\delta(q, \epsilon, x)) = 0$
2. $Card(\delta(q, a, x)) = 0 \forall a \in \Sigma$ y $Card(d(q, e, x)) \leq 1$

La primera condición impone que no haya transiciones sin consumir ningún símbolo de entrada y que el conjunto de transiciones posibles para una entrada tenga un único elemento. La segunda condición permite transiciones sólo sin consumir ningún símbolo de la cadena de entrada.

10.2. Simulaciónde un PDA determinista

La práctica consistirá en la realización de un programa en C que lea desde un fichero las especificaciones de un PDA determinista, a continuación comience a imprimir una traza de los estados por los que transita el autómata mediante las entradas en forma de caracteres que se leerán desde otro fichero, y al final indique si la entrada se acepta o no. Tanto el nombre del fichero de definición del PDA como el de entradas para el mismo se leerán en la línea de comandos del programa.

Un parámetro adicional pasado también en la línea de comandos indicará también si el autómata reconocerá/rechazará la cadena de entrada por pila vacía o por estado de aceptación, de forma que al final del proceso de una cadena el programa imprimirá en pantalla: ”La cadena X pertenece (no pertenece) a L(M)” o bien ”La cadena X pertenece (no pertenece) a N(M)”.

Todos los ficheros que se usarán serán ficheros de texto. Los estados del autómata se representarán mediante números enteros sin signo (`unsigned`), y la numeración de los estados corresponderá a los primeros números naturales comenzando con 0.

10.2.1. Estructurade los ficheros de definición de autómatas

Estos ficheros tendrán la extensión `.PDA` y el siguiente formato:

- Línea 1: El número total de estados del PDA.
- Línea 2: Estado de arranque del PDA.
- Línea 3: Símbolo inicial de la pila.

A continuación viene una línea para cada estado, conteniendo los siguientes números, separados entre sí por espacios en blanco.

- Número identificador del estado.
- Un 1 si se trata de un estado de aceptación o un 0 si es un estado de rechazo.
- Número de transiciones que posee el estado.

A continuación para cada una de las transiciones y separados por espacios:

- Símbolo de entrada necesario para que se produzca la transición.
- Símbolo de pila necesario para la transición
- Estado destino de la transición.
- Cadena de símbolos de pila que se han de apilar (esta cadena se colocará entre comillas dobles, puesto que su longitud es variable)

La Figura 10.1 muestra un fichero de definición de PDA. Este PDA se ha construido partiendo del PDA que se ha utilizado como ejemplo, modificando las transiciones del estado q_1 para hacer que el PDA resultante sea determinista.

```
4
1
1 0 1 a A 2 "BA"
2 0 2 a B 2 "BB" b B 3 ~
3 0 2 ~ A 4 "A" b B 3 ~
4 1 0
```

Figura 10.1: Fichero de definición del autómata de la Figura 10.1 (modificado)

10.2.2. Ficheros de entradas

Estos ficheros tendrán la extensión .in Se supondrá que los alfabetos de entrada y de pila están constituidos por todos los caracteres alfabéticos. Las frases de entrada al autómata figuran en los ficheros de entrada sin separación entre los caracteres de la misma. Así para el autómata anterior podríamos tener la entrada: aabb