Teoría de la Computación

Víctor Melchor

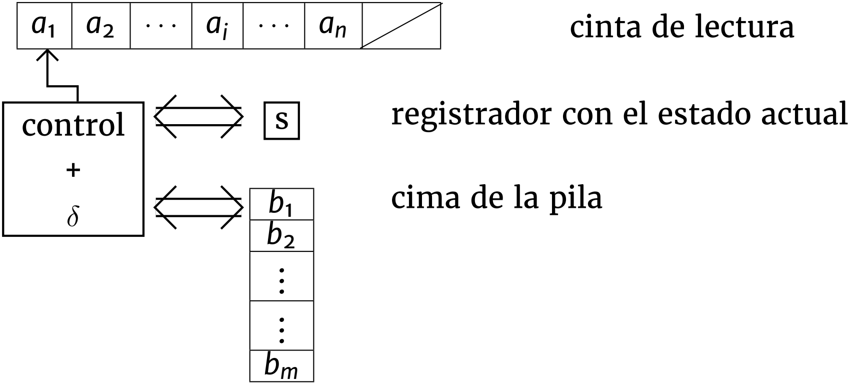
Clase 23

Teoría de Autómatas

Capítulo 1

Autómatas PushDown (PDA)

Un autómata pushdown puede ser esbozado como una máquina similar a:



Al igual que la cinta, la pila se divide en celdas que almacenan un símbolo cada una, pero el cabezal de lectura de la pila sólo se posiciona en la celda de la cima de la pila.

En un principio, el registrador contiene el estado inicial del PDA, la cinta contiene la palabra de entrada a partir de su primera celda; el cabezal de la cinta está posicionado en la primera celda de la cinta y la pila está vacía.

Suponga un PDA con su conjunto de estados S, un alfabeto de entrada Σ y un alfabeto de la pila Γ. Cada transición del PDA será de la forma:

$$\Delta(s_1, a, b) = (s_2, z) \qquad (s_2, z) \in \Delta(s_1, a, b) \text{ para un PDA no determinista}$$

donde $s_1, s_2 \in S;\ a \in \Sigma \cup \{\varepsilon\},\ b \in \Gamma \cup \{\varepsilon\}$ y $z \in \Gamma^*$

Tal transición se dirá una transición de s_1 a s_2 sobre a con b/z que es representada en un diagrama de estados de la siguiente forma:

$$s_1 \xrightarrow{a, b/z} s_2$$

Si $a \neq \varepsilon$, $b \neq \varepsilon$, $z \neq \varepsilon$

Si estuviéramos posicionados en el estado S_1, si el próximo símbolo de entrada fuera a y el símbolo en la cima de la pila fuera b, hay una transición al estado s_2, b se desapila y z se apila (el símbolo más a la izquierda de z en la cima).

Si $a = \varepsilon$ no se consume ningún símbolo de entrada y la transición se dirá una transición ε.

Si $b = \varepsilon$ la transición se da sin consultar a la pila y no se desapila nada.

Si $z = \varepsilon$ no se apila nada.

1.1 Autómata PushDown Determinista

Un autómata pushDown Determinista (PDA D) es una séptupla $M = (S, \Sigma, \Gamma, \Delta, s_0, \gamma_0, F)$ donde:

S es un conjunto finito de estados.

Σ es el alfabeto de símbolos de entrada

Γ es el alfabeto de la pila

$s_0 \in S$ es el estado inicial

γ_0 es el símbolo inicial de la pila, $\gamma_0 \in \Gamma$

F es el conjunto de estados finales. $\emptyset \neq F \subseteq S$

Δ es la función de transición

$$\Delta : S \times (\Sigma \cup \varepsilon) \times \Gamma \to (S \times \Gamma^*)$$

El APFD procesa cadenas sobre una cinta de entrada semiinfinita, además hay una cinta llamada pila, que se utiliza como lugar de almacenamiento.

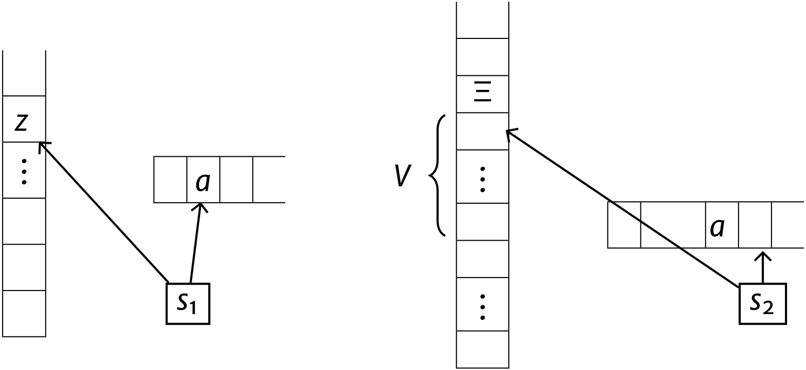
1.2 Paso Computacional

La transición

$$\Delta(s_1, a, z) = (s_2, v)$$

representa un paso computacional.

La unidad de control pasa del estado s_1 al estado s_2 y se mueve a la derecha; además borra el símbolo z que está en la cima de la pila, escribe la cadena v ($v \in \Gamma^*$) y pasa a escanear la nueva cima de la pila. Se puede representar gráficamente como sigue:



Un paso computacional

1.3 Casos Especiales de Transiciones

1. $\Delta(s_1, a, z) = (s_2, z)$. El contenido de la pila no se altera.

2. $\Delta(s_1, a, z) = (s_2, \varepsilon)$

   Se borra el símbolo z de la cima de la pila, y el control pasa a leer la nueva cima de la pila

3. $\Delta(s_1, \varepsilon, z) = (s_2, w)$

   Esta es una transición ε. No se procesa el símbolo sobre la cinta de entrada, la unidad de control no se mueve a la derecha, pero la cima de la pila z es reemplazada por la cadena w.

   Para garantizar el determinismo, $\Delta(s, a, z)$ y $\Delta(s, \varepsilon, z)$ con $a \in \Sigma$ no pueden estar simultáneamente definidos (sino sería no determinista).

La transición ε en un PDA D hacen que el autómata cambie el contenido de la pila sin consumir símbolos en la cinta de entrada.

1.4 Configuración Instantánea (C.I.)

Es una terna de la forma: (s, au, zv)

Representa lo siguiente:

El autómata está en el estado s, au es la parte no procesada de la cadena de entrada y la unidad de control está leyendo el símbolo a.

La cadena zv es el contenido total de la pila. z es el símbolo colocado en la cima.

Para representar el paso computacional escribiremos:

$$(s_1, au, zw) \longmapsto (s_2, u, vw)$$

El autómata utilizó la transición $\Delta(s_1, a, z) = (s_2, v)$

La notación

$$(s_1, u, \beta) \overset{*}{\longmapsto} (s_2, v, \gamma)$$

significa que el autómata pasa de la C.I (s_1, u, β) a la CI (s_2, v, γ) en cero, uno o más pasos computacionales

1.5 Configuración Inicial

Para una cadena de entrada $w \in \Sigma^*$, la configuración inicial es (s_0, w, γ_0)

El contenido de la pila es γ_0 al iniciar el procesamiento de la cadena de entrada.

1.6 Configuración de Aceptación

La configuración $(s_a, \varepsilon, \beta)$ se llama configuración de aceptación si $s_a \in F$.

Para que una cadena de entrada sea aceptada debe ser procesada completamente y la unidad de control debe apuntar a un estado de aceptación.

La cadena β que queda en la pila puede ser cualquier cadena de símbolos en Γ^*

1.7 Lenguaje aceptado por un PDA D

Definimos el lenguaje aceptado por PDA D M mediante:

$$L(M) = \{w \in \Sigma^* / (s_0, w, \gamma_0) \overset{*}{\longmapsto} (s_a, \varepsilon, \beta),\ s_a \in F\}$$

Una cadena es aceptada si pasa de la configuración inicial a una configuración de aceptación en cero, uno o más pasos.

- En el modelo PDA se permite que la transición $\Delta(s, a, z)$ no esté definida para algunos valores $s \in S, a \in \Sigma, z \in \Gamma$. En algunos casos el cómputo de algunas cadenas puede abortarse sin que se procesen completamente.
- No debe confundirse la terna que aparece en $\Delta(s, a, z)$ con la terna (s, u, β) que representa una configuración instantánea.
- La función de transición Δ requiere por lo menos que haya un símbolo en la pila. No hay cómputos con pila vacía.

Los analizadores sintácticos en compiladores se comportan generalmente como PDA.

Un PDA–D puede simular un AFD simplemente ignorando la pila, luego los lenguajes regulares son aceptados por PDA.

Teorema: Todo lenguaje regular L es aceptado por algún PDA–D

Demostración

Sea $M_1 = (S, \Sigma, \delta, F, s_0)$ un AFD que acepta a L.

El PDA–D $M_2 = (S, \Sigma, \Gamma, \Delta, s_0, \gamma_0, F)$ definido haciendo:

$$\Gamma = \{\gamma_0\} \text{ y}$$
$$\Delta(s, a, \gamma_0) = (\delta(s, a), \gamma_0) \quad \forall a \in \Sigma,\ s \in S$$

satisface claramente $L(M_2) = L(M_1) = L$

Ejemplo: Diseñar un PDA–D que acepte el lenguaje $L = \{a^i b^i / i \geq 1\}$ sobre el alfabeto $\Sigma = \{a, b\}$

Recordar que L no es regular y no puede ser aceptado por ningún autómata sin pila.

Solución

Se buscará copiar los símbolos a en la pila y borrar una a por cada b que sea leída en la cinta. Se aceptará una cadena si es procesada completamente y en la pila sólo queda el marcador γ_0.

Sea $M = (S, \Sigma, \Gamma, \Delta, s_0, \gamma_0, F)$ donde

$\Sigma = \{a, b\}$

$\Gamma = \{\gamma_0, A, B\}$

$S = \{s_0, s_1, s_2\}$

$F = \{s_2\}$

y la función de transición Δ queda definida por:

$$\begin{gathered}
\Delta(s_0, a, \gamma_0) = (s_0, A\gamma_0) \\
\Delta(s_0, a, A) = (s_0, AA) \\
\Delta(s_0, b, A) = (s_1, \varepsilon) \\
\Delta(s_0, b, A) = (s_1, \varepsilon) \\
\Delta(s_0, \varepsilon, \gamma_0) = (s_2, \gamma_0)
\end{gathered}$$

Veamos el procesamiento de algunas cadenas de entrada

1. Sea $w = aaabbb$

$$\begin{aligned}
&(s_0, aaabbb, \gamma_0) \longmapsto (s_0, aabbb, A\gamma_0) \longmapsto (s_0, abbb, AA\gamma_0) \\
&\vdash (s_0, bbb, AAA\gamma_0) \longmapsto (s_1, bb, AA\gamma_0) \longmapsto (s_1, b, A\gamma_0) \\
&\vdash (s_1, \varepsilon, \gamma_0) \longmapsto (s_2, \varepsilon, \gamma_0)
\end{aligned}$$

la última es una configuración de aceptación por lo que w es aceptada.

2. Sea $w = aabbb$. Se tiene:

$$(s_0, aabbb, \gamma_0) \longmapsto (s_0, abbb, A\gamma_0) \longmapsto (s_0, bbb, AA\gamma_0)$$
$$\vdash (s_1, bb, A\gamma_0) \longmapsto (s_1, b, \gamma_0) \longmapsto (s_2, b, \gamma_0)$$

¡cómputo abortado!

la cadena w no es aceptada, porque a pesar de estar en el estado final s_2 que es de aceptación, sin embargo w no se ha procesado completamente.

3. Sea $w = aaabb$. Se tiene:

$$(s_0, aaabb, \gamma_0) \vdash (s_0, aabb, A\gamma_0) \vdash (s_0, abb, AA\gamma_0)$$

$$\vdash (s_0, bb, AAA\gamma_0) \vdash (s_1, b, AA\gamma_0) \vdash (s_1, \varepsilon, A\gamma_0)$$

La configuración $(s_1, \varepsilon, A\gamma_0)$ no es de aceptación, porque a pesar de haber consumido toda la cadena w, el estado final s_1 no es un estado de aceptación.

Luego, $w = aaabb$ no es aceptada por el PDA–D.

Ejemplo: Diseñar un PDA–D que acepte el lenguaje de todas las cadenas sobre $\Sigma = \{a, b\}$ diferentes de ε que tienen igual cantidad de símbolos a y b.

Solución

El objetivo será acumular las a's o b's consecutivas en la pila

- Si en la cima hay una A y se lee una b, se borrará la A.
- Si en la cima hay una B y se lee una a, se borrará la B.

Se aceptará una cadena si es procesada completamente y en la pila sólo queda el marcador de fondo γ_0.

Sea $M = (S, \Sigma, \Gamma, \Delta, s_0, \gamma_0, F)$ donde:

$$\begin{aligned} S &= \{s_0, s_1, s_2\} \\ \Sigma &= \{a, b\} \\ \Gamma &= \{\gamma_0, A, B\} \end{aligned}$$

$$F = \{s_2\}$$

y la función de transición está dada por:

$$\Delta(s_0, a, \gamma_0) = (s_1, A\gamma_0)$$

$$\Delta(s_0, b, \gamma_0) = (s_1, B\gamma_0)$$

$$\Delta(s_1, a, \gamma_0) = (s_1, A\gamma_0)$$

$$\Delta(s_1, b, \gamma_0) = (s_1, B\gamma_0)$$

$$\Delta(s_1, a, A) = (s_1, AA)$$

$$\Delta(s_1, b, B) = (s_1, BB)$$

$$\Delta(s_1, a, B) = (s_1, \varepsilon)$$

$$\Delta(s_1, b, A) = (s_1, \varepsilon)$$

$$\Delta(s_1, \varepsilon, \gamma_0) = (s_2, \gamma_0)$$

Procesamos algunas cadenas:

1. $w = aababab b$

$$
\begin{aligned}
(s_0, aabababb, \gamma_0) &\vdash (s_1, abababb, A\gamma_0) \longmapsto (s_1, bababb, AA\gamma_0) \\
&\vdash (s_1, ababb, A\gamma_0) \longmapsto (s_1, babb, AA\gamma_0) \longmapsto (s_1, abb, A\gamma_0) \\
&\vdash (s_1, bb, AA\gamma_0) \longmapsto (s_1, b, A\gamma_0) \longmapsto (s_1, \varepsilon, \gamma_0) \vdash (s_2, \varepsilon, \gamma_0)
\end{aligned}
$$

La configuración $(s_2, \varepsilon, \gamma_0)$ es de aceptación por lo que la cadena $w = aabababb$ es aceptada.

2. $w = bbbaba$

$$\begin{aligned}(s_0, bbbaba, \gamma_0) &\vdash (s_1, bbaba, B\gamma_0) \vdash (s_1, baba, BB\gamma_0) \vdash (s_1, aba, BBB\gamma_0)\\ &\vdash (s_1, ba, BB\gamma_0) \vdash (s_1, a, BBB\gamma_0) \vdash (s_1, \varepsilon, BB\gamma_0)\end{aligned}$$

la cadena w se procesó completamente pero la configuración final no es de aceptación. Luego w es rechazada.

Ejemplo: Diseñar un PDA D que acepte el lenguaje

$$L = \{wcw^R / w \in \{a, b\}^*\}$$

sobre $\Sigma = \{a, b, c\}$.
Observar que las cadenas w y w^R sólo poseen a's y b's

Solución

El objetivo es acumular los símbolos en la pila hasta que aparezca la c.

Luego debemos comparar los símbolos leídos con los almacenados en la pila, removiendo en cada paso la cima de la pila.

Se aceptará una cadena si es procesada por completo y en la pila sólo queda el marcador γ_0

Sea $M = (S, \Sigma, \Gamma, \Delta, s_0, \gamma_0, F)$

donde

$$\begin{aligned} S &= \{s_0, s_1, s_2\} \\ \Sigma &= \{a, b, c\} \\ \Gamma &= \{\gamma_0, A, B\} \\ F &= \{s_2\} \end{aligned}$$

y la función de transición está dada por:

$$\Delta(s_0, a, \gamma_0) = (s_0, A\gamma_0)$$
$$\Delta(s_0, b, \gamma_0) = (s_0, B\gamma_0)$$
$$\Delta(s_0, c, \gamma_0) = (s_2, \gamma_0)$$
$$\Delta(s_0, a, A) = (s_0, AA)$$

$$\Delta(s_0, a, B) = (s_0, AB)$$
$$\Delta(s_0, b, A) = (s_0, BA)$$
$$\Delta(s_0, b, B) = (s_0, BB)$$
$$\Delta(s_0, c, A) = (s_1, A)$$
$$\Delta(s_0, c, B) = (s_1, B)$$
$$\Delta(s_1, a, A) = (s_1, \varepsilon)$$
$$\Delta(s_1, b, B) = (s_1, \varepsilon)$$
$$\Delta(s_1, \varepsilon, \gamma_0) = (s_2, \gamma_0)$$