Teoría de la Computación

Víctor Melchor

Capítulo 1

Autómatas Push Down No Deterministas

Un autómata Push Down No Determinístico acepta a los Lenguajes Libres de Contexto.

Sabemos que los lenguajes regulares permiten describir identificadores, palabras claves y patrones de uso común. Sin embargo, en computación necesitamos un modelo más poderoso para describir las estructuras sintácticas de los lenguajes de programación.

Las gramáticas Libres de Contexto (GLC) y los lenguajes que generan (Lenguajes Libres de Contexto) permiten definir la sintaxis de los lenguajes de programación.
Los modelos mecánicos que corresponden a las GLC son los Autómatas de Pila que son como los AFD pero tienen adicionalmente una pila para almacenamiento. En esta estructura la información se registra en forma LIFO (último en entrar, primero en salir).

Definición: Un autómata Push Down No Determinístico (PDA–ND) es una séptupla

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

donde:

- S, Σ, Γ y F son como en los PDA–D
- La función de transición Δ es de la forma:

  $$\Delta : S \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to \mathcal{P}(S \times \Gamma^*)$$

  $\mathcal{P}(S \times \Gamma^*)$ está conformado por los subconjuntos finitos de $S \times \Gamma^*$.

Para $s \in S$, $a \in \Sigma \cup \{\varepsilon\}$ y $\gamma \in \Gamma$, $\Delta(s, a, \gamma)$ es de la forma:

$$\Delta(s, a, \gamma) = \{(s_1, \gamma_1), (s_2, \gamma_2), \ldots, (s_k, \gamma_k)\}$$

lo cual significa:

Al leer el símbolo a en la cinta de entrada, la unidad de control pasa aleatoriamente a uno de los estados s_i y se mueve a la derecha.

En la pila, se desapila γ y escribe la cadena γ_i ($\gamma_i \in \Gamma^*$)

A diferencia de lo que sucede con los PDA-D, en el modelo PDA-ND las transiciones ε, $\Delta(s, \varepsilon, \gamma)$ no tienen ninguna restricción.

Definición: Sea un PDA–ND $M = (S, \Sigma, \Gamma, \Delta, s_0, \gamma_0, F)$. El lenguaje aceptado por M es:

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

w será aceptada si existe por lo menos un procesamiento de w desde la configuración inicial hasta una configuración de aceptación.

Ejemplo: Diseñar un PDA–ND que acepte el lenguaje $\{a^i b^i / i \geq 0\}$ sobre el alfabeto Σ.

Solución

$L = \{a^i b^i / i \geq 0\}$ incluye a la cadena ε ($i = 0$)

Añadiremos una transición más al PDA-D visto en la sesión anterior.

Definimos el PDA-ND $M = (S, \Sigma, \Gamma, \Delta, s_0, \gamma_0, F)$ donde:

$$S = \{s_0, s_1, s_2\}$$
$$\Sigma = \{a, b\}$$
$$\Gamma = \{\gamma_0, A\}$$
$$F = \{s_2\}$$

La función de transición está dada por:

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

Aquí surge el no-determinismo por la presencia simultánea de $\Delta(s_0, a, \gamma_0)$ y $\Delta(s_0, \varepsilon, \gamma_0)$

Ejemplo: Diseñar un PDA-ND que acepte el lenguaje de las cadenas sobre $\Sigma = \{a, b\}$ con la misma cantidad de símbolos a y b

Solución

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

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

y Δ está dada por:

$$
\begin{aligned}
\Delta(s_0, a, \gamma_0) &= \{(s_0, A\gamma_0)\} \\
\Delta(s_0, b, \gamma_0) &= \{(s_0, B\gamma_0)\} \\
\Delta(s_0, a, A) &= \{(s_0, AA)\} \\
\Delta(s_0, b, B) &= \{(s_0, BB)\}
\end{aligned}
$$

$$\Delta(s_0, a, B) = \{(s_0, \varepsilon)\}$$
$$\Delta(s_0, b, A) = \{(s_0, \varepsilon)\}$$
$$\Delta(s_0, \varepsilon, \gamma_0) = \{(s_1, \gamma_0)\}$$

El no determinismo se da por la presencia simultánea de $\Delta(s_0, a, \gamma_0)$ y $\Delta(s_0, \varepsilon, \gamma_0)$.

Los modelos PDA–D y PDA–ND no son computacionalmente equivalentes. Existen lenguajes aceptados por PDA–ND que no pueden ser aceptados por ningún PDA–D.

Ejemplo:[Aceptación por Estado Final] Diseñar un PDA–ND que acepte el lenguaje

$$L = \{ww^R / w \in \Sigma^*\}; \quad \Sigma = \{a, b\}$$

Solución

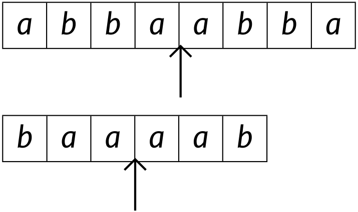
Cada vez que lleguen 2 símbolos iguales seguidos cabe la posibilidad que estemos en el centro de la cadena.

Se define el PDA-ND $M = (S, \Sigma, \Gamma, \Delta, s_0, \gamma_0, F)$ donde:

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

La función de transición está dada por:

$$
\begin{aligned}
\Delta(s_0, a, \gamma_0) &= \{(s_0, A\gamma_0)\} \\
\Delta(s_0, b, \gamma_0) &= \{(s_0, B\gamma_0)\} \\
\Delta(s_0, \varepsilon, \gamma_0) &= \{(s_2, \gamma_0)\} \\
\Delta(s_0, a, A) &= \{(s_0, AA), (s_1, \varepsilon)\} \\
\Delta(s_0, a, B) &= \{(s_0, AB)\} \\
\Delta(s_0, b, A) &= \{(s_0, BA)\} \\
\Delta(s_0, b, B) &= \{(s_0, BB), (s_1, \varepsilon)\} \\
\Delta(s_1, a, A) &= \{(s_1, \varepsilon)\} \\
\Delta(s_1, b, B) &= \{(s_1, \varepsilon)\} \\
\Delta(s_1, \varepsilon, \gamma_0) &= \{(s_2, \gamma_0)\}
\end{aligned}
$$



Las dos transiciones:

$$\Delta(s_0, a, A) = \{(s_0, AA), (s_1, \varepsilon)\}$$
$$\Delta(s_0, b, B) = \{(s_0, BB), (s_1, \varepsilon)\}$$

le permiten al autómata un comportamiento no determinista:

o sigue acumulando símbolos en la pila en el estado s_0;

o supone que se ha llegado a la mitad de la cadena de entrada y pasa al estado s_1 y empieza a desapilar los símbolos ya almacenados en la pila.

1.1 Lenguaje Aceptado por un Autómata Push Down

Existen dos enfoques que permiten definir el lenguaje de un PDA.

Definición:[Aceptación por Estado Final]
Sea $M = (S, \Sigma, \Gamma, \Delta, s_0, \gamma_0, F)$ un PDA. Entonces el lenguaje aceptado por M por estado final es:

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

El contenido β en la pila es irrelevante

Definición:[Aceptación por Pila Vacía]
Dado un PDA $M = (S, \Sigma, \Gamma, \Delta, s_0, \varepsilon_0, F)$. Definimos el lenguaje aceptado por M por pila vacía como sigue:

$$N(M) = \{w \in \Sigma^*/(s_0, w, \gamma_0) \overset{*}{\longmapsto} (s, \varepsilon, \varepsilon);\ s \in S\}$$

En este caso el estado s es irrelevante. Podemos considerar $F = \emptyset$

Ejemplo: Diseñar un automata de Pila Vacía que acepte el lenguaje $L = \{a^i b^i / i \geq 0\}$

Solución Definimos el PDA–ND $M = (S, \Sigma, \Gamma, \Delta, s_0, \gamma_0, F)$ donde

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

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

Evalúe si el PDA acepta las cadenas $w_1 = \varepsilon$ y $w_2 = aabb$.

1. Para $w_1 = \varepsilon$

$$(s_0, \varepsilon, \gamma_0) \longmapsto (s_0, \varepsilon, \varepsilon)$$

   se acepta w_1 por Pila Vacía.

2. Para $w_2 = aabb$

$$\begin{aligned}(s_0, aabb, \gamma_0) &\vdash (s_0, abb, A\gamma_0) \vdash (s_0, bb, AA\gamma_0) \\ &\longmapsto (s_1, b, A\gamma_0) \longmapsto (s_1, \varepsilon, \gamma_0) \longmapsto (s_1, \varepsilon, \varepsilon)\end{aligned}$$

   se acepta w_2 por Pila Vacía.

Ejemplo: Diseñar un PDA–ND M que acepte el lenguaje

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

Solución Se presentan algunos casos como cadena de entrada:

- Si $w = ababcbaba \rightarrow w \in L$
- Si $w = abcab \rightarrow w \notin L$
- Si $w = cbc \rightarrow w \notin L$

Definimos el PDA $M = (S, \Sigma, \Gamma, \Delta, s_0, F)$ donde:

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

la función de transición está dada por:

$$\Delta(s_0, a, \gamma_0) = \{(s_0, A\gamma_0)\} \tag{1.1}$$

$$\Delta(s_0, b, \gamma_0) = \{(s_0, B\gamma_0)\} \tag{1.2}$$

$$\Delta(s_0, \varepsilon, \gamma_0) = \{(s_2, \gamma_0)\} \tag{1.3}$$

$$\Delta(s_0, a, A) = \{(s_0, AA), (s_1, \varepsilon)\} \tag{1.4}$$

$$\Delta(s_0, a, B) = \{(s_0, AB)\} \tag{1.5}$$

$$\Delta(s_0, b, A) = \{(s_0, BA)\} \tag{1.6}$$

$$\Delta(s_0, b, B) = \{(s_0, BB), (s_1, \varepsilon)\} \tag{1.7}$$

$$\Delta(s_1, a, A) = \{(s_1, \varepsilon)\} \tag{1.8}$$

$$\Delta(s_1, b, B) = \{(s_1, \varepsilon)\} \tag{1.9}$$

$$\Delta(s_1, \varepsilon, \gamma_0) = \{(s_2, \gamma_0)\} \tag{1.10}$$

$$\tag{1.11}$$

A continuación se presenta la secuencia de transiciones para $w = abbcbba$

Transición Usada	Estado	Entrada no Leída	Pila
–	s_0	$abbcbba$	γ_0
1	s_0	$bbcbba$	$A\gamma_0$
2	s_0	$bcbba$	$BA\gamma_0$
2	s_0	$cbba$	$BBA\gamma_0$
3	s_1	bba	$BBA\gamma_0$
5	s_1	ba	$BA\gamma_0$
5	s_1	a	$A\gamma_0$
4	s_1	ε	γ_0

Como la CF $(s_1, \varepsilon, \gamma_0)$ es una CA se acepta $w = abbcbba$

1.2 Lema de Bombeo para Lenguajes no Regulares

Sea L un lenguaje regular. Entonces existe una constante n (que depende de L) tal que para toda cadena $w \in L$ con $|w| \geq n$, podemos descomponer w en tres cadenas, $w = xyz$ tal que satisfacen:

1. $y \neq \varepsilon$
2. $|xy| \leq n$
3. xy^kz también pertenece a $L \quad \forall k \geq 0$

Siempre podemos hallar una cadena no vacía y no demasiado alejada del principio de w que puede "bombearse"

Si se repite y cualquier número de veces o se borra (cuando $k = 0$) la cadena resultante también pertenece a L

Ejemplo: El lenguaje $L = \{a^i b^i / i \geq 0\}$ no es regular.

Si lo fuese, según el Lema sería aplicable para algún entero n.

Consideremos la cadena $w = a^n b^n \in L$.

Según el teorema, podemos reescribir $w = xyz$ tal que $|xy| \leq n$ y $y \neq \varepsilon$, i.e. $y = a^i$ para algún $i > 0$

Como $w = a^n b^n = xa^i z$ entonces $xz = a^{n-i} b^n \notin L$ lo cual contradice el teorema.

Ejemplo: Sea el lenguaje $L = \{0^n 1^n / n \geq 0\}$

Probaremos que L no es regular usando el Lema de Bombeo.

Por contradicción.

Supongamos que L es regular. Sea n la longitud de bombeo.

Elegimos $w = 0^p 1^p$

Como $w \in L$ y $|w| \geq p$, el LB garantiza que $w = xyz$, donde $\forall i \geq 0$ $xy^i z \in L$

Consideraremos 3 casos que verifican que este resultado es imposible.

1. y consiste solo de ceros. Luego la cadena $xyyz$ tiene más ceros que unos luego no es miembro de L ($\rightarrow\leftarrow$ condición iii)
2. y consiste solo de unos. Esto también lleva a una contradicción.
3. y consiste de 0s y 1s

   La cadena $xyyz$ puede tener el mismo cantidad de 0s y 1s pero podría salir del orden con algún 1s de 0s, luego no será miembro de L