Teoría de la Computación

Clase 14: Autómatas de pila

Mauro Artigiani

10 septiembre 2021

Universidad del Rosario, Bogotá

Un modelo de computación para `CFL`

Descripción informal

Después de haber introducido las `CFLs`, vamos a ver un modelo computacional nuevo: los autómatas de pila En las próximas clases mostraremos que un lenguaje es independiente del contexto si y solo si es reconocido por un autómata de pila.

Descripción informal

Después de haber introducido las `CFLs`, vamos a ver un modelo computacional nuevo: los autómatas de pila En las próximas clases mostraremos que un lenguaje es independiente del contexto si y solo si es reconocido por un autómata de pila.

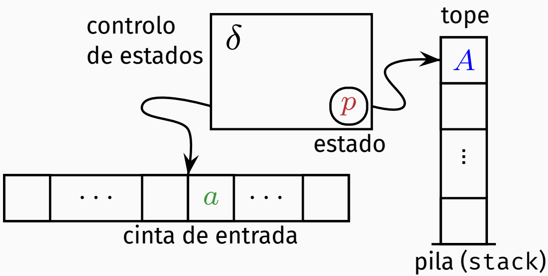
Antes de introducir estos autómatas, es útil dar una descripción a grandes rasgos de como sirve un autómata de estados finitos. Podemos ver un autómata de estados finitos como un máquina que lee una cinta (en donde está escrito la palabra de input). La cabeza de la maquina lee el símbolo en la cinta, cambia estado y avanza su lectura de la cinta.

Autómatas de pila

Un autómata de pila (*pushdown automaton* en inglés), o `PDA` se comporta como un autómata de estados finitos, pero tiene una componente más: la pila (o *stack*).

Autómatas de pila

Un autómata de pila (*pushdown automaton* en inglés), o `PDA` se comporta como un autómata de estados finitos, pero tiene una componente más: la pila (o *stack*).



La pila es una memoria infinita. El autómata solo tiene acceso al tope de la pila, que puede leer, borrar y escribir, "empujando" más abajo los demás elementos de la pila.

Autómatas de pila

Hemos visto que el lenguaje

$$\{0^n1^n, n \geq 0\},$$

es un lenguaje independiente del contexto (pero no es regular).

Autómatas de pila

Hemos visto que el lenguaje

$$\{0^n 1^n, n \geq 0\},$$

es un lenguaje independiente del contexto (pero no es regular).

Damos una descripción del porque un autómata de pila podría reconocer este lenguaje.

Autómatas de pila

Hemos visto que el lenguaje

$$\{0^n 1^n, n \geq 0\},$$

es un lenguaje independiente del contexto (pero no es regular).

Damos una descripción del porque un autómata de pila podría reconocer este lenguaje.

Empezamos leyendo el input. Si leemos un 0 lo escribimos también en la pila.

Autómatas de pila

Hemos visto que el lenguaje

$$\{0^n 1^n, n \geq 0\},$$

es un lenguaje independiente del contexto (pero no es regular).

Damos una descripción del porque un autómata de pila podría reconocer este lenguaje.

Empezamos leyendo el input. Si leemos un 0 lo escribimos también en la pila. Lo más pronto vemos un 1 empezamos a borrar 0s de la pila.

Autómatas de pila

Hemos visto que el lenguaje

$$\{0^n 1^n, n \geq 0\},$$

es un lenguaje independiente del contexto (pero no es regular).

Damos una descripción del porque un autómata de pila podría reconocer este lenguaje.

Empezamos leyendo el input. Si leemos un 0 lo escribimos también en la pila. Lo más pronto vemos un 1 empezamos a borrar 0s de la pila. Si acabamos de leer el input en el momento en que la pila se vuelve vacía, aceptamos la palabra.

Autómatas de pila

Hemos visto que el lenguaje

$$\{0^n 1^n, n \geq 0\},$$

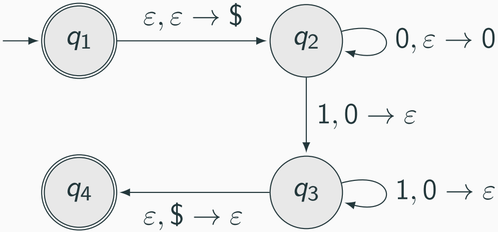
es un lenguaje independiente del contexto (pero no es regular).

Damos una descripción del porque un autómata de pila podría reconocer este lenguaje.

Empezamos leyendo el input. Si leemos un 0 lo escribimos también en la pila. Lo más pronto vemos un 1 empezamos a borrar 0s de la pila. Si acabamos de leer el input en el momento en que la pila se vuelve vacía, aceptamos la palabra. Si la pila se vacía antes de terminar de leer el input, rechazamos la palabra. De manera análoga, si la palabra acaba antes de que la pila sea vacía, rechazamos la palabra.

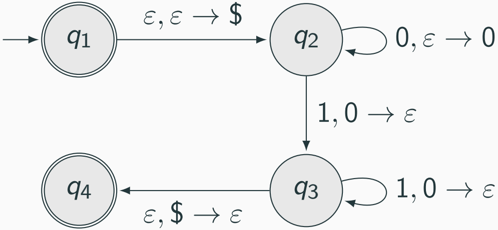
Ejemplo: diagrama de estados

Podemos describir un `PDA` también con un diagrama de estados. Ahora en cara arista debemos también decir como el autómata modifica la pila. Escribimos $a, b \to c$ si, leyendo el símbolo a, el `PDA` remplaza el símbolo b en el tope de la pila con el símbolo c.



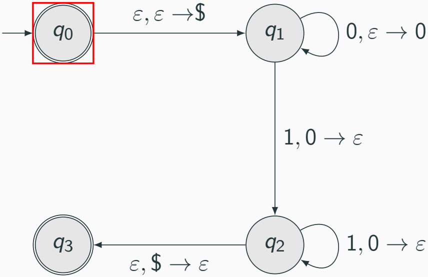
Ejemplo: diagrama de estados

Podemos describir un PDA también con un diagrama de estados. Ahora en cara arista debemos también decir como el autómata modifica la pila. Escribimos $a, b \to c$ si, leyendo el símbolo a, el PDA remplaza el símbolo b en el tope de la pila con el símbolo c.

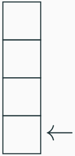


Nótese que para hacer entender al autómata que el input se ha acabado, hemos diseñado el autómata en manera que los estados finales persisten solo si la palabra de input se ha acabado.

Procesamiento en un ejemplo



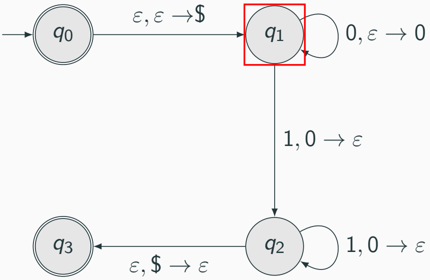
Pila



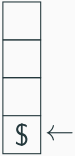
Cinta de entrada

Procesando

Procesamiento en un ejemplo



Pila

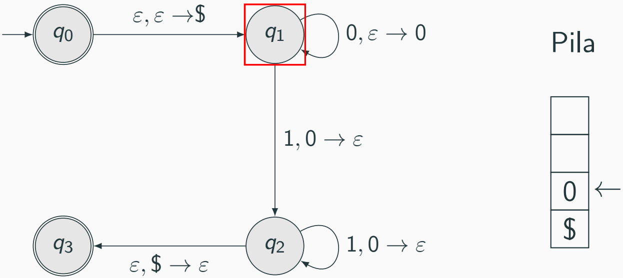


Cinta de entrada

Procesando

$q_0 \xrightarrow{\varepsilon,\varepsilon\to\$} q_1$

Procesamiento en un ejemplo



Cinta de entrada

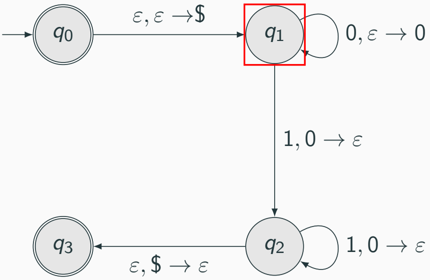
0	0	1	1	$\equiv$

$\uparrow$

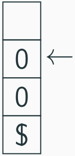
Procesando

$q_1 \xrightarrow{0,\varepsilon\to 0} q_1$

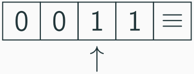
Procesamiento en un ejemplo



Pila



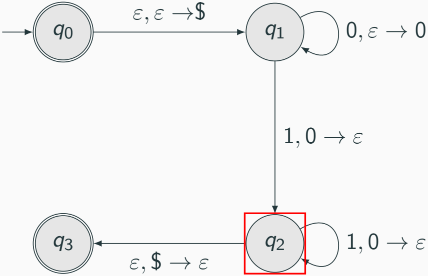
Cinta de entrada



Procesando

$q_1 \xrightarrow{0,\varepsilon\to 0} q_1$

Procesamiento en un ejemplo



Pila

0	$\leftarrow$
\$	

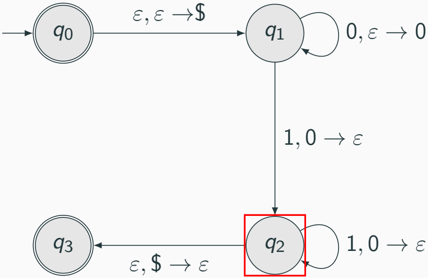
Cinta de entrada

0	0	1	1	$\equiv$
			$\uparrow$	

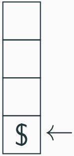
Procesando

$q_1 \xrightarrow{1,0 \to \varepsilon} q_2$

Procesamiento en un ejemplo



Pila

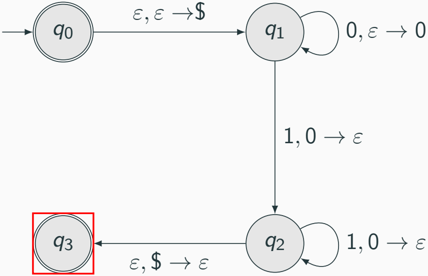


Cinta de entrada

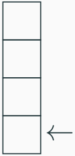
Procesando

$q_2 \xrightarrow{1,0\to\varepsilon} q_2$

Procesamiento en un ejemplo



Pila



Cinta de entrada

0	0	1	1	$\equiv$

$\uparrow$

Procesando

$$q_2 \xrightarrow{\varepsilon,\$\to\varepsilon} q_3$$

Definición de autómata de pila

Definición de autómata de pila

Hay autómatas de pilas deterministas y no deterministas. En este caso pero, los dos no tienen el mismo poder computacional. Por eso, nosotros solo hablaremos de autómatas de pilas no deterministas.

Definición de autómata de pila

Hay autómatas de pilas deterministas y no deterministas. En este caso pero, los dos no tienen el mismo poder computacional. Por eso, nosotros solo hablaremos de autómatas de pilas no deterministas.

Un PDA es muy similar a un NFA, con la adición de la pila. Para describir la pila necesitamos de otro alfabeto, que denotamos con Γ.

Definición de autómata de pila

Hay autómatas de pilas deterministas y no deterministas. En este caso pero, los dos no tienen el mismo poder computacional. Por eso, nosotros solo hablaremos de autómatas de pilas no deterministas.

Un PDA es muy similar a un NFA, con la adición de la pila. Para describir la pila necesitamos de otro alfabeto, que denotamos con Γ. En cada momento, el autómata se encuentra en un estado, lee el símbolo de input y el símbolo en el tope de la pila. Cada uno de los símbolos puede ser ε y entonces el autómata puede moverse de manera espontanea.

Definición de autómata de pila

Hay autómatas de pilas deterministas y no deterministas. En este caso pero, los dos no tienen el mismo poder computacional. Por eso, nosotros solo hablaremos de autómatas de pilas no deterministas.

Un `PDA` es muy similar a un `NFA`, con la adición de la pila. Para describir la pila necesitamos de otro alfabeto, que denotamos con Γ. En cada momento, el autómata se encuentra en un estado, lee el símbolo de input y el símbolo en el tope de la pila. Cada uno de los símbolos puede ser ε y entonces el autómata puede moverse de manera espontanea. Al computar el autómata cambia estado y modifica la pila. Dado que nuestro `PDA` es no determinista la función de transición será definida de la siguiente manera

$$\delta\colon Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \to \mathcal{P}(Q \times \Gamma_\varepsilon)$$

Definición de autómata de pila

Definición

Un autómata de pila (no determinista) es una 6-tupla $(Q, \Sigma, \Gamma, \delta, q_0, F)$, donde:

- Q es el conjunto finito de los estados;
- Σ es el alfabeto de input (finito);
- Γ es el alfabeto de la pila (finito);
- $\delta\colon Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \to \mathcal{P}(Q \times \Gamma_\varepsilon)$ es la función de transición;
- $q_0 \in Q$ es el estado inicial;
- $F \subseteq Q$ es el conjunto de los estados de aceptación.

Ejemplo: definición

Describimos ahora en manera formal el `PDA` que reconoce el lenguaje $\{0^n1^n, n \geq 0\}$.

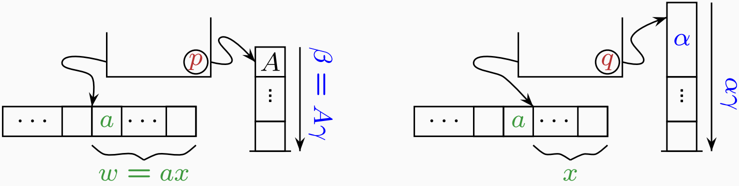
Ejemplo: definición

Describimos ahora en manera formal el `PDA` que reconoce el lenguaje $\{0^n1^n, n \geq 0\}$. Sean $Q = \{q_1, q_2, q_3, q_4\}$, $\Sigma = \{0, 1\}$, $\Gamma = \{0, \$\}$, $F = \{q_1, q_4\}$ y δ es como sigue (los símbolos que faltan son $\emptyset$):

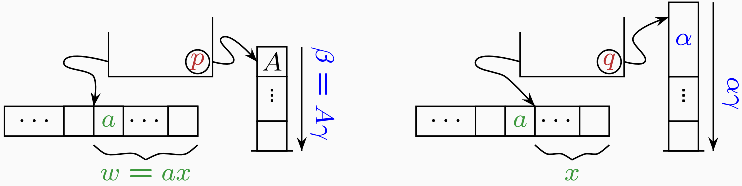
Input:	0			1			ε		
Stack:	0	\$	ε	0	\$	ε	0	\$	ε
q_1									$\{(q_2, \$)\}$
q_2			$\{(q_2, 0)\}$	$\{(q_3, \$)\}$					
q_3				$\{(q_3, \$)\}$				$\{(q_4, \varepsilon)\}$	
q_4									

El símbolo \$ es un símbolo especial que utilizamos señalar el comienzo de la pila: cuando lo leemos sabemos que la pila está vacía.

Procesamiento en general



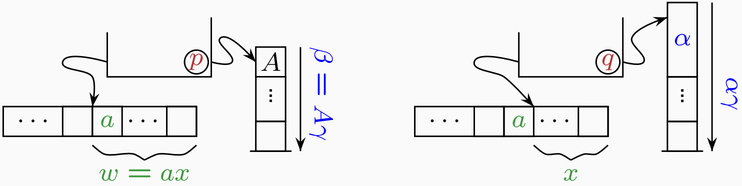
Procesamiento en general



Un PDA acepta una palabra $w = w_1 w_2 \ldots w_m$, con $w_i \in \Sigma_\varepsilon$ si existe una secuencia de estados $r_0, r_1, \ldots, r_m \in Q$ y de palabras $s_0, s_1, \ldots, s_m \in \Gamma^*$ tales que:

1. $r_0 = q_0$ y $s_0 = \varepsilon$;

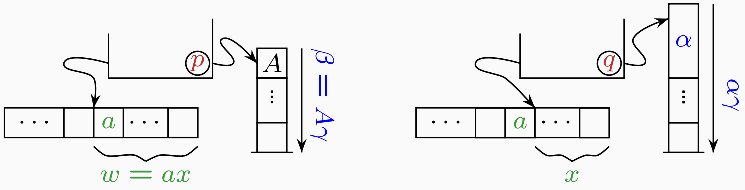
Procesamiento en general



Un PDA acepta una palabra $w = w_1 w_2 \dots w_m$, con $w_i \in \Sigma_\varepsilon$ si existe una secuencia de estados $r_0, r_1, \dots, r_m \in Q$ y de palabras $s_0, s_1, \dots, s_m \in \Gamma^*$ tales que:

1. $r_0 = q_0$ y $s_0 = \varepsilon$;
2. Para todos $i = 0, \dots, m-1$, se tiene $(r_{i+1}, b) \in \delta(r_i, w_{i+1}, a)$ donde $s_i = at$ y $s_{i+1} = bt$, con $a, b \in \Gamma_\varepsilon$ y $t \in \Gamma^*$;

Procesamiento en general

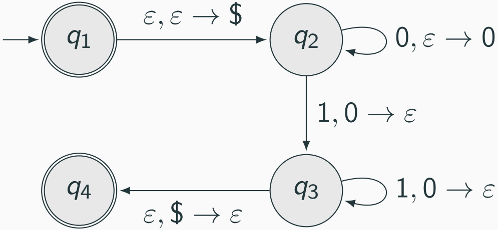


Un PDA acepta una palabra $w = w_1 w_2 \dots w_m$, con $w_i \in \Sigma_\varepsilon$ si existe una secuencia de estados $r_0, r_1, \dots, r_m \in Q$ y de palabras $s_0, s_1, \dots, s_m \in \Gamma^*$ tales que:

1. $r_0 = q_0$ y $s_0 = \varepsilon$;
2. Para todos $i = 0, \dots, m-1$, se tiene $(r_{i+1}, b) \in \delta(r_i, w_{i+1}, a)$ donde $s_i = at$ y $s_{i+1} = bt$, con $a, b \in \Gamma_\varepsilon$ y $t \in \Gamma^*$;
3. $r_m \in F$.

Más ejemplos

Ejemplo 2



Haciendo unas pequeñas modificaciones, se puede adaptar este PDA para reconocer el lenguaje

$$\{ww^{\mathcal{R}}, w \in \{0,1\}^*\},$$

donde $w^{\mathcal{R}}$ significa w escrita a revés.

Ejemplo 3

Hagamos un ejemplo más complicado. Consideramos el lenguaje

$$\{\mathtt{a}^i\mathtt{b}^k\mathtt{c}^l, i, k, l \geq 0 \text{ con } i = k \text{ o } i = l\}$$

Ejemplo 3

Hagamos un ejemplo más complicado. Consideramos el lenguaje

$$\{\mathtt{a}^i\mathtt{b}^k\mathtt{c}^l, i, k, l \geq 0 \text{ con } i = k \text{ o } i = l\}$$

La idea es empezar escribiendo todas las `a` que leemos en la pila. Después iremos quitandolas al momento de leer `b` o `c`. El problema es que no sabemos si comparar las `a` con las `b` o con la `c`.

Ejemplo 3

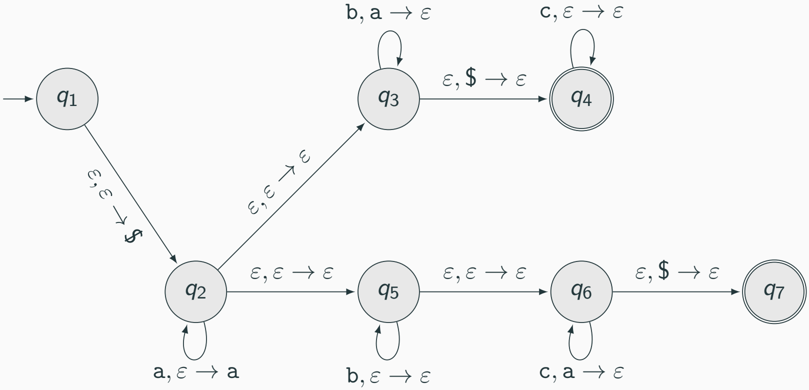
Hagamos un ejemplo más complicado. Consideramos el lenguaje

$$\{\mathtt{a}^i\mathtt{b}^k\mathtt{c}^l, i, k, l \geq 0 \text{ con } i = k \text{ o } i = l\}$$

La idea es empezar escribiendo todas las `a` que leemos en la pila. Después iremos quitandolas al momento de leer `b` o `c`. El problema es que no sabemos si comparar las `a` con las `b` o con la `c`.

Por esto es útil el no determinismo: ponemos una bifurcación espontanea cuando se acaben las `a`.

Ejemplo 3



q_1
$\varepsilon, \varepsilon \to \$$
q_2
$\mathtt{a}, \varepsilon \to \mathtt{a}$
$\varepsilon, \varepsilon \to \varepsilon$
q_3
$\mathtt{b}, \mathtt{a} \to \varepsilon$
$\varepsilon, \$ \to \varepsilon$
q_4
$\mathtt{c}, \varepsilon \to \varepsilon$
$\varepsilon, \varepsilon \to \varepsilon$
q_5
$\mathtt{b}, \varepsilon \to \varepsilon$
$\varepsilon, \varepsilon \to \varepsilon$
q_6
$\mathtt{c}, \mathtt{a} \to \varepsilon$
$\varepsilon, \$ \to \varepsilon$
q_7

Resumen

Resumen

Hoy aprendimos:

- Qué es un autómata de pila (o `PDA`);
- Cómo procesa un `PDA`;
- Cómo representar un `PDA` en un diagrama de estados.