Equivalencia Entre PDA y CFL

El Lenguaje aceptado por un Autómata con Pila

Universidad de Cantabria

Esquema

Lenguaje Aceptado por un Autómata

Como en los autómatas finitos, se puede definir en los autómatas con pila las palabras que son aceptadas por estos autómatas.

Lenguaje Aceptado por un Autómata

Esto es, una palabra es aceptada por un autómata si al terminar de leer la palabra por el autómata, se llega a un estado final.

Lenguaje Aceptado por un Autómata

Pero recordar también, que como el autómata lee también de la pila, el autómata se puede detener sin haber leído toda la palabra.

Lenguaje Aceptado por un Autómata

Hay dos maneras de interpretar el lenguaje aceptado por un autómata con pila: por estado final aceptador y por pila y cinta vacías.

Lenguaje Aceptado Mediante Estado Final

Definición (Lenguaje aceptado mediante estado final aceptador)

Sea $A := (Q, \Sigma, \Gamma, Q_0, Z_0, F, \delta)$ *un autómata con pila y sea* $\mathcal{S}_A$ *el sistema de transición asociado. Para cada palabra* $\omega \in \Sigma^*$*, definimos la configuración inicial en* ω *a la configuración:*

$$I_A(\omega) := (Q_0, \omega, Z_0) \in \mathcal{S}_A.$$

Llamaremos lenguaje aceptado (mediante estado final final aceptador) por el autómata A (y lo denotaremos por $L(A)$*) al conjunto siguiente:*

$$L(A) := \{\omega \in \Sigma^* \ : \ I_A(\omega) \vdash_A (f, \lambda, z) \in \mathcal{S}_A, \ f \in F\}.$$

Lenguaje Aceptado Mediante Estado Final

Este concepto nos lleva a un modelo de programa en el que la condición de parada viene dada por los estados finales aceptadores y por tener la cinta vacía.

Lenguaje Aceptado Mediante Pila Vacía

Entrada: $\omega \in \Sigma^*$
Inicializar: $I := (Q_0, \omega, Z_0)$.
mientras $I \notin F \times \{\lambda\} \times Z_0\Gamma^*$ **hacer**
  Hallar $c' \in S_A$ tal que $I \to_A c'$
  Com.: Realiza un paso en el sistema de transición.
  $I := c'$
fin hacer
Salida: ACEPTAR
fin

Lenguaje Aceptado por Pila y Cinta Vacías

Definición (Lenguaje aceptado mediante pila y cinta vacías)

Sea $A := (Q, \Sigma, \Gamma, Q_0, Z_0, F)$ *un autómata con pila y sea* $\mathcal{S}_A$ *el sistema de transición asociado. Llamaremos lenguaje aceptado (mediante pila y cinta vacías) por el autómata A (y lo denotaremos por* $L_\emptyset(A)$*) al conjunto siguiente:*

$$L(A) := \{\omega \in \Sigma^* \ : \ I_A(\omega) \vdash_A (f, \lambda, \lambda) \in \mathcal{S}_A, f \in Q \setminus \{Q_0\}.$$

Lenguaje Aceptado por Pila y Cinta Vacías

De la misma forma, se puede pensar en estos autómatas como un programa, donde la condición de parada es que tanto la pila como la cinta estén vacías.

Lenguaje Aceptado por Autómatas

En cualquiera de los dos casos, admitimos que los autómatas puedan operar indefinidamente, al menos en principio.

Los programas que hemos escrito son solo válidos para autómatas *deterministas*.

Lenguaje Aceptado por Autómatas

En cualquiera de los dos casos, admitimos que los autómatas puedan operar indefinidamente, al menos en principio.

Los programas que hemos escrito son solo válidos para autómatas *deterministas*.

Equivalencia

Teorema

Un lenguaje es aceptado por un autómata con pila mediante pila y cinta vacías si y solamente si es aceptado por un autómata con pila mediante estado final aceptador.

Algoritmo

Mostraremos un mecanismo de paso, construyendo para cada lenguaje $L(A)$ aceptado por un autómata con pila A mediante estado final aceptador un autómata con pila $\bar{A}$ que acepta el mismo lenguaje mediante pila y cintas vacías y recíprocamente.

Algoritmo

Dado un autómata con pila $A := (Q, \Sigma, \Gamma, Q_0, Z_0, F, \delta)$ que acepta el lenguaje $L(A)$ mediante estado final aceptador, construyamos el nuevo autómata que aceptará el mismo lenguaje mediante pila vacía $\bar{A} := (\bar{Q}, \bar{\Sigma}, \bar{\Gamma}, \bar{Q}_0, \bar{Z}_0, \bar{F}, \bar{\delta})$

Algoritmo

- Sea $p_0, p_f \notin Q$ dos nuevos estados y definamos $\bar{Q} := Q \cup \{p_0, p_f\}$.
- $\bar{\Sigma} := \Sigma, \bar{Q}_0 := p_0, \bar{Z}_0 := Z_0$.
- La idea clave consiste en introducir un nuevo símbolo en el alfabeto de la pila X_0 que "protegerá" el símbolo de fondo de la pila. Así, elegiremos $X_0 \notin \Gamma$ y $\bar{\Gamma} := \Gamma \cup \{X_0\}$.
- $\bar{F} := F$, dejamos el mismo conjunto de estados finales aceptadores, aunque no son necesarios.

Algoritmo

- Definamos

$$\bar{\delta} : \bar{Q} \times (\bar{\Sigma} \cup \{\lambda\}) \times (\bar{\Gamma} \cup \{Z_0\}) \to \bar{Q} \times \bar{\Gamma}^*,$$

mediante:

  - $\bar{\delta}(p_0, w, Z_0) = (Q_0, Z_0 X_0)$. Es decir, inicializamos "protegiendo" Z_0 con una variable X_0.
  - Mientras "vivamos" en el "viejo" autómata no cambiamos la función de transición.
  - Si alguna vez leemos Z_0, actuamos como si el autómata antiguo se hubiera detenido.
  - En caso de llegar a un estado final, nos movemos a p_f y vaciamos la pila.

Algoritmo

Recíprocamente, dado un autómata con pila $A := (Q, \Sigma, \Gamma, Q_0, Z_0, F, \delta)$ que acepta el lenguaje $L_\emptyset(A)$ mediante pila y cinta vacías, construyamos el nuevo autómata que aceptará el mismo lenguaje mediante estados finales aceptadores $\bar{A} := (\bar{Q}, \bar{\Sigma}, \bar{\Gamma}, \bar{Q}_0, \bar{Z}_0, \bar{F}, \bar{\delta})$

Algoritmo

Introduciremos un estado final aceptador nuevo p_f y definimos $\bar{F} := \{p_f\}, \bar{Q} := Q \cup \{p_f\}$. Introducimos un nuevo símbolo inicial para la pila $\bar{Z}_0 := X_0$ y definimos $\bar{\Gamma} := \Gamma \cup \{Z_0\}$.

Algoritmo

La idea en este caso, es que cuando alcancemos el fondo de pila en el autómata sera el momento en el que el nuevo autómata se moverá al nuevo estado final, ya que *las computaciones podrán seguir ya que la pila tiene en el fondo* X_0*.*

Resultado

Teorema

Los lenguajes libres de contexto son exactamente los lenguajes aceptados por los autómatas con pila mediante cinta y pila vacías. Es decir, se verifican las siguiente dos propiedades:

1. *Para cada gramática libre de contexto G sobre un alfabeto Σ de símbolos terminales, existe un autómata con pila A tal que $L(G) = L_\emptyset(A)$.*
2. *Para cada autómata A con alfabeto de cinta Σ existe una gramática libre de contexto G tal que el lenguaje generado por G coincide con $L_\emptyset(A)$.*

Más aún, daremos procedimientos de construcción en ambos sentidos.

Algoritmo

Bastará con lo probemos para gramáticas en Forma Normal de Chomsky. El resto se obtiene en las progresivas transformaciones de gramáticas. Así, supongamos que G es dada mediante $G := (V, \Sigma, Q_0, P)$, donde Q_0 es el símbolo inicial.

Algoritmo

Definiremos un autómata con pila $A := (Q, \Sigma, \Gamma, Q_0, Z_0, F, \delta)$ de la forma siguiente:

- $Q := \{Q_0\}$ posee un único estado (que es también el estado inicial).
- El símbolo de fondo de la pila es un símbolo auxiliar.
- El alfabeto de la pila reune a todos los símbolos (terminales o no) de la gramática $\Gamma := V \cup \Sigma \cup \{Q_0\}$.

Algoritmo

La función de transición estará dada del modo siguiente:

- $\delta(Q_0, \lambda, Z_0) := (Q_0, Z_0 Q_0)$ (al comenzar pongamos Q_0 justo encima del fondo de la pila).
- Si la gramática tiene una producción del tipo $A \mapsto a \in \Sigma \cup \{\lambda\}$, escribamos:

$$\delta(Q_0, \lambda, A) := (Q_0, a).$$

- Si la gramática tiene una producción del tipo $A \mapsto CD$, con $C, D \in V$, pongamos:

$$\delta(Q_0, \lambda, A) := (Q_0, DC).$$

- Finalmente, para cada $a \in \Sigma$, pongamos:

$$\delta(Q_0, a, a) := (Q_0, \lambda).$$

Algoritmo

Para la segunda de las afirmaciones, consideremos dado un autómata con pila $A := (Q, \Sigma, \Gamma, Q_0, Z_0, \delta)$ que acepta un lenguaje $L_\emptyset(A)$.

Algoritmo

Construyamos la gramática $G := (V, \Sigma, Q_0, P)$ dependiendo del alfabeto de la pila y los estados que tenga el autómata.

Algoritmo

$V := Q \times (\Gamma \cup \{Z_0\}) \times Q \cup \{Q_0\}$. Utilizaremos la notación $\langle qAp \rangle$ para representar el símbolo no terminal $(q, A, p) \in V$.

Algoritmo

El símbolo inicial Q_0 lleva acompañada unas producciones del tipo siguiente:

$$Q_0 \mapsto \langle Q_0 Z_0 p \rangle,$$

para cada $p \in Q$.

Algoritmo

Si la función de transición δ satisface $\delta(p, a, A) = (q, \lambda)$ con $a \in \Sigma \cup \{\lambda\}$ y $A \in \Gamma$, escribiremos la producción:

$$\langle pAq \rangle \mapsto a.$$

Algoritmo

Si la función de transición δ satisface $\delta(p, a, A) = (q, B_1 \cdots B_n)$ con $a \in \Sigma \cup \{\lambda\}$ y $B_1, \ldots, B_n \in \Gamma \cup \{Z_0\}$, escribiremos las producciones siguientes:

$$\langle pAq \rangle \mapsto a \langle pB_n s_1 \rangle \langle s_1 B_{n-1} s_2 \rangle \langle s_2 B_{n-2} s_3 \rangle \cdots \langle s_{n-1} B_1 q \rangle,$$

para todos los estados $(s_1, \ldots, s_{n-2}) \in Q^{n-2}$.

Observaciones

Nótese que la construcción de la gramática asociada a un autómata con pila introduce un número exponencial (en el número de estados) de producciones por lo que es poco aconsejable utilizar esa construcción.

Observaciones

También se puede demostrar que se pueden utilizar autómatas que tengan un solo estado, lo que hace que la formula se simplifique un poco.