Una Jerarquía de Lenguajes Formales y Autómatas II

José de Jesús Lavalle Martínez

Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación
Lenguajes Formales y Autómatas CCOS 014

Primavera 2021

Contenido

Motivación 1 I

- Si bien no es posible extender el poder de la máquina de Turing estándar complicando la estructura de la cinta, es posible limitarla restringiendo la forma en que se puede usar la cinta.

Motivación 1 I

- Ya hemos visto un ejemplo de esto con los autómatas de pila.

Motivación 1 I

- Un autómata de de pila puede considerarse como una máquina de Turing no determinista con una cinta que está restringida a usarse como una pila.

Motivación 1 I

- También podemos restringir el uso de la cinta de otras formas; por ejemplo, podríamos permitir que sólo una parte finita de la cinta se utilice como espacio de trabajo.

Motivación 1 I

- Se puede demostrar que esto nos lleva de regreso a los autómatas finitos, por lo que no necesitamos seguir adelante.

Motivación 1 I

- Pero hay una forma de limitar el uso de la cinta que conduce a una situación más interesante: permitimos que la máquina utilice solo la parte de la cinta ocupada por la entrada.

Motivación 1 II

- Por lo tanto, hay más espacio disponible para cadenas de entrada largas que para cadenas cortas, lo que genera otra clase de máquinas, los **autómatas acotados linealmente** (o **lba**).

Motivación 1 II

- Un autómata acotado linealmente, como una máquina de Turing estándar, tiene una cinta ilimitada, pero la cantidad de cinta que se puede utilizar depende de la entrada.

Motivación 1 II

- En particular, restringimos la parte utilizable de la cinta a exactamente las celdas tomadas por la entrada.

Motivación 1 II

- En algunas definiciones, la parte utilizable de la cinta es un múltiplo de la longitud de entrada, donde el múltiplo puede depender del lenguaje, pero no de la entrada.

Motivación 1 II

- Aquí usamos sólo la longitud exacta de la cadena de entrada, pero permitimos máquinas multipistas, con la entrada en una sola pista.

Motivación 1 III

- Para hacer cumplir esto, podemos imaginar la entrada entre corchetes con dos símbolos especiales, el **marcador del extremo izquierdo** [y el **marcador del extremo derecho**].

Motivación 1 III

- Para una entrada w, la configuración inicial de la máquina de Turing viene dada por la descripción instantánea $q_0[w]$.

Motivación 1 III

- Los marcadores de finalización no se pueden reescribir y la cabeza de lectura y escritura no se puede mover a la izquierda de [ni a la derecha de].

Motivación 1 III

- A veces decimos que la cabeza de lectura y escritura “rebota” en los marcadores finales.

Autómatas acotados linealmente I

Definición 1

Un autómata acotado linealmente es una máquina de Turing no determinista $M = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$, sujeto a la restricción de que Σ debe contener dos símbolos especiales [y], tal que $\delta(q_i, [)$ sólo puede contener elementos de la forma $(q_j, [, R)$, y $\delta(q_i,])$ sólo puede contener elementos de la forma $(q_j,], L)$.

Autómatas acotados linealmente II

Definición 2

Una cadena w es aceptada por un autómata acotado linealmente si hay una posible secuencia de movimientos

$$q_0[w] \overset{*}{\vdash} [x_1 q_f x_2]$$

para algunos $q_f \in F, x_1, x_2 \in \Gamma^*$. El lenguaje aceptado por el lba es el conjunto de todas las cadenas aceptadas.

Ejemplo 1

Ejemplo 1

El lenguaje

$$L = \{a^n b^n c^n : n \geq 1\}$$

es aceptado por algún autómata acotado linealmente.

Ejemplo 1

Ejemplo 1

El lenguaje

$$L = \{a^n b^n c^n : n \geq 1\}$$

es aceptado por algún autómata acotado linealmente.

Solución:

- Una estrategia para reconocerlo es cambiar los símbolos a, b y c, por x, y y z, al final si todos los símbolos de entrada fueron reemplazados entonces se acepta la cadena, de lo contrario se rechaza.

□

Ejemplo 1

Ejemplo 1

El lenguaje

$$L = \{a^n b^n c^n : n \geq 1\}$$

es aceptado por algún autómata acotado linealmente.

Solución:

- Una estrategia para reconocerlo es cambiar los símbolos a, b y c, por x, y y z, al final si todos los símbolos de entrada fueron reemplazados entonces se acepta la cadena, de lo contrario se rechaza.
- Note que este cálculo no requiere más espacio que el ocupado por la cadena de entrada, así que puede ser realizado por un autómata acotado linealmente.

□

Ejemplo 2 I

Ejemplo 2

Encuentre un autómata acotado linealmente que acepte el lenguaje

$$L = \{a^{n!} : n \geq 0\}.$$

Ejemplo 2 I

Ejemplo 2

Encuentre un autómata acotado linealmente que acepte el lenguaje

$$L = \{a^{n!} : n \geq 0\}.$$

Solución:

- Una forma de resolver el problema es dividir el número de símbolos a sucesivamente por $2, 3, 4, \ldots$, hasta que podamos aceptar o rechazar la cadena.

Ejemplo 2 I

Ejemplo 2

Encuentre un autómata acotado linealmente que acepte el lenguaje

$$L = \{a^{n!} : n \geq 0\}.$$

Solución:

- Si la entrada está en L, eventualmente quedará una sola a; si no, en algún momento surgirá un resto distinto de cero.

Ejemplo 2 I

Ejemplo 2

Encuentre un autómata acotado linealmente que acepte el lenguaje

$$L = \{a^{n!} : n \geq 0\}.$$

Solución:

- Esbozamos la solución para señalar una implicación tácita de la Definición 1.

Ejemplo 2 I

Ejemplo 2

Encuentre un autómata acotado linealmente que acepte el lenguaje

$$L = \{a^{n!} : n \geq 0\}.$$

Solución:

- Dado que la cinta de un autómata acotado linealmente puede ser multipista, las pistas adicionales se pueden utilizar como espacio de trabajo.

Ejemplo 2 II

- Para este problema, podemos utilizar una cinta de dos pistas.

Ejemplo 2 II

- La primera pista contiene el número de símbolos a que quedan durante el proceso de división, y la segunda pista contiene el divisor actual (Figura 1).

Ejemplo 2 II

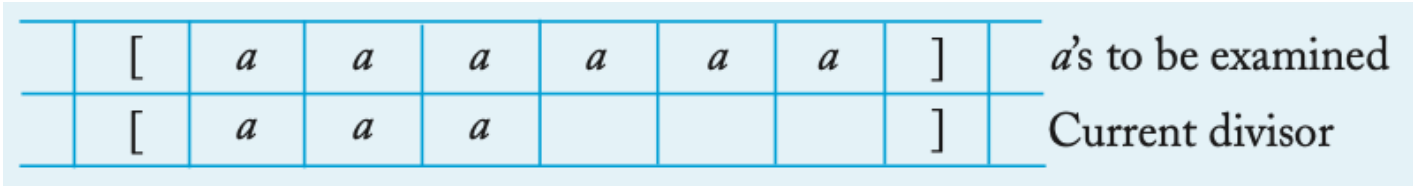


Figure 1: Las dos pistas del autómata acotado linealmente para el Ejemplo 2.

Ejemplo 2 II

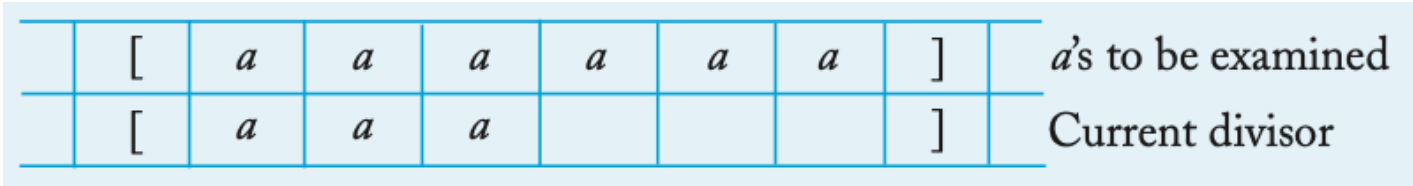
- 

Figure 1: Las dos pistas del autómata acotado linealmente para el Ejemplo 2.

- La solución real es bastante simple.

Ejemplo 2 II

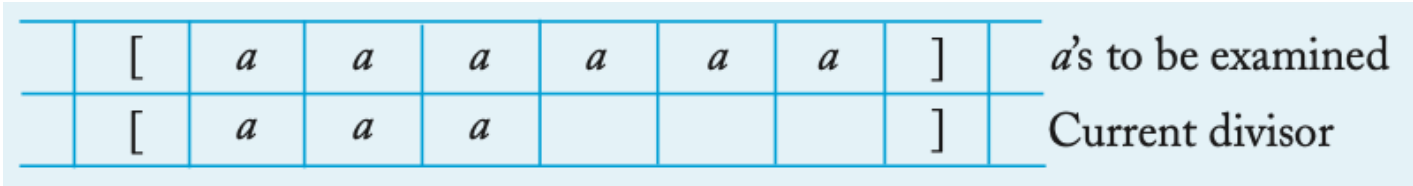


Figure 1: Las dos pistas del autómata acotado linealmente para el Ejemplo 2.

- Usando el divisor en la segunda pista, dividimos el número de símbolos a en la primera pista, por ejemplo, eliminando todos los símbolos excepto aquellos en múltiplos del divisor.

Ejemplo 2 II

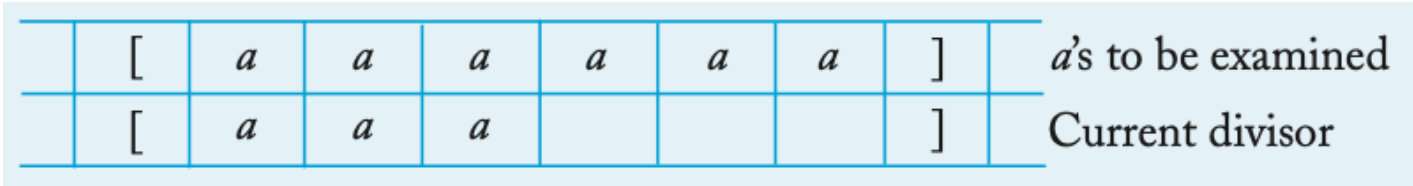


Figure 1: Las dos pistas del autómata acotado linealmente para el Ejemplo 2.

- Después de esto, incrementamos el divisor en uno y continuamos hasta que encontremos un resto distinto de cero o nos quedemos con una sola a.

Autómatas acotados linealmente III

- Los dos últimos ejemplos sugieren que los autómatas acotados linealmente son más poderosos que los autómatas de pila, ya que ninguno de dichos lenguajes es libre de contexto.

Autómatas acotados linealmente III

- Para probar tal conjetura, todavía tenemos que demostrar que cualquier lenguaje libre de contexto puede ser aceptado por un autómata acotado linealmente.

Autómatas acotados linealmente III

- Haremos esto más tarde de una manera un tanto indirecta.

Autómatas acotados linealmente III

- No es tan fácil hacer una conjetura sobre la relación entre las máquinas de Turing y los autómatas acotados linealmente.

Autómatas acotados linealmente IV

- Problemas como el Ejemplo 2 se pueden resolver invariablemente mediante un autómata acotado linealmente, ya que se dispone de una cantidad de espacio temporal proporcional a la longitud de la entrada.

Autómatas acotados linealmente IV

- De hecho, es bastante difícil llegar a un lenguaje concreto y explícitamente definido que no pueda ser aceptado por ningún autómata acotado linealmente.

Autómatas acotados linealmente IV

- Posteriormente mostraremos que la clase de autómatas acotados linealemente es menos poderosa que la clase de máquinas de Turing no restringidas, pero una demostración de esto requiere mucho más trabajo.

Ejercicios 1

Explique cómo se podrían construir autómatas acotados linealmente para aceptar los siguientes lenguajes:

1. $L = \{a^n : n = m^2, m \geq 1\}$.
2. $L = \{a^n : n$ es un número primo$\}$.
3. $L = \{ww : w \in \{a, b\}^+\}$.

Motivación 2

- Para investigar la conexión entre lenguajes y gramáticas recursivamente enumerables, volvemos a la definición general de una gramática en el Capítulo 1.

Motivación 2

- En la Definición 1.2.1 se permitió que las reglas de producción tomaran cualquier forma, pero luego se establecieron varias restricciones para obtener tipos de gramáticas específicas.

Motivación 2

- Si tomamos la forma general y no imponemos restricciones, obtenemos gramáticas no restringidas.

Gramáticas no restringidas I

Definición 3

Una gramática $G = (V, T, S, P)$ se llama **no restringida** (**sin restricciones**) si todas las producciones son de la forma

$$u \to v,$$

donde u está en $(V \cup T)^+$ y v está en $(V \cup T)^*$.

Gramáticas no restringidas II

- En una gramática no restringida, esencialmente no se imponen condiciones a las producciones.

Gramáticas no restringidas II

- Cualquier número de variables y terminales puede estar a la izquierda o a la derecha, y estos pueden ocurrir en cualquier orden.

Gramáticas no restringidas II

- Solo hay una restricción: λ no está permitido como el lado izquierdo de una producción.

Gramáticas no restringidas II

- Como veremos, las gramáticas no restringidas son mucho más poderosas que las formas restringidas como las gramáticas regulares y libres de contexto que hemos estudiado hasta ahora.

Gramáticas no restringidas II

- De hecho, las gramáticas no restringidas corresponden a la familia más grande de lenguajes, por lo que podemos esperar reconocerlas por medios mecánicos; es decir, las gramáticas no restringidas generan exactamente la familia de lenguajes enumerables de forma recursiva.

Gramáticas no restringidas II

- Mostramos esto en dos partes; el primero es bastante sencillo, pero el segundo implica una construcción larga.

Gramáticas no restringidas III

Teorema 1

Cualquier lenguaje generado por una gramática sin restricciones es enumerable de forma recursiva.

Gramáticas no restringidas III

Teorema 1

Cualquier lenguaje generado por una gramática sin restricciones es enumerable de forma recursiva.

Demostración:

- La gramática en efecto define un procedimiento para enumerar todas las cadenas en el lenguaje de forma sistemática.

Teorema 1

Cualquier lenguaje generado por una gramática sin restricciones es enumerable de forma recursiva.

Demostración:

- Por ejemplo, podemos enumerar todos las w en L de manera que

$$S \Rightarrow w,$$

es decir, w se deriva en un paso.

Gramáticas no restringidas III

Teorema 1

Cualquier lenguaje generado por una gramática sin restricciones es enumerable de forma recursiva.

Demostración:

- Dado que el conjunto de producciones de la gramática es finito, habrá un número finito de tales cadenas.

Gramáticas no restringidas IV

- A continuación, enumeramos todos los w en L que se pueden derivar en dos pasos

$$S \Rightarrow x \Rightarrow w,$$

y así.

Gramáticas no restringidas IV

- Podemos simular estas derivaciones en una máquina de Turing y, por lo tanto, tener un procedimiento de enumeración para el lenguaje.

Gramáticas no restringidas IV

- Por tanto, es recursivamente enumerable.

■

Gramáticas no restringidas V

- Esta parte de la correspondencia entre lenguajes enumerables de forma recursiva y gramáticas no restringidas no es sorprendente.

Gramáticas no restringidas V

- La gramática genera cadenas mediante un proceso algorítmico bien definido, por lo que las derivaciones se pueden realizar en una máquina de Turing.

Gramáticas no restringidas V

- Para mostrar la recíproca, describimos cómo cualquier máquina de Turing puede ser imitada por una gramática sin restricciones.

Gramáticas no restringidas V

- Se nos da una máquina de Turing $M = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$ y queremos producir una gramática G tal que $L(G) = L(M)$.

Gramáticas no restringidas V

- La idea detrás de la construcción es relativamente simple, pero su implementación se vuelve notoriamente engorrosa.

Gramáticas no restringidas VI

- Dado que el cálculo de la máquina de Turing puede describirse mediante la secuencia de descripciones instantáneas

$$q_0 w \overset{*}{\vdash} x q_f w, \tag{1}$$

Gramáticas no restringidas VI

- Dado que el cálculo de la máquina de Turing puede describirse mediante la secuencia de descripciones instantáneas

$$q_0 w \overset{*}{\vdash} x q_f w, \tag{1}$$

- intentaremos ordenarlo para que la gramática correspondiente tenga la propiedad de que

$$q_0 w \overset{*}{\Rightarrow} x q_f w, \tag{2}$$

si y sólo si (1) se cumple.

Gramáticas no restringidas VI

- Dado que el cálculo de la máquina de Turing puede describirse mediante la secuencia de descripciones instantáneas

$$q_0 w \overset{*}{\vdash} x q_f w, \tag{1}$$

- intentaremos ordenarlo para que la gramática correspondiente tenga la propiedad de que

$$q_0 w \overset{*}{\Rightarrow} x q_f w, \tag{2}$$

  si y sólo si (1) se cumple.
- Esto no es difícil de hacer; lo que es más difícil de ver es cómo hacer la conexión entre (2) y lo que realmente queremos, es decir,

$$S \Rightarrow w$$

  para toda w que satisfaga (1).

Gramáticas no restringidas VII

- Para lograrlo, construimos una gramática que, a grandes rasgos, tiene las siguientes propiedades:
  1. S puede derivar $q_0 w$ para toda $w \in \Sigma^+$.
  2. (2) es posible si y sólo si (1) se cumple.
  3. Cuando se genera una cadena $xq_f y$ con $q_f \in F$, la gramática transforma esta cadena en la w original.

Gramáticas no restringidas VII

- Para lograrlo, construimos una gramática que, a grandes rasgos, tiene las siguientes propiedades:
  1. S puede derivar $q_0 w$ para toda $w \in \Sigma^+$.
  2. (2) es posible si y sólo si (1) se cumple.
  3. Cuando se genera una cadena $xq_f y$ con $q_f \in F$, la gramática transforma esta cadena en la w original.
- La secuencia completa de derivaciones es entonces

$$S \overset{*}{\Rightarrow} q_0 w \overset{*}{\Rightarrow} xq_f y \overset{*}{\Rightarrow} w. \tag{3}$$

Gramáticas no restringidas VII

- Para lograrlo, construimos una gramática que, a grandes rasgos, tiene las siguientes propiedades:
  1. S puede derivar $q_0 w$ para toda $w \in \Sigma^+$.
  2. (2) es posible si y sólo si (1) se cumple.
  3. Cuando se genera una cadena $xq_f y$ con $q_f \in F$, la gramática transforma esta cadena en la w original.
- La secuencia completa de derivaciones es entonces

$$S \overset{*}{\Rightarrow} q_0 w \overset{*}{\Rightarrow} xq_f y \overset{*}{\Rightarrow} w. \tag{3}$$

- El tercer paso en la derivación anterior es el problemático.

Gramáticas no restringidas VII

- Para lograrlo, construimos una gramática que, a grandes rasgos, tiene las siguientes propiedades:
  1. S puede derivar $q_0 w$ para toda $w \in \Sigma^+$.
  2. (2) es posible si y sólo si (1) se cumple.
  3. Cuando se genera una cadena $xq_f y$ con $q_f \in F$, la gramática transforma esta cadena en la w original.
- La secuencia completa de derivaciones es entonces

$$S \overset{*}{\Rightarrow} q_0 w \overset{*}{\Rightarrow} xq_f y \overset{*}{\Rightarrow} w. \tag{3}$$

- El tercer paso en la derivación anterior es el problemático.
- ¿Cómo puede la gramática recordar w si se modifica durante el segundo paso?

Gramáticas no restringidas VIII

- Resolvemos esto codificando cadenas de modo que la versión codificada originalmente tenga dos copias de w.

Gramáticas no restringidas VIII

- Resolvemos esto codificando cadenas de modo que la versión codificada originalmente tenga dos copias de w.
- La primera se guarda, mientras que la segunda se usa en los pasos de (2). Cuando se ingresa una configuración final, la gramática borra todo excepto la w guardada.

Gramáticas no restringidas VIII

- Resolvemos esto codificando cadenas de modo que la versión codificada originalmente tenga dos copias de w.
- La primera se guarda, mientras que la segunda se usa en los pasos de (2). Cuando se ingresa una configuración final, la gramática borra todo excepto la w guardada.
- Para producir dos copias de w y manejar el símbolo de estado de M (que eventualmente tiene que ser eliminado por la gramática), introducimos las variables V_{ab} y V_{aib} para todo $a \in \Sigma \cup \{\square\}$, $b \in \Gamma$ y todo i tal que $q_i \in Q$.

Gramáticas no restringidas VIII

- Resolvemos esto codificando cadenas de modo que la versión codificada originalmente tenga dos copias de w.
- La primera se guarda, mientras que la segunda se usa en los pasos de (2). Cuando se ingresa una configuración final, la gramática borra todo excepto la w guardada.
- Para producir dos copias de w y manejar el símbolo de estado de M (que eventualmente tiene que ser eliminado por la gramática), introducimos las variables V_{ab} y V_{aib} para todo $a \in \Sigma \cup \{\square\}$, $b \in \Gamma$ y todo i tal que $q_i \in Q$.
- La variable V_{ab} codifica los dos símbolos a y b, mientras que V_{aib} codifica a y b, así como el estado q_i.

Gramáticas no restringidas IX

- El primer paso en (3) se puede lograr (en la forma codificada) mediante

$$S \to V_{\square\square}S|SV_{\square\square}|T, \tag{4}$$
$$T \to TV_{aa}|V_{a0a}, \tag{5}$$

para todo $a \in \Sigma$.

Gramáticas no restringidas IX

- El primer paso en (3) se puede lograr (en la forma codificada) mediante

$$S \to V_{\square\square}S|SV_{\square\square}|T, \tag{4}$$
$$T \to TV_{aa}|V_{a0a}, \tag{5}$$

  para todo $a \in \Sigma$.
- Estas producciones permiten que la gramática genere una versión codificada de cualquier cadena q_0w con un número arbitrario de espacios en blanco al principio y al final.

Gramáticas no restringidas X

- Para el segundo paso, para cada transición

$$\delta(q_i, c) = (q_j, d, R)$$

de M, ponemos en la gramática producciones de la forma

$$V_{aic}V_{pq} \rightarrow V_{ad}V_{pjq}, \tag{6}$$

para todo $a, p \in \Sigma \cup \{\square\}$, $q \in \Gamma$.

Gramáticas no restringidas X

- Para el segundo paso, para cada transición

$$\delta(q_i, c) = (q_j, d, R)$$

de M, ponemos en la gramática producciones de la forma

$$V_{aic}V_{pq} \rightarrow V_{ad}V_{pjq}, \tag{6}$$

para todo $a, p \in \Sigma \cup \{\square\}$, $q \in \Gamma$.

- Para cada

$$\delta(q_i, c) = (q_j, d, L)$$

de M, incluimos en G

$$V_{pq}V_{aic} \rightarrow V_{pjq}V_{ad}, \tag{7}$$

para todo $a, p \in \Sigma \cup \{\square\}, q \in \Gamma$.

Gramáticas no restringidas XI

- Si en el segundo paso, M entra en un estado final, la gramática debe entonces deshacerse de todo excepto w, que se guarda en los primeros índices de las V.

Gramáticas no restringidas XI

- Si en el segundo paso, M entra en un estado final, la gramática debe entonces deshacerse de todo excepto w, que se guarda en los primeros índices de las V.
- Por lo tanto, para cada $q_j \in F$, incluimos producciones

$$V_{ajb} \to a, \tag{8}$$

para todo $a \in \Sigma \cup \{\square\}, b \in \Gamma$.

Gramáticas no restringidas XI

- Si en el segundo paso, M entra en un estado final, la gramática debe entonces deshacerse de todo excepto w, que se guarda en los primeros índices de las V.
- Por lo tanto, para cada $q_j \in F$, incluimos producciones

$$V_{ajb} \to a, \tag{8}$$

para todo $a \in \Sigma \cup \{\square\}, b \in \Gamma$.

- Esto crea el primer terminal en la cadena, que luego provoca una reescritura en el resto mediante

$$cV_{ab} \to ca, \tag{9}$$
$$V_{ab}c \to ac, \tag{10}$$

para todo $a, c \in \Sigma \cup \{\square\}, b \in \Gamma$.

Gramáticas no restringidas XII

- También necesitamos una producción especial

$$\square \rightarrow \lambda. \tag{11}$$

Gramáticas no restringidas XII

- También necesitamos una producción especial

$$\square \rightarrow \lambda. \tag{11}$$

- Esta última producción se ocupa del caso en que M se mueva fuera de la parte de la cinta ocupada por la entrada w.

Gramáticas no restringidas XII

- También necesitamos una producción especial

$$\square \rightarrow \lambda. \tag{11}$$

- Esta última producción se ocupa del caso en que M se mueva fuera de la parte de la cinta ocupada por la entrada w.
- Para que las cosas funcionen en este caso, primero debemos usar (4) y (5) para generar

$$\square \cdots \square q_0 w \square \cdots \square,$$

que representa toda la región de la cinta utilizada.

Gramáticas no restringidas XII

- También necesitamos una producción especial

$$\square \to \lambda. \tag{11}$$

- Esta última producción se ocupa del caso en que M se mueva fuera de la parte de la cinta ocupada por la entrada w.
- Para que las cosas funcionen en este caso, primero debemos usar (4) y (5) para generar

$$\square \cdots \square q_0 w \square \cdots \square,$$

que representa toda la región de la cinta utilizada.

- Los espacios en blanco extraños se eliminan al final por (11).

Gramáticas no restringidas XII

- También necesitamos una producción especial

$$\square \to \lambda. \tag{11}$$

- Esta última producción se ocupa del caso en que M se mueva fuera de la parte de la cinta ocupada por la entrada w.
- Para que las cosas funcionen en este caso, primero debemos usar (4) y (5) para generar

$$\square \cdots \square q_0 w \square \cdots \square,$$

que representa toda la región de la cinta utilizada.

- Los espacios en blanco extraños se eliminan al final por (11).
- El siguiente ejemplo ilustra esta complicada construcción.

Gramáticas no restringidas XII

- También necesitamos una producción especial

$$\square \to \lambda. \tag{11}$$

- Esta última producción se ocupa del caso en que M se mueva fuera de la parte de la cinta ocupada por la entrada w.
- Para que las cosas funcionen en este caso, primero debemos usar (4) y (5) para generar

$$\square \cdots \square q_0 w \square \cdots \square,$$

que representa toda la región de la cinta utilizada.

- Los espacios en blanco extraños se eliminan al final por (11).
- El siguiente ejemplo ilustra esta complicada construcción.
- Revise cuidadosamente cada paso en el ejemplo para ver qué hacen las distintas producciones y por qué son necesarias.

Ejemplo 1 I

Ejemplo 3

Sea $M = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$ una máquina de Turing con

$$Q = \{q_0, q_1\},$$
$$\Gamma = \{a, b, \square\},$$
$$\Sigma = \{a, b\},$$
$$F = \{q_1\},$$

y

$$\delta(q_0, a) = (q_0, a, R),$$
$$\delta(q_0, \square) = (q_1, \square, L).$$

Esta máquina acepta $L(aa^*)$.

Ejemplo 1 II

- Considere ahora el cálculo

$$q_0aa \vdash aq_0a \vdash aaq_0\square \vdash aq_1a\square, \tag{12}$$

que acepta la cadena aa.

Ejemplo 1 II

- Considere ahora el cálculo

$$q_0aa \vdash aq_0a \vdash aaq_0\square \vdash aq_1a\square, \tag{12}$$

que acepta la cadena aa.

- Para derivar esta cadena con G, primero usamos reglas de la forma (4) y (5) para obtener la cadena inicial apropiada,

$$S \Rightarrow SV_{\square\square} \Rightarrow TV_{\square\square} \Rightarrow TV_{aa}V_{\square\square} \Rightarrow V_{a0a}V_{aa}V_{\square\square}.$$

Ejemplo 1 II

- Considere ahora el cálculo

$$q_0aa \vdash aq_0a \vdash aaq_0\square \vdash aq_1a\square, \tag{12}$$

que acepta la cadena aa.

- Para derivar esta cadena con G, primero usamos reglas de la forma (4) y (5) para obtener la cadena inicial apropiada,

$$S \Rightarrow SV_{\square\square} \Rightarrow TV_{\square\square} \Rightarrow TV_{aa}V_{\square\square} \Rightarrow V_{a0a}V_{aa}V_{\square\square}.$$

- La última forma sentencial es el punto de partida para la parte de la derivación que imita el cálculo de la máquina de Turing.

Ejemplo 1 II

- Considere ahora el cálculo

$$q_0aa \vdash aq_0a \vdash aaq_0\square \vdash aq_1a\square, \tag{12}$$

que acepta la cadena aa.

- Para derivar esta cadena con G, primero usamos reglas de la forma (4) y (5) para obtener la cadena inicial apropiada,

$$S \Rightarrow SV_{\square\square} \Rightarrow TV_{\square\square} \Rightarrow TV_{aa}V_{\square\square} \Rightarrow V_{a0a}V_{aa}V_{\square\square}.$$

- La última forma sentencial es el punto de partida para la parte de la derivación que imita el cálculo de la máquina de Turing.
- Contiene la entrada original $aa\square$ en la secuencia de los primeros índices y la descripción instantánea inicial $q_0aa\square$ en los índices restantes.

Ejemplo 1 III

- A continuación, aplicamos

$$V_{a0a}V_{aa} \rightarrow V_{aa}V_{a0a},$$

y

$$V_{a0a}V_{\square\square} \rightarrow V_{aa}V_{\square 0\square},$$

que son instancias específicas de (6), y

$$V_{aa}V_{\square 0\square} \rightarrow V_{a1a}V_{\square\square}$$

procedente de (7).

Ejemplo 1 III

- A continuación, aplicamos

$$V_{a0a}V_{aa} \to V_{aa}V_{a0a},$$

y

$$V_{a0a}V_{\square\square} \to V_{aa}V_{\square 0\square},$$

que son instancias específicas de (6), y

$$V_{aa}V_{\square 0\square} \to V_{a1a}V_{\square\square}$$

procedente de (7).

- Entonces los siguientes pasos en la derivación son

$$V_{a0a}V_{aa}V_{\square\square} \Rightarrow V_{aa}V_{a0a}V_{\square\square} \Rightarrow V_{aa}V_{aa}V_{\square 0\square} \Rightarrow V_{aa}V_{a1a}V_{\square}.$$

Ejemplo 1 III

- A continuación, aplicamos

$$V_{a0a}V_{aa} \to V_{aa}V_{a0a},$$

y

$$V_{a0a}V_{\square\square} \to V_{aa}V_{\square 0\square},$$

que son instancias específicas de (6), y

$$V_{aa}V_{\square 0\square} \to V_{a1a}V_{\square\square}$$

procedente de (7).

- Entonces los siguientes pasos en la derivación son

$$V_{a0a}V_{aa}V_{\square\square} \Rightarrow V_{aa}V_{a0a}V_{\square\square} \Rightarrow V_{aa}V_{aa}V_{\square 0\square} \Rightarrow V_{aa}V_{a1a}V_{\square}.$$

- La secuencia de los primeros índices sigue siendo la misma, siempre recordando la entrada inicial.

Ejemplo 1 IV

- La secuencia de los otros índices es

$$0aa\square, a0a\square, aa0\square, a1a\square,$$

que es equivalente a la secuencia de descripciones instantáneas en (12).

Ejemplo 1 IV

- La secuencia de los otros índices es

$$0aa\square, a0a\square, aa0\square, a1a\square,$$

que es equivalente a la secuencia de descripciones instantáneas en (12).

- Finalmente, (8) a (11) se utilizan en los últimos pasos

$$V_{aa}V_{a1a}V_{\square\square} \Rightarrow V_{aa}aV_{\square\square} \Rightarrow V_{aa}a\square \Rightarrow aa\square \Rightarrow aa.$$

Ejemplo 1 IV

- La secuencia de los otros índices es

$$0aa\square, a0a\square, aa0\square, a1a\square,$$

que es equivalente a la secuencia de descripciones instantáneas en (12).

- Finalmente, (8) a (11) se utilizan en los últimos pasos

$$V_{aa}V_{a1a}V_{\square\square} \Rightarrow V_{aa}aV_{\square\square} \Rightarrow V_{aa}a\square \Rightarrow aa\square \Rightarrow aa.$$

- La construcción descrita en (4) a (11) es la base de la prueba del siguiente resultado.

□

Gramáticas no restringidas XIII

Teorema 2

Para cada lenguaje L recursivamente enumerable, existe una gramática G sin restricciones, tal que $L = L(G)$.

Teorema 2

Para cada lenguaje L recursivamente enumerable, existe una gramática G sin restricciones, tal que $L = L(G)$.

Demostración:

- La construcción descrita garantiza que si

$$x \vdash y,$$

entonces

$$e(x) \Rightarrow e(y)$$

donde $e(x)$ denota la codificación de una cadena de acuerdo con la convención dada.

Gramáticas no restringidas XIV

- Mediante inducción sobre el número de pasos, podemos demostrar que

$$e(q_0 w) \stackrel{*}{\Rightarrow} e(y)$$

si y sólo si

$$q_0 w \stackrel{*}{\vdash} y.$$

Gramáticas no restringidas XIV

- Mediante inducción sobre el número de pasos, podemos demostrar que

$$e(q_0 w) \stackrel{*}{\Rightarrow} e(y)$$

si y sólo si

$$q_0 w \stackrel{*}{\vdash} y.$$

- También debemos mostrar que podemos generar todas las configuraciones iniciales posibles y que w se reconstruye adecuadamente si y sólo si M ingresa a una configuración final.

Gramáticas no restringidas XIV

- Mediante inducción sobre el número de pasos, podemos demostrar que

$$e(q_0 w) \stackrel{*}{\Rightarrow} e(y)$$

si y sólo si

$$q_0 w \stackrel{*}{\vdash} y.$$

- También debemos mostrar que podemos generar todas las configuraciones iniciales posibles y que w se reconstruye adecuadamente si y sólo si M ingresa a una configuración final.
- Los detalles, que no son demasiado difíciles, se dejan como ejercicio.

■

Ejercicios 2

1. ¿Qué lenguaje deriva la siguiente gramática sin restricciones?

$$\begin{aligned} S &\to S_1 B, \\ S_1 &\to a S_1 b, \\ bB &\to bbbB, \\ aS_1 b &\to aa, \\ B &\to \lambda. \end{aligned}$$

2. Construya una máquina de Turing para $L(01(01)^*)$, luego encuentre una gramática sin restricciones para ella usando la construcción del Teorema 2. Dé una derivación para 0101 usando la gramática resultante.