Teoría de Lenguajes

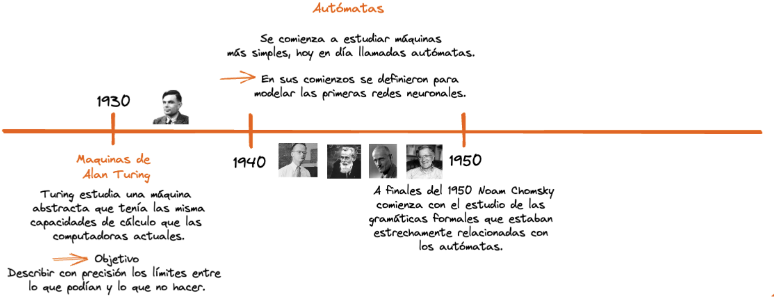
Autómatas de pila

Sabrina Silvero

Departamento de Computación,
Facultad de Ciencias Exactas y Naturales,
Universidad de Buenos Aires

Primer cuatrimestre 2025

¿Por qué estudiamos lo que estudiamos?



¿Por qué estudiamos lo que estudiamos?

1930

Maquinas de Alan Turing

Turing estudia una máquina abstracta que tenía las misma capacidades de cálculo que las computadoras actuales.

→ Objetivo
Describir con precisión los límites entre lo que podían y lo que no hacer.

Autómatas

Se comienza a estudiar máquinas más simples, hoy en día llamadas autómatas.

→ En sus comienzos se definieron para modelar las primeras redes neuronales.

1940

1950

A finales del 1950 Noam Chomsky comienza con el estudio de las gramáticas formales que estaban estrechamente relacionadas con los autómatas.

1960

Lenguajes libres de contexto

Toman un papel central en la tecnología de compilación.

¿Por qué estudiamos lo que estudiamos?

Los lenguajes independientes del contexto son reconocidos por los autómatas de pila y gramáticas libres de contexto.
Hoy vamos a trabajar con autómatas de pila, algunas utilidades de estos son:

- Análisis de protocolos de seguridad por Susan Landau
- Investigación en el razonamiento sobre ontologías y el procesamiento del lenguaje natural por Sheila McIlraith
- Análisis de modelos de sistemas de tiempo real por Christel Baier
- Verificación de programas concurrentes y la modelización de sistemas biológicos por Nancy Lynch


Susan Landau


Sheila McIlraith


Christel Baier


Nancy Lynch

Introducción

Lenguajes regulares

- Son los más simples, el tipo 3 de la jerarquía de Chomsky
- Reconocidos por autómatas finitos / expresiones regulares (formalismos equivalentes)
- No tienen memoria (no pueden contar, salvo que se requieran estados finito)
- Lema de pumping para demostrar que un lenguaje es no regular

Introducción

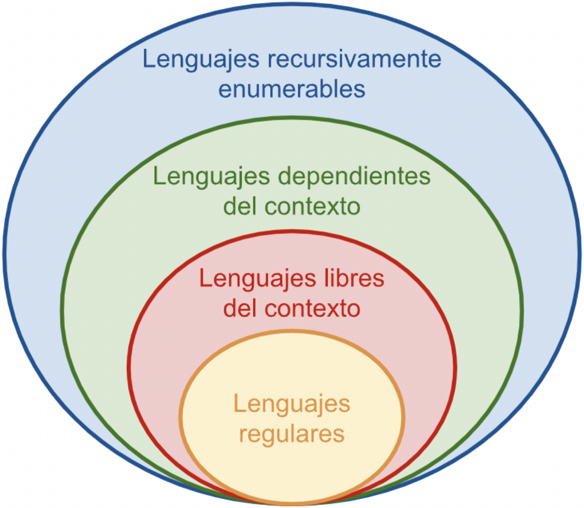
Lenguajes regulares

- Son los más simples, el tipo 3 de la jerarquía de Chomsky
- Reconocidos por autómatas finitos / expresiones regulares (formalismos equivalentes)
- No tienen memoria (no pueden contar, salvo que se requieran estados finito)
- Lema de pumping para demostrar que un lenguaje es no regular

Lenguajes independientes (libres) del contexto

- Tipo 2 en la jerarquía de Chomsky
- Reconocidos por autómatas de pila y gramáticas libres de contexto
- Mayor poder expresivo, la pila permite tener memoria y poder contar (ciertas cosas)

Jerarquía de Noam Chomsky



Autómatas de Pila

¿Podemos construir un autómata finito para reconocer los siguientes lenguajes?

$$L_a = \{s^3 i^3\}$$

Autómatas de Pila

¿Podemos construir un autómata finito para reconocer los siguientes lenguajes?

$$L_a = \{s^3 i^3\} \checkmark$$

$$L_b = \{s^n i^n \mid 1 \leq n\}$$

¿Podemos construir un autómata finito para reconocer los siguientes lenguajes?

$$L_a = \{s^3 i^3\} \checkmark$$

$$L_b = \{s^n i^n \mid 1 \leq n\} \times$$

$$L_c = \{s^j i^j \mid 1 \leq j \leq k \text{ con k fijo} \in \mathbb{N}\}$$

¿Podemos construir un autómata finito para reconocer los siguientes lenguajes?

$$L_a = \{s^3 i^3\} \ \checkmark$$

$$L_b = \{s^n i^n \mid 1 \leq n\} \ \times$$

$$L_c = \{s^j i^j \mid 1 \leq j \leq k \text{ con k fijo} \in \mathbb{N}\} \ \checkmark$$

$$L_b = L_1 = \{s^n i^n \mid 1 \leq n\}$$

En clases anteriores demostramos lenguajes muy similares a este no son regulares. Para poder reconocer lenguajes de este tipo necesitamos un formalismo con mayor poder expresivo: un autómata de pila.

Autómatas de Pila

$$L_b = L_1 = \{s^n i^n \mid 1 \leq n\}$$

En clases anteriores demostramos lenguajes muy similares a este no son regulares. Para poder reconocer lenguajes de este tipo necesitamos un formalismo con mayor poder expresivo: un autómata de pila.

- La pila es una cadena de símbolos
- Las transiciones pueden depender del símbolo en el tope de la pila, además del primer símbolo de la cadena de entrada
- En cada transición se desapila el símbolo del tope y se puede apilar una cadena

Definición

Un autómata de pila M se define como:

$$M = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F \rangle$$

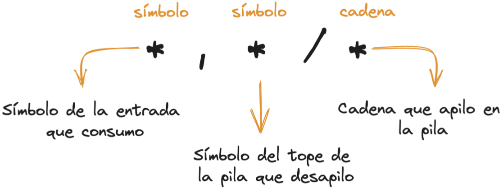
Donde:

- Q es un conjunto finito de estados
- Σ es el alfabeto de entrada
- Γ es el alfabeto de la pila
- $q_0 \in Q$ es el estado inicial
- $Z_0 \in \Gamma$ es el símbolo inicial de la pila
- $F \subseteq Q$ es el conjunto de estados finales
- δ es la función de transición: $Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \to P(Q \times \Gamma^*)$

Y recuerden que la tupla

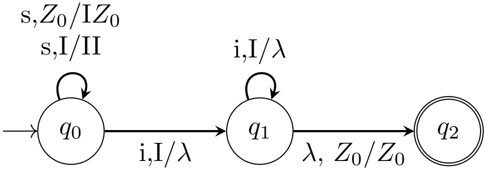
Formalizando

Las transiciones se definen de la siguiente manera:



Ejercicio 1

Dado el siguiente autómata
δ_1 :



$$M_1 = \langle \{q_0, q_1, q_2\}, \{s, i\}, \{I, Z_0\}, \delta, q_0, Z_0, \{q_2\} \rangle$$

Determinar si las cadenas pertenecen al lenguaje que reconoce M_1:

- ¿λ?
- ¿ssi?
- ¿$sssssiiiii$?
- ¿i?

Ejercicio 2

Construir un AP que reconozca el siguiente lenguaje,

$$L_2 = \{\omega\#\omega^r | w \in (a|b)^*\}$$

Ejercicio 2

Construir un AP que reconozca el siguiente lenguaje,

$$L_2 = \{\omega\#\omega^r | w \in (a|b)^*\}$$

Ejemplos de cadenas:

$\alpha_1 = \#$ ✓ $\alpha_2 = \lambda$ × $\alpha_3 = \text{a\#a}$ ✓ $\alpha_4 = \text{ab\#ba}$ ✓ $\alpha_5 = \text{abba}$ ×

Ejercicio 2

Construir un AP que reconozca el siguiente lenguaje,

$$L_2 = \{\omega\#\omega^r | w \in (a|b)^*\}$$

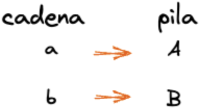
Ejemplos de cadenas:

$\alpha_1 = \#$ ✓ $\alpha_2 = \lambda$ × $\alpha_3 = \text{a\#a}$ ✓ $\alpha_4 = \text{ab\#ba}$ ✓ $\alpha_5 = \text{abba}$ ×

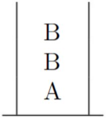
Las pilas funciona de forma **LIFO** (**L**ast **I**n, **F**irst **O**ut)

Solución ejercicio 2

Dado que nuestro alfabeto de entrada es $\Sigma = \{a, b\}$ en este caso tomaremos a $\Gamma = \{A, B, Z_0\}$ donde la representación será,



Vamos a aprovechar que la pila es LIFO para leer la reversa de una cadena apilada. Si $\alpha = \text{abb}$, entonces en la pila tenemos:



Solución ejercicio 2

Estrategia para construir el autómata de pila:

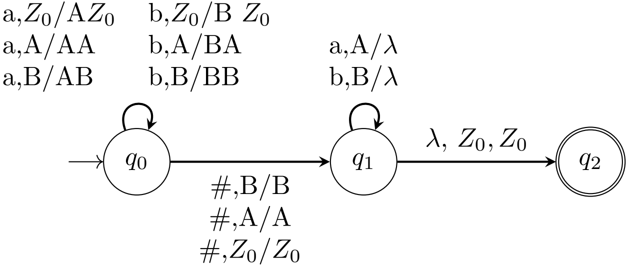
- Paso 1: Leemos ω y apilamos
- Paso 2: Leemos # sin tocar la pila
- Paso 3: Desapilamos verificando que coincida con ω^r

Solución ejercicio 2

Estrategia para construir el autómata de pila:

- Paso 1: Leemos ω y apilamos
- Paso 2: Leemos # sin tocar la pila
- Paso 3: Desapilamos verificando que coincida con ω^r

δ_2 :



$$M_2 = \langle \{q_0, q_1, q_2\}, \{a, b\}, \{A, B, Z_0\}, \delta, q_0, Z_0, \{q_2\} \rangle$$

Configuración instantánea

En un autómata finito, teníamos como configuración instantánea un elemento de:

$$Q \times \Sigma^*$$

donde q es el estado actual y es lo que resta por consumir de la cadena de entrada.

¿Cómo será una configuración instantánea en un autómata de pila?

Configuración instantánea

En un autómata finito, teníamos como configuración instantánea un elemento de:

$$Q \times \Sigma^*$$

donde q es el estado actual y es lo que resta por consumir de la cadena de entrada.

¿Cómo será una configuración instantánea en un autómata de pila?

$$(q, \alpha, \gamma) \in Q \times \Sigma^* \times \Gamma^*$$

donde:

- q es el estado actual
- α es la cadena de entrada que resta consumir
- γ es el contenido de la pila

Lenguaje aceptado

Relación de transición entre configuraciones:
$\forall q_1, q_2 \in Q$, $a \in \Sigma$, $\alpha \in \Sigma^*$, $b \in \Gamma$, $\beta \in \Gamma^*$:

- $(q_1, a\alpha, b\gamma) \vdash (q_2, \alpha, \beta\gamma) \iff (q_2, \beta) \in \delta(q_1, a, b)$
- $(q_1, \alpha, b\gamma) \vdash (q_2, \alpha, \beta\gamma) \iff (q_2, \beta) \in \delta(q_1, \lambda, b)$

Notar que *siempre* se saca el tope de la pila. Si en una transición no se quiere modificar la pila hay que volver a apilar el mismo símbolo.

Lenguaje aceptado por estado final

Lenguaje aceptado por estado final:

$$\alpha \in L(M) \Longleftrightarrow \exists q_f \in F, \gamma \in \Gamma^* \mid (q_0, \alpha, Z_0) \vdash^* (q_f, \lambda, \gamma)$$

Notar qué en este caso nos importa en que estado termina y lo que contiene la pila nos deja de importar.

Ejercicio 3

Exhibir un autómata de algún tipo que reconozca L_3, indicando claramente este tipo.

$$L_3 = \{b^n\alpha \mid \alpha \in \{a,b\}^* \ \wedge \ n = \text{número de apariciones de } a \text{ en } \alpha\}.$$

Solución del ejercicio 3

Cadenas de ejemplos:

- b abb ✓
- a ×
- bb bbbaba ✓
- b a ✓
- b bbba ✓
- b aab ×

Solución del ejercicio 3

Notar que una misma cadena puede ser aceptada o rechazada según cómo se divida:

- $\underbrace{b}_{b^1}\underbrace{abb}_{\alpha}$ ✓
- $\underbrace{bb}_{b^2}\underbrace{bbbaba}_{\alpha}$ ✓
- $\underbrace{\lambda}_{b^0}\underbrace{a}_{\alpha}$ ×
- $\underbrace{bb}_{b^2}\underbrace{bba}_{\alpha}$ ×
- $\underbrace{b}_{b^1}\underbrace{bbba}_{\alpha}$ ✓
- $\underbrace{\lambda}_{b^0}\underbrace{b}_{\alpha}$ ✓
- $\underbrace{b}_{b^1}\underbrace{\lambda}_{\alpha}$ ×

Debemos aceptar si hay alguna forma de separarla que la haga pertenecer al lenguaje.

Solución del ejercicio 3

- Paso 1: Leer b y contarlas (apilar I)
- Paso 2: ¿?
- Paso 3: Leer α y desapilar por cada aparición de a

Solución ejercicio 3

¿Como sabemos que terminamos de leer bs y comenzamos a leer α?

Solución ejercicio 3

¿Como sabemos que terminamos de leer bs y comenzamos a leer α?
No lo sabemos, hay que predecirlo.

Solución ejercicio 3

¿Como sabemos que terminamos de leer bs y comenzamos a leer α?
No lo sabemos, hay que predecirlo.

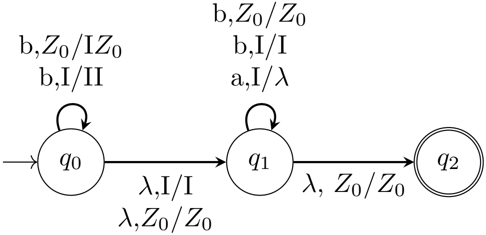
No serán 14.000.605 alternativas, pero dependiendo del autómata pueden ser muchas.

Solución del ejercicio 3

- Paso 1: Leer b y contarlas (apilar I)
- Paso 2: Decidir no deterministicamente que terminé de consumir b y comenzar a leer α
- Paso 3: Leer α y desapilar por cada aparición de a

Solución del ejercicio 3

Dado el siguiente autómata
δ_3 :

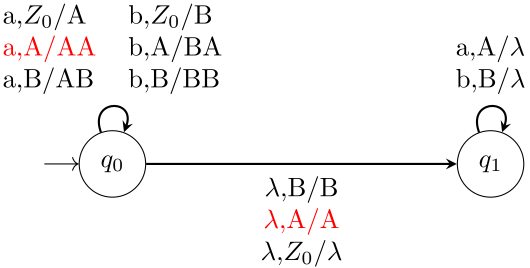


$$M_3 = \langle \{q_0, q_1, q_2\}, \{a, b\}, \{I, Z_0\}, \delta, q_0, Z_0, \{q_2\} \rangle$$

Autómata deterministico

Un autómata de pila es determinístico si $\forall q \in Q, z \in \Gamma, a \in \Sigma$ vale:

- $|\delta(q, a, z)| \leq 1$
- $|\delta(q, \lambda, z)| \leq 1$
- $|\delta(q, \lambda, z)| = 1 \Longrightarrow |\delta(q, a, z)| = 0$



ítem 1 ✓ ítem 2 ✓ ítem 3 ×

¿Nos tomamos un break?

Lema ("Pumping" para lenguajes libres de contexto.)

Para todo lenguaje L libre de contexto, existe $n > 0$ tal que para toda cadena $\alpha \in L$ con $|\alpha| \geq n$,

- Existe una descomposición de α en cadenas r, x, y, z, s, es decir $\alpha = rxyzs$.
- $|xyz| \leq n$
- $|xz| \geq 1$
- Para todo $i \geq 0$, la cadena rx^iyz^is pertenece a L

Ejercicio 4

El siguiente lenguaje no es libre de contexto. Demostrarlo usando Pumping.

$$L_4 = \{a^k b^k c^k | k \geq 0\}$$

Ejercicio 4

El siguiente lenguaje no es libre de contexto. Demostrarlo usando Pumping.

$$L_4 = \{a^k b^k c^k | k \geq 0\}$$

Demostración

Supongamos L_4 es libre de contexto. Sea n la constante del Lema de Pumping.

$$\text{Sea } \alpha = a^n b^n c^n.$$

Ejercicio 4

El siguiente lenguaje no es libre de contexto. Demostrarlo usando Pumping.

$$L_4 = \{a^k b^k c^k | k \geq 0\}$$

Demostración

Supongamos L_4 es libre de contexto. Sea n la constante del Lema de Pumping.

$$\text{Sea } \alpha = a^n b^n c^n.$$

- Por el Lema de Pumping, hay una descomposición $\alpha = rxyzs$ con $|xyz| \leq n$ y $|xz| \geq 1$ tal que $rx^i yz^i s$ en L_4, para todo $i \geq 0$.

Ejercicio 4

El siguiente lenguaje no es libre de contexto. Demostrarlo usando Pumping.

$$L_4 = \{a^k b^k c^k | k \geq 0\}$$

Demostración

Supongamos L_4 es libre de contexto. Sea n la constante del Lema de Pumping.

$$\text{Sea } \alpha = a^n b^n c^n.$$

- Por el Lema de Pumping, hay una descomposición $\alpha = rxyzs$ con $|xyz| \leq n$ y $|xz| \geq 1$ tal que $rx^i yz^i s$ en L_4, para todo $i \geq 0$.
- Dado que $|xyz| \leq n$, xyz no puede tener as, bs y cs. Como $|xz| \geq 1$, xz tiene alguna de las tres letras, pero **no las tres**.

Ejercicio 4

El siguiente lenguaje no es libre de contexto. Demostrarlo usando Pumping.

$$L_4 = \{a^k b^k c^k | k \geq 0\}$$

Demostración
Supongamos L_4 es libre de contexto. Sea n la constante del Lema de Pumping.

$$\text{Sea } \alpha = a^n b^n c^n.$$

- Por el Lema de Pumping, hay una descomposición $\alpha = rxyzs$ con $|xyz| \leq n$ y $|xz| \geq 1$ tal que $rx^i yz^i s$ en L_4, para todo $i \geq 0$.
- Dado que $|xyz| \leq n$, xyz no puede tener as, bs y cs. Como $|xz| \geq 1$, xz tiene alguna de las tres letras, pero **no las tres**.
- Por Lema de Pumping, $rx^0 yz^0 s = rys$ está en L_4.

Ejercicio 4

El siguiente lenguaje no es libre de contexto. Demostrarlo usando Pumping.

$$L_4 = \{a^k b^k c^k | k \geq 0\}$$

Demostración
Supongamos L_4 es libre de contexto. Sea n la constante del Lema de Pumping.

$$\text{Sea } \alpha = a^n b^n c^n.$$

- Por el Lema de Pumping, hay una descomposición $\alpha = rxyzs$ con $|xyz| \leq n$ y $|xz| \geq 1$ tal que $rx^i yz^i s$ en L_4, para todo $i \geq 0$.
- Dado que $|xyz| \leq n$, xyz no puede tener as, bs y cs. Como $|xz| \geq 1$, xz tiene alguna de las tres letras, pero **no las tres**.
- Por Lema de Pumping, $rx^0 yz^0 s = rys$ está en L_4.

Pero esto es imposible

Ejercicio 4

El siguiente lenguaje no es libre de contexto. Demostrarlo usando Pumping.

$$L_4 = \{a^k b^k c^k | k \geq 0\}$$

Demostración
Supongamos L_4 es libre de contexto. Sea n la constante del Lema de Pumping.

$$\text{Sea } \alpha = a^n b^n c^n.$$

- Por el Lema de Pumping, hay una descomposición $\alpha = rxyzs$ con $|xyz| \leq n$ y $|xz| \geq 1$ tal que $rx^i yz^i s$ en L_4, para todo $i \geq 0$.
- Dado que $|xyz| \leq n$, xyz no puede tener as, bs y cs. Como $|xz| \geq 1$, xz tiene alguna de las tres letras, pero **no las tres**.
- Por Lema de Pumping, $rx^0 yz^0 s = rys$ está en L_4.

Pero esto es imposible

- Ya que xy no tiene la misma cantidad de las tres letras por lo tanto rys no tiene la misma cantidad de as, bs y cs.

Ejercicio 4

El siguiente lenguaje no es libre de contexto. Demostrarlo usando Pumping.

$$L_4 = \{a^k b^k c^k | k \geq 0\}$$

Demostración

Supongamos L_4 es libre de contexto. Sea n la constante del Lema de Pumping.

Sea $\alpha = a^n b^n c^n$.

- Por el Lema de Pumping, hay una descomposición $\alpha = rxyzs$ con $|xyz| \leq n$ y $|xz| \geq 1$ tal que $rx^i yz^i s$ en L_4, para todo $i \geq 0$.
- Dado que $|xyz| \leq n$, xyz no puede tener as, bs y cs. Como $|xz| \geq 1$, xz tiene alguna de las tres letras, pero **no las tres**.
- Por Lema de Pumping, $rx^0 yz^0 s = rys$ está en L_4.

Pero esto es imposible

- Ya que xy no tiene la misma cantidad de las tres letras por lo tanto rys no tiene la misma cantidad de as, bs y cs.

Llegamos a esta contradicción pues partimos de que L_4 era libre de contexto. Entonces concluimos que no lo es.

Ejercicio 5

El siguiente lenguaje no es libre de contexto. Demostrarlo usando Pumping.

$$L_5 = \{0^i 1^j 2^i 3^j | k \geq 1, i \geq 1\}$$

Ejercicio 5

El siguiente lenguaje no es libre de contexto. Demostrarlo usando Pumping.

$$L_5 = \{0^i 1^j 2^i 3^j | k \geq 1, i \geq 1\}$$

Demostración
Supongamos L_5 es libre de contexto. Sea n la constante del Lema de Pumping.

$$\text{Sea } \alpha = 0^n 1^n 2^n 3^n.$$

Ejercicio 5

El siguiente lenguaje no es libre de contexto. Demostrarlo usando Pumping.

$$L_5 = \{0^i 1^j 2^i 3^j | k \geq 1, i \geq 1\}$$

Demostración
Supongamos L_5 es libre de contexto. Sea n la constante del Lema de Pumping.

$$\text{Sea } \alpha = 0^n 1^n 2^n 3^n.$$

- Por el Lema de Pumping, hay una descomposición $\alpha = rxyzs$ con $|xyz| \leq n$ y $|xz| \geq 1$ tal que $rx^i yz^i s$ en L_5, para todo $i \geq 0$.

Ejercicio 5

El siguiente lenguaje no es libre de contexto. Demostrarlo usando Pumping.

$$L_5 = \{0^i 1^j 2^i 3^j | k \geq 1, i \geq 1\}$$

Demostración
Supongamos L_5 es libre de contexto. Sea n la constante del Lema de Pumping.

$$\text{Sea } \alpha = 0^n 1^n 2^n 3^n.$$

- Por el Lema de Pumping, hay una descomposición $\alpha = rxyzs$ con $|xyz| \leq n$ y $|xz| \geq 1$ tal que $rx^i yz^i s$ en L_5, para todo $i \geq 0$.
- Dado que $|xyz| \leq n$, xyz no puede tener $0s$, $1s$, $2s$ y $3s$. Como $|xz| \geq 1$, xz tiene alguno de los números.

Ejercicio 5

El siguiente lenguaje no es libre de contexto. Demostrarlo usando Pumping.

$$L_5 = \{0^i 1^j 2^i 3^j | k \geq 1, i \geq 1\}$$

Demostración
Supongamos L_5 es libre de contexto. Sea n la constante del Lema de Pumping.

$$\text{Sea } \alpha = 0^n 1^n 2^n 3^n.$$

- Por el Lema de Pumping, hay una descomposición $\alpha = rxyzs$ con $|xyz| \leq n$ y $|xz| \geq 1$ tal que $rx^i yz^i s$ en L_5, para todo $i \geq 0$.
- Dado que $|xyz| \leq n$, xyz no puede tener $0s$, $1s$, $2s$ y $3s$. Como $|xz| \geq 1$, xz tiene alguno de los números.

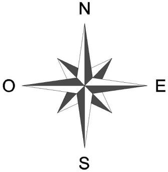
Ejercicio 5

- Si xyz consta de un solo símbolos entonces rys tiene 3 símbolos diferentes y por lo tanto la potencia a la cual elevemos a x y z va a alterar el exponente del símbolo de xyz dejando a α no perteneciente a L_5
- Si xyz consta de dos símbolos entonces, digamos 1s y 2s, entonces rys le faltan algunos 1s y 2s y por lo tanto α no perteneciente a L_5

Llegamos a esta contradicción pues partimos de que L_5 era libre de contexto. Entonces concluimos que no lo es.

Ejercicio extra de la práctica

Dado el alfabeto $\{ \uparrow, \leftarrow, \downarrow, \rightarrow \}$ podemos interpretar una cadena como una serie de pasos a dar sobre una cuadrícula. Por ejemplo, siguiendo la cadena $\leftarrow\leftarrow\leftarrow\downarrow$, terminamos tres pasos al oeste y un paso al sur del lugar donde comenzamos. Sea L_P el lenguaje de las cadenas que terminan dos pasos al norte del punto inicial (sin importar cuántos pasos al este o al oeste), y en las que un paso al sur nunca es inmediatamente seguido por un paso al este. Por ejemplo, $\downarrow\uparrow\rightarrow\uparrow\rightarrow\uparrow$ es una cadena de L_P, mientraS que $\rightarrow\downarrow\rightarrow$ y $\uparrow\downarrow\rightarrow\uparrow\uparrow$ no lo son. Dar un autómata de pila que reconozca L_P. ¿Es un autómata determinístico?



Solución del ejercicio extra de la práctica

Ejemplos de cadenas:

$$\alpha_1 = \uparrow\uparrow\uparrow\downarrow\rightarrow \checkmark \quad \alpha_2 = \rightarrow\uparrow\uparrow\downarrow \times \quad \alpha_3 = \uparrow\downarrow\leftarrow\uparrow\uparrow \checkmark$$

El alfabeto que vamos a usar es:

Idea : podemos empezar apilando ss entonces cuando lleguemos a Z_0 es porque ya mos hemos movido dos pasos al norte.

Solución del ejercicio 4 de la práctica

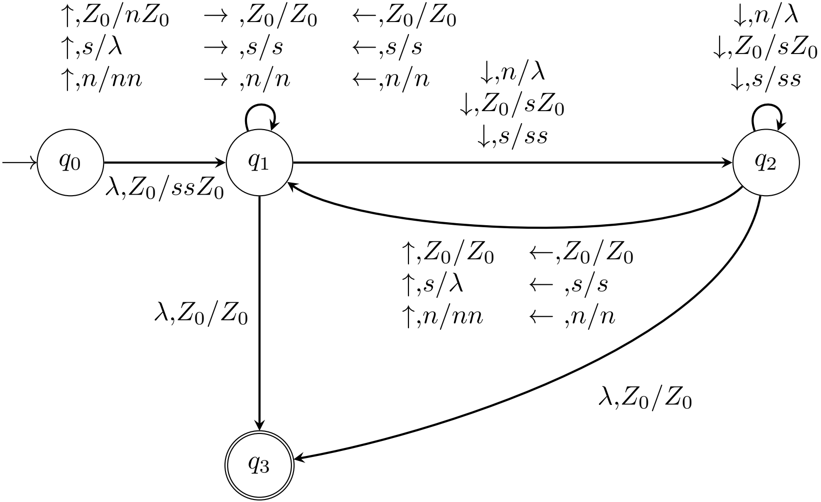
- Paso 1: Apilamos ss (ver idea)
- Paso 2: Por cada $\uparrow$:
  - Si hay una s en el tope de la pila entonces desapilo
  - Si hay una n en el tope de la pila entonces apilo una n
  - Si hay Z_0 en el tope de la pila entonces apilo nZ_0

  Por cada $\downarrow$:
  - Si hay una s en el tope de la pila entonces apilo una s
  - Si hay una n en el tope de la pila entonces desapilo
  - Si hay Z_0 en el tope de la pila entonces apilo sZ_0
  - No le puede seguir un $\rightarrow$

  Por cada $\leftarrow$ o $\rightarrow$:
  - Dejo la pila igual
- Paso 3: Acepto si termine de consumir toda la cadena y en la pila hay Z_0

Solución del ejercicio extra de la práctica

δ_P :



$$M_P = \langle \{q_0, q_1, q_2, q_3\}, \{a, b\}, \{n, sZ_0\}, \delta_P, q_0, Z_0, \{q_3\} \rangle$$

Cosas importantes

- LIC (tipo 2) tienen mayor poder expresivo que los LR, pueden contar ciertas cosas
- Los autómatas de pila por estado final aceptan una cadena si la misma se puede consumir en su totalidad y al finalizar se llega a un estado final, sin importar el estado de la pila

Cosas importantes

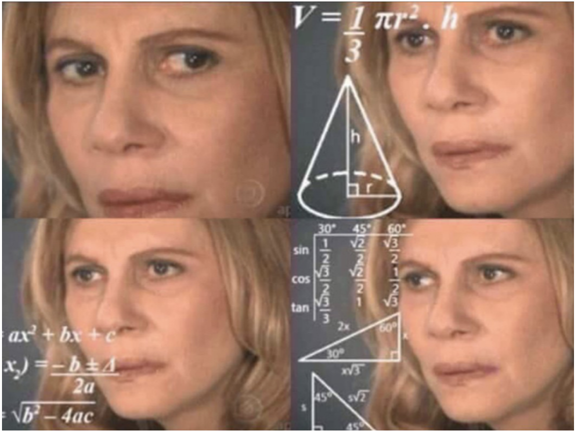
- Los lenguajes generados por APND tienen mayor poder expresivo que los APD
- Los lenguajes generados por APND por estado final tienen el mismo poder expresivo que los APND por pila vacía
- Los lenguajes generados por APD por estado final tienen mayor poder expresivo que los APD por pila vacía
- Los lenguajes generados por APD por pila vacía son libres de prefijos

Bibliografía

Hopcroft J.E., Motwani R., Ullman J.D (2007). Introduction to Automata Theory, Languages, and Computation, 3rd edition. Addison Wesley. Capitulo 1 y 6.

Aho A.V. Ullman J.D. (1971). The Theory of Parsing, Translating and Compiling, Volume 1: Parsing. Prentice-Hall. Capitulo 2, sección 5.

¿Preguntas?



V = 1/3 πr². h
h
r
30° 45° 60°
sin
cos
tan
ax² + bx + c
2a
√b² – 4ac
2x
60°
30°
x√3
45°
s√2
s
45°