2017

Apuntes del 26 de abril

COMPILADORES E INTÉRPRETES

XIMENA BOLAÑOS FONSECA.
2015073844

INSTITUTO TECNOLÓGICO DE COSTA RICA | Sede de San José

Recordatorios

Recordar que ya está en el foro el último desglose con el examen.

Repaso

Antes del examen se vio lo que es el principio del palomar y ¿Qué tiene que ver eso con toda la materia? Eso es lo que vamos a ver a continuación.

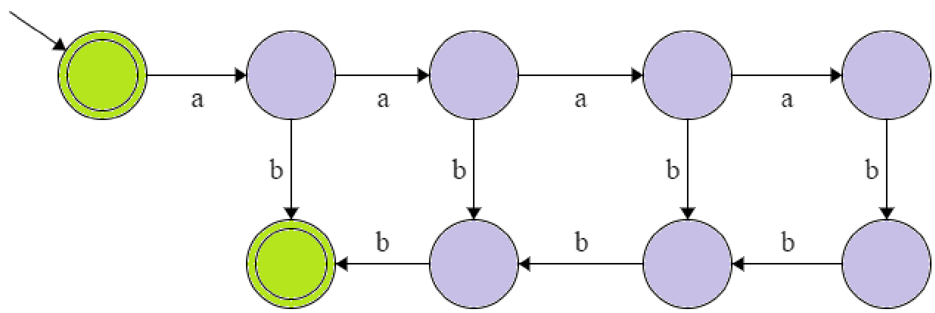
¿Para que servía el Pumping Lemma? Para decir que un lenguaje no es regular.

¿Cómo se prueba que es un Lenguaje Regular? Si se tiene un DFA que lo pueda representar.

Análisis Léxico

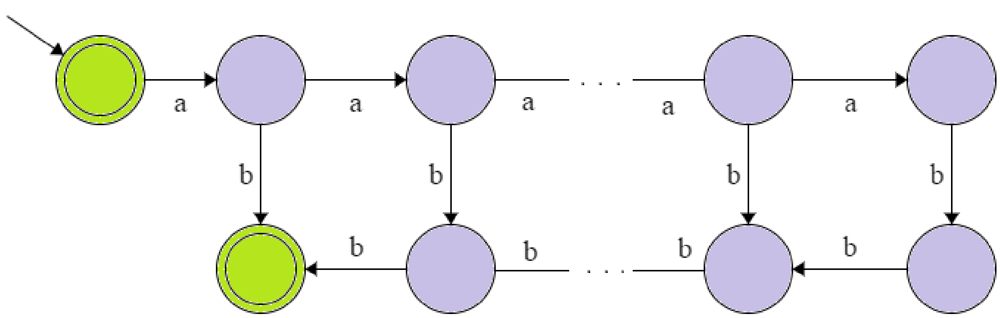
Un Lenguaje Regular

Sea $\mathcal{L}$ el lenguaje sobre ∑= {a, b} definido como $\{a^n b^n | n \leq 4\}$, es decir que tenga la misma cantidad de a's y de b's. Diseñe un DFA que reconozca a $\mathcal{L}$.



Otro lenguaje

Sea L el lenguaje sobre ∑= {a, b} definido como $\{a^n b^n | n \geq 0\}$.



¡NO!, no se puede resolver así no es regular porque es infinito y no se puede contar ya que no sé cuántas a´s para saber cuánto regresarse para cuantas b´s. Por lo que no hay un DFA que lo reconozca.

Comentario

Díganme que todos odian esta puerta de vidrio.

Opciones para arreglar el asunto de la Puerta:

- *Podemos traer un cartón de refrigerado.*
- *NO estudien computación.*
- *Todos estamos Muertos. He Walking Dead.*

Lenguajes no regulares

- Existen lenguajes no regulares. Hasta ahora se conoce 1.
- No existe un DFA que lo reconozca.
- Hay que demostrar que no existe un DFA
- Pumping Lemma (por las bombas de sacar agua).

Aceptando una Hilera de Longitud N

- Se sabe que al reconocer una hilera de n símbolos el autómata va a tener n+1 estados.
- Sea M= (Q, ∑, δ, q_0, F) un DFA.
- Sea w=w_1, w_2, ..., w_n una hilera de longitud n sobre ∑.
- M acepta w si existe una sección de estados r_0, r_1, r_2, ..., r_n tal que:
    1. Todos los r_i pertenecen a Q
    2. r_0=q_0
    3. δ (r_i, wi_{+1}) =r_{i+1} para i=0, ..., n-1
    4. $r_n \in F$

Lenguajes Finitos e Infinitos.

- Todo lenguaje finito es regular(trivial): siempre se puede dibujar un DFA que lo reconozca.
- Si un lenguaje fuera "no regular", debe ser infinito.
- Nos interesan los lenguajes infinitos.
- Al ser infinitos hay hileras de longitud arbitraria.
- Hay hileras de longitud infinitas.

Lenguajes Regulares Infinitos.

- Sea $\mathcal{L}$ un lenguaje regular infinito.
- Con $\mathcal{L}$ es infinito, tiene hileras de longitud arbitraria.
- Como $\mathcal{L}$ es regular, existe un DFA que lo reconoce.
- Este DFA tiene k estados (k finito). ¿Puede haber hileras más grandes que la cantidad de estados del autómata? Sí.
- ¿Existen hileras en L de longitud n ≥ k?
- Sí, porque L es infinito.

Aceptando una hilera de longitud n ≥ k

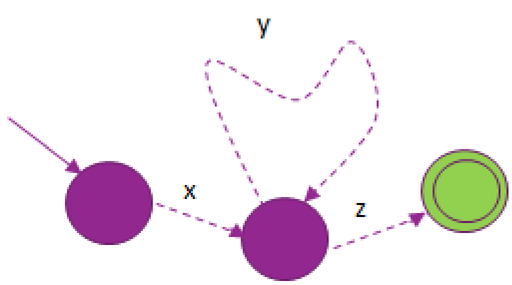
- Sean $\mathcal{L}$ un lenguaje regular infinito reconocido por un DFA de k estados
- Sea w ∈ $\mathcal{L}$, una hilera tal que |w| tal que n ≥ k
- Al procesar (y aceptar) a w con el DFA que reconoce a $\mathcal{L}$, se visitarán n+1 estados
- Ya que n+1>k, entonces se visita por lo menos 2 veces un mismo estado.
- ¿Por qué?
- Principio del Palomar.

Hileras de longitud n ≥ k y Ciclos en DFA

- Sean L un lenguaje regular infinito reconocido por un DFA de k estados
- Sea w ∈ $\mathcal{L}$, una hilera tal que |w| tal que n ≥ k
- Ya que w ∈ $\mathcal{L}$ hay una ruta en el DFA que va desde el estado inicial, pasa por estados, hasta un estado de aceptación consumiendo los símbolos de w.
- Esta ruta visita por lo menos 2 veces un mismo estado
- Por lo tanto:

> Esta ruta siempre contiene por lo menos un ciclo en el grafo del DFA que reconoce a $\mathcal{L}$.

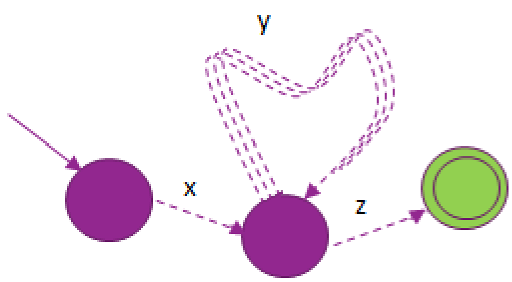
- A procesar a w ε L la ruta tiene un ciclo si |w|≥k. Por lo que por principio del palomar pasa por lo menos por un estado 2 veces.
- Hay 3 partes en esta ruta que corresponden a trazos consecutivos de w.
  - Un pedazo de la hilera pasa del estado inicial, que sería x.
  - y son los ciclos que se hacen en medio
  - z es el que lleva al estado final de aceptación.
  - w=xyz ∈ $\mathcal{L}$



Aquí viene la magia.

- ¿Qué otra hilera pertenece a esto?
- ¡xy!
*-*El profe va a escribir* No, espere porque le hago caso.*
- ¡xz! ¡xz!
-Comencemos de nuevo.
- ¿Qué otra hilera pertenece a esto?
- ¡xz!
- ¡WOW! ¡Que genial idea!

- Podemos quitar el estado “y” y aun así llegar a un estado de aceptación
  - $w=xz \in \mathcal{L}$
- Podemos quitar el ciclo “y” y llegar a un estado de aceptación
  - $w=xyyyz \in \mathcal{L}$
  - $w=xy^iz \in \mathcal{L}$ $(i\geq0)$ “*bombeen y”* ¡Pumping Lemma!
  - $w=xyz \in \mathcal{L}$



Pumping Lemma

Sea $\mathcal{L}$ un lenguaje regular reconocido por un DFA se k estados. Sea $w \in L$, una hilera tal que $|w|\geq k$. Entonces w puede ser separad en tres piezas $w=xyz$ talque se cumple que:

1. $|y|>0$
2. $|xy|\leq k$
3. $xy^iz \in L$, para todo $i\geq0$

Todo lenguaje regular infinito satisface el Pumping Lemma.

- ¿Profe, ya pregunto las preguntas aquellas?

-Paciencia.

¡Aprecien el Pumping Lemma! ¡Es hermoso!

Ejemplo

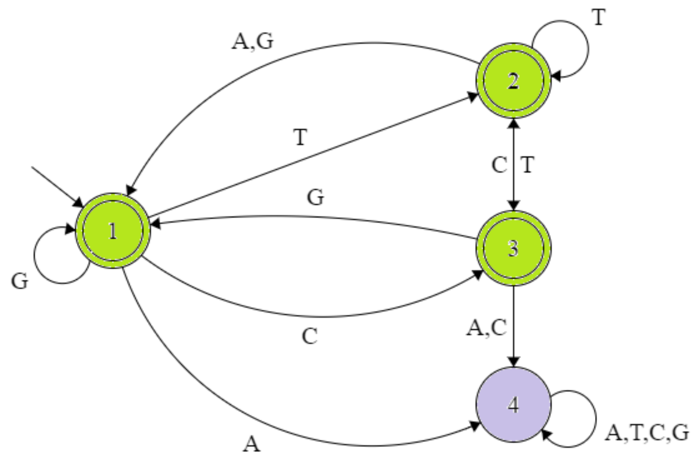
Sea L el lenguaje sobre ∑= {A, T, C, G} de hilera que no contenga la sub hilera “CC” y donde

La hilera “TCTAGGC” $\in \mathcal{L}$ y tiene 7 símbolos (7>4).

Al procesarla se repite el estado 2 antes del cuarto símbolo.

$\underbrace{T}_{x}\ \underbrace{CT}_{y}\ \underbrace{AGGC}_{z}$

(pedazo mágico)



También pertenece a L:

- TCTCTAGGC
- TCTCTCTCTCTAGGGC
- TCTCTCTCTCTCTCTCTCTCTCTCTCTCTCTAGGC

Comentario

- Hey, ahora que está ahí. Ponga un pupitre ahí en la puerta y se sienta.

Se pone un pupitre. Hacen mucho ruido. Sale el profe y los va a callar.

-Igual no se pueden sentar ahí.

-Vaya regáñelos.

Tigres y Rayas

- Supongamos que todos los tigres tienen rayas.

- Si alguien nos describe un animal que no estábamos viendo y nos dice.

“Tiene rayas”

“Es un tigre”

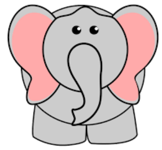
No, es una cebra.

- Si alguien nos dice un animal que no estábamos viendo y nos dice.

  "No tiene rayas"

  ¿Qué podemos afirmar?

  No es un tigre.



- El punto:

  Si un lenguaje no cumple el Pumping Lemma no es regular, ya que el Pumping Lemma hace que se quiebre.

Comentario

-No usen la palabra ciclo usen la palabra Pumping Lemma.

Sí el Pumping Lemma se cumple no implica que sea regular.

Tigres, Rayas, Lenguajes Regulares y Pumping Lemma.

- Si alguien describe un lenguaje y nos dice:
  - No satisface el Pumping Lemma.
  - ¿Qué se puede afirmar?
    - No es regular. *Es un elefante*.
- Si alguien nos describe un lenguaje y nos dice que:
  - Satisface el Pumping Lemma
  - No se puede afirmar que sea regular.

¿Cómo se usa el Pumping Lemma?

Queremos demostrar por contradicción que $\mathcal{L}$ no es un lenguaje regular. ¿Cómo? "bombeando".

1. Suponemos que $\mathcal{L}$ es regular y hay un DFA con k estado que lo reconoce.
2. Nos damos cuenta que una hilera w $\in \mathcal{L}$ cuya longitud sea > k.
3. Demostramos que hace el "pump up" o "pump down" para cualquier sub hilera y dentro de w tal como se describe en el Pumping Lemma, se genera al menos una hilera que no pertenece a $\mathcal{L}$.
4. Encontramos que $\mathcal{L}$ no satisface el Pumping Lemma.
5. Concluimos que $\mathcal{L}$ no es regular.

**Encontrar la hilera es lo más importante.

Ejemplo 1

**Siempre que se sepa dónde está y, bombear hace que falle el Pumping Lemma.

Sea $\mathcal{L}$ el lenguaje sobre $\sum$= {0,1} definido como $\{0^n1^n | n \geq 0\}$

Demuestre por contradicción que L no es regular

- Suponemos que $\mathcal{L}$ es regular y que es reconocido por un DFA de k estados
- Sea $w=0^k1^k$

$$\overbrace{\underbrace{000}_{x}\underbrace{000...000}_{y}}^{k}\overbrace{\underbrace{111111...111}_{z}}^{k}$$

- Según el Pumping Lemma w puede separarse en xyz.
- De acuerdo a las reglas “y” estará formado solo por 0’s.
- Problema: si “bombeamos y” 2 o más veces generamos una hilera con más 0´s que 1’s, y que no pertenece a $\mathcal{L}$
- $\mathcal{L}$ no cumple el Pumping Lemma, entonces $\mathcal{L}$ no es regular.

Ejemplo 2

- Sea $\mathcal{L}$ el lenguaje sobre ∑= {0,1} de hileras con las mismas cantidades de 0’s y de 1’s.
  ****Es distinto al lenguaje anterior, pero el anterior es un subconjunto de este y al ser no regular este no es regular.**
  ****w nosotros lo escogemos, con mucho cuidado, como los puercoespines…**

  - Reconoce:
    - 10001
    - ε
- 0^k1^k Como se demostró anteriormente no pertenece al alfabeto.
- Demuestre por contradicción que $\mathcal{L}$ no es regular
  1. $\mathcal{L}$ es Regular.
  2. Suponemos que $\mathcal{L}$ es reconocida por un DFA de k estados.
  3. Sea w =0^k1^k

$$\overbrace{\underbrace{000}_{x}\underbrace{000\ldots000}_{y}}^{k}\overbrace{\underbrace{111111\ldots111}_{z}}^{k}$$

- Según el Pumping Lemma w puede separarse en xyz.
- De acuerdo a las reglas y estará formado solo por 0’s.
- Problema: si “bombeamos y” 2 o más veces generamos una hilera con más 0´s que 1’s, y que no pertenece a $\mathcal{L}$
- $\mathcal{L}$ no cumple el Pumping Lemma, entonces $\mathcal{L}$ no es regular.

Ejemplo 3

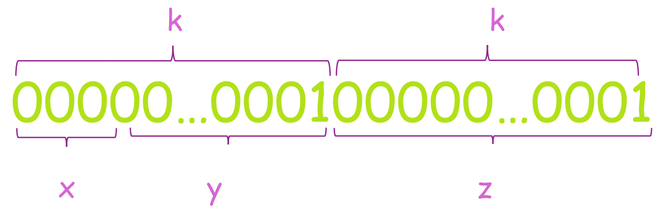
Continua la historia de la puerta:

-La teoría del chorizo

Que se levanten firmas. La de madera era mejor.

-La de madera no es caliente. :(

- Sea $\mathcal{L}$= {ss | s ∊ {0,1} *}
- Demuestre por contradicción que $\mathcal{L}$ no es regular:
  1. $\mathcal{L}$ es regular
  2. Hay un DFA que reconoce el lenguaje $\mathcal{L}$ con k estados
  3. w=0^k10^k1 sigue siendo parte del lenguaje, ya que este ejemplo corre el 1 y hace que no queden iguales las mitades.



- Según el Pumping Lemma w puede separarse en xyz.
- De acuerdo a las reglas y estará formado solo por 0’s.
- Problema: si “bombeamos y” 2 o más veces generamos una hilera cuya primera mitad es diferente al segundo, y que no pertenece a $\mathcal{L}$.
- $\mathcal{L}$ no cumple el Pumping Lemma, entonces $\mathcal{L}$ no es regular.

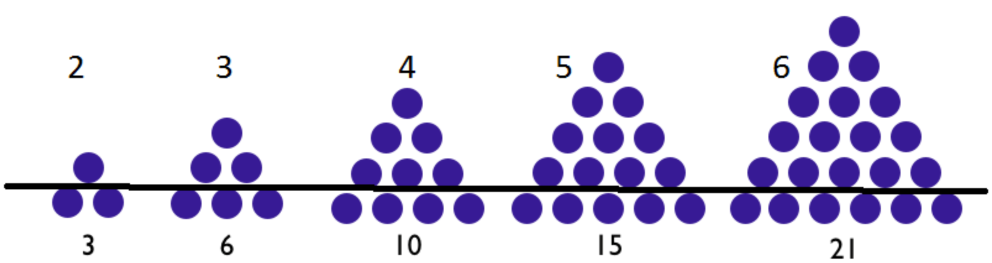
- ¡Es bello! Seguro lo veo bello porque mi vida es muy triste :(

Ejemplo 4

Sea L= {0^n|n es un cuadrado perfecto} Los números cuadrados, es porque se puede formar un cuadrado con piedritas.

Sabías que:

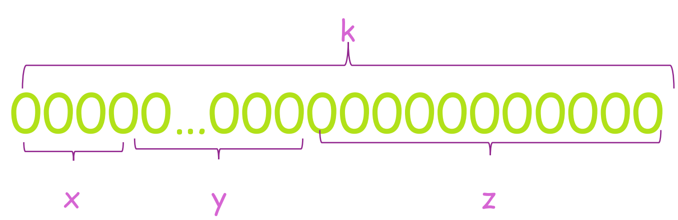
Existen también los números triangulares. Pero no fueron tan populares.



Volviendo:

Hay que demostrar que L no es regular.

1. Suponemos que $\mathcal{L}$ es regular.
2. Hay un DFA que reconoce a $\mathcal{L}$ con k estados.
3. Sea $w=0^{k\text{^}2}$
4. Nótese que w ∊ $\mathcal{L}$ y que |w|>k, entonces debiera de cumplirse el Pumping Lemma.



Con “pump up”

Tome en cuenta que:

$0^{(k+1)\,{}^{\wedge}2 => (k+1)\,{}^{\wedge}2 = k{}^{\wedge}2+2k+1}$

Se sabe que

$0<|y|\leq k$ y $|w|=k^2$

$w'=|w|+|y|$

$k^2+0<k^2+|y|\leq k^2+k$

$k^2<|w'|\leq k^2+k<k^2+2k+1$ *No es un cuadrado perfecto*.

Formalmente:

- Según el Pumping Lemma w puede separarse en xyz.
- $|xyz|=k^2$
- De acuerdo a las reglas $0<|y|\leq k$
- $|xy^2z|=k^2+|y|\leq k^2+2k+1=(k+1)^2$
- $|xy^2z|$no es un cuadrado perfecto, entonces xy^2z no pertenece a $\mathcal{L}$.
- $\mathcal{L}$ no cumple el Pumping Lemma, entonces $\mathcal{L}$ no es regular.

Ejemplo 5

****De hoy en ocho vuelven los quices.**

- Sea $\mathcal{L}$ el lenguaje sobre $\sum = \{0,1\}$ definido como $\{0^m1^n|m>n\}$
- Demuestre por contradicción que $\mathcal{L}$ no es regular.
    - Supongamos que $\mathcal{L}$ es regular.
    - Hay un DFA que reconoce a $\mathcal{L}$ con k estados.
    - Sea $w=0^{k+1}1^k$
    - Nótese que $w \in \mathcal{L}$ y que $|w|>k$, entonces debiera de cumplirse el Pumping Lemma.

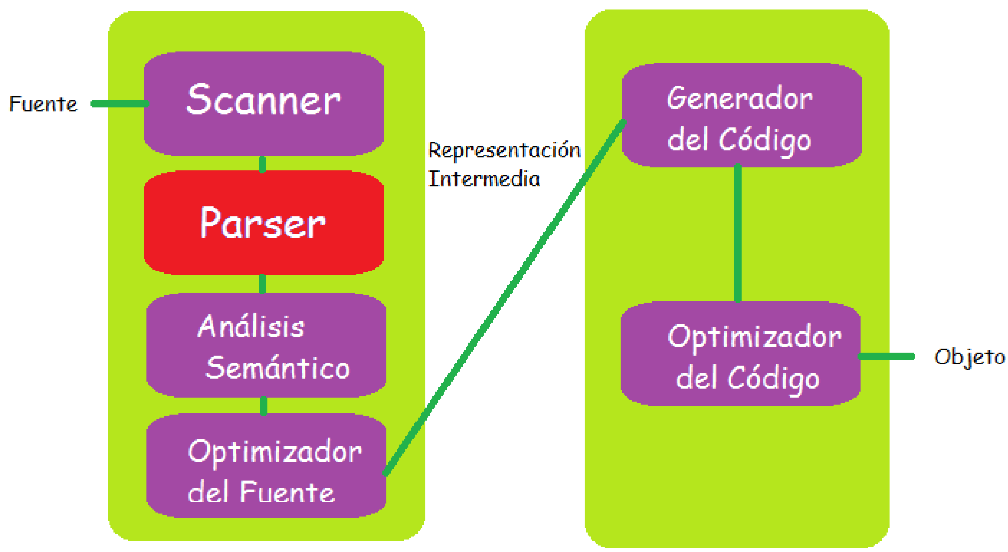
- Según el Pumping Lemma w puede separarse en xyz.
- De acuerdo a las reglas y estará formado solo por 0’s.
- Problema: si “si hacemos un pump down (i.e., i = 0) de y” generamos una hilera con menos o igual cantidad de 0’s que 1’s, y que no pertenece a $\mathcal{L}$.
- $\mathcal{L}$ no cumple el Pumping Lemma, entonces $\mathcal{L}$ no es regular.

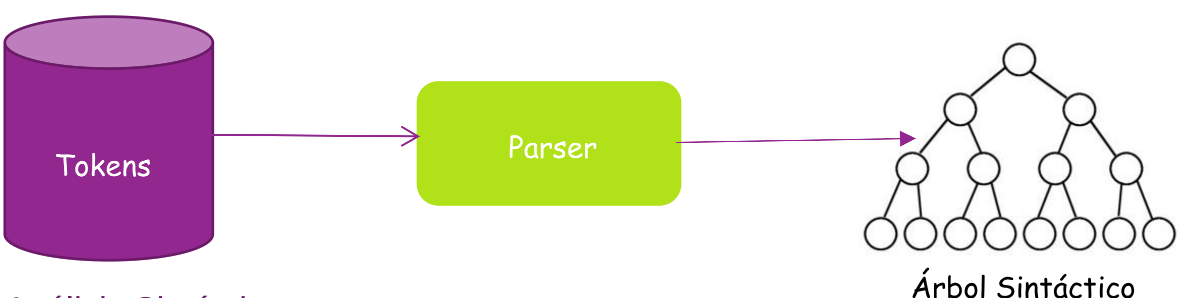
Análisis Sintáctico

Es tan malo que da risa.

Conceptos Generales

Parser





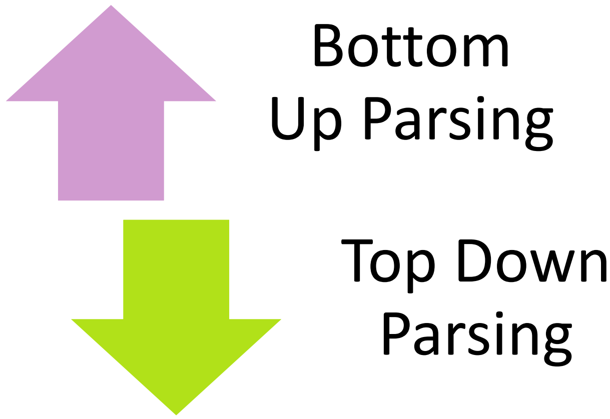
Análisis Sintáctico

- Revisa que la sintaxis de un programa esté correcta.
- Conocido como parser (Entender la estructura de algo).

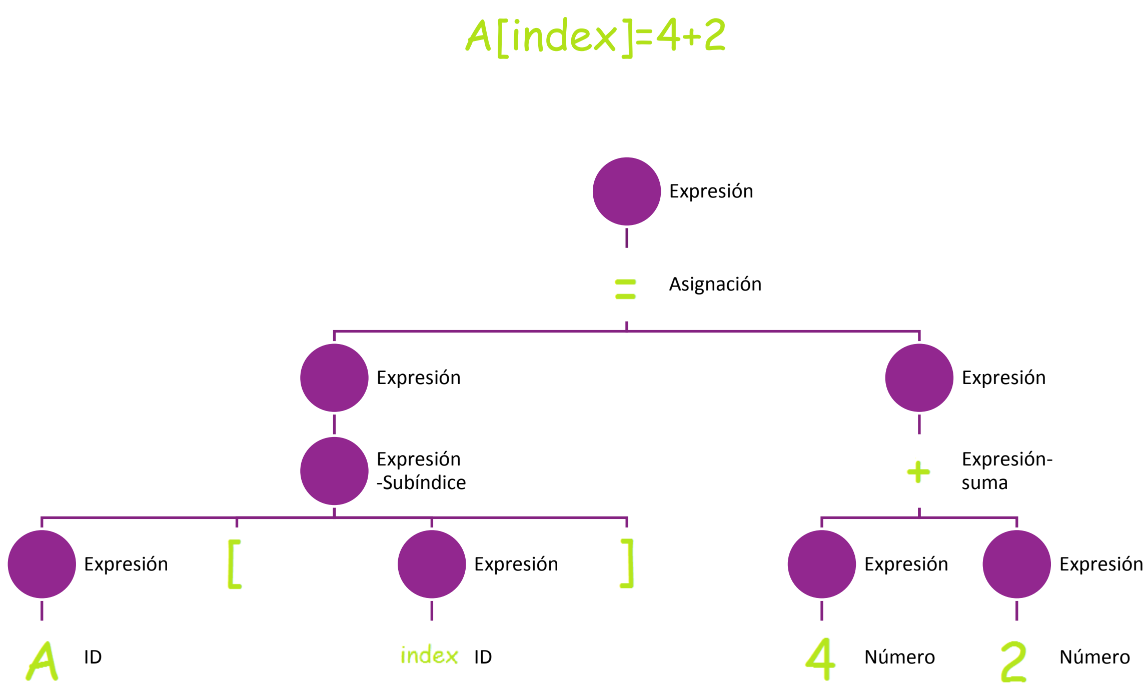
- Toma la secuencia de tokens y verifica que correspondan a constructores válidos.
- Requerimientos de una gramática.
- Hay muchos algoritmos posibles de parsing. *Buenas o malas noticias.*

Tipos de Parsing

Hay 2 grandes familias de parsers.



Árbol Sintáctico.



**Es como un domino satánico.

Se resolvió a punta de gramáticas.

Errores de Sintaxis

- El manejo de errores es muy difícil en el Análisis Sintáctico.

- En el análisis léxico es relativamente fácil.
- El parser debe hacer muchas cosas frente a un error (triple R)
    - Reportar(Report)
    - Recuperar(Recover). Poder seguir revisando más.
    - Reparar(Repair) Usted no quiere que un compilador lo repare, usted no sabe que está reparando porque puede haber muchos caminos.

Reportar

Programadores viven de esto.

- Encontrar que hay un error que es automático con los algoritmos de parsing.
- Se sabe que la hilera está mala.
    - Localización del error
        - Posición exacta del token.
          Ofensivo es difícil.
        - Puede ser la ausencia de un token.
    - Mensaje de error (es todo un arte)
        - Peor de los casos error de sintaxis.
        - Redacción apropiada es muy difícil.

Recuperar

- Con un único error, ya toda la hilera está mal todo el programa está malo.
- Es impráctico que solo se reporte el primer error.
- ¿10 mil líneas? El parser debe recuperarse para seguir analizando el resto del programa.
- Avanzar hasta “punto seguro” y continuar proceso de parsing.
- Problemas:
    - Saltar demasiado.
    - Cascada de errores. Por una mala recuperación de errores.
    - Técnicas son adhoc (macheteado, para este caso y NO, no es la marca de zapatos.)

-Que se recupere.

-No profe, no da risa.

-Váyanse malditos.