

# Práctica 2: Analizadores sintácticos.

Compiladores 2017-1

Elisa Viso Gurovich

September 8, 2016

## 1 Analizadores sintácticos descendentes (*Top-Down*)

- Ejercicios teóricos:

1. ¿Cuales són las maneras de implementar un analizador sintáctico descendente?
2. ¿Que características debe de cumplir una gramática libre del contexto para que pueden tener un reconocedor descendente recursivo sin caer en ciclos ni hacer backtrack?
3. Implementa un anlizador sintáctico para la siguiente gramática:

```
S -> Aa | b
A -> Ac | Sd | epsilon
```

No debe caer en ciclos ni hacer backtrack.

## 2 Analizadores Sintácticos ascendentes (*Bottom-UP*)

- Instalación de **Byacc/J**

1. Descargar binario de <http://byaccj.sourceforge.net/#download>
2. Instalar bibliotecas de compatibilidad con software para arquitectura de 32 bits. <http://askubuntu.com/questions/454253/how-to-run-32-bit-app-in-ubuntu-64-bit> . En algunos sistemas operativos ya están instaladas.
3. Descomprimir tar.gz
4. Hasta aquí ya tiene un binario que puede ser ejecutado como cualquier otro binario. La sugerencia es pasarlo a algún lugar en el PATH para que puedan ejecutarlo desde cualquier ruta.

```
$ cp yacc.linux /usr/bin/byacc
$ byacc
```

- Ejercicios:

1. Implementa un intérprete para cada una de las siguientes dos gramáticas. Utiliza **byacc** y **jflex**.

– Gramática 1

```
E -> E + T
    | E - T
    | T
T -> T * F
    | T / F
    | F
F -> NUMBER
    | - NUMBER
```

– Gramática 2

```
E -> T + E
    | T - E
    | T
T -> F * T
    | F / T
    | F
F -> NUMBER
    | - NUMBER
```

2. Encuentra a manera de imprimir la pila de reconocimiento cada que se hace una reducción.
3. ¿Qué resultado da la evaluación de la expresión **3-2+8**? Explica el motivo de los resultados.

### 3 Administrativos:

1. Se deberá crear una nueva carpeta **REPO/Prácticas/Práctica2** con los siguientes archivos.
  - tokens.l : script que hará el analizador léxico.
  - arith-left.y : script con las reglas de la gramática 1.
  - arith-rigth.y : script con las reglas de la gramática 2.

- make: tendrá dos tags right y left. Permitirá introducir un archivo para prueba.
- README.md : archivo que contendrá las respuestas de los ejercicios teóricos.

2. Se entregará antes del 23.09.16