



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA



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

Trabajo Práctico 1

Teoria de Lenguajes

Segundo Cuatrimestre de 2015

Apellido y Nombre	LU	E-mail
Rodriguez Pedro	197/12	pedro3110.jim@gmail.com
Matias Pizzagali	COMP/ETAR	COMPLETAR@COMPLETAR.com

Índice

1. Introducción	3
2. Desarrollo	3

1. Introducción

El objetivo de este trabajo práctico es desarrollar un compositor de fórmulas matemáticas. El mismo tomará como entrada la descripción de una fórmula en una versión muy simplificada del lenguaje utilizado por LATEX y producirá como salida un archivo SVG (Scalable Vector Graphics).

2. Desarrollo

Para poder realizar el TP, utilizamos la librería PLY para Python, la cual permite parsear las cadenas de entrada en función de una gramática que nosotros le proveemos.

La siguiente gramática ambigua fue la propuesta por la cátedra:

$$\begin{array}{lcl}
 E & \rightarrow & E \ E \\
 & | & E \ / \ E \\
 & | & E \ ^ \ E \\
 & | & E \ - \ E \\
 & | & E \ ^ \ E \ - \ E \\
 & | & E \ - \ E \ ^ \ E \\
 & | & (\ E \) \\
 & | & \{ \ E \ } \\
 & | & l
 \end{array}$$

Para evitar tener que resolver los conflictos Shift/Reduce y Reduce/Reduce que produciría esta gramática, lo primero que hicimos fue desambiguarla, para obtener la siguiente gramática alternativa, que genera el mismo lenguaje que la propuesta por la cátedra. Para hacer esto, tuvimos en cuenta que la división es la de menor precedencia, seguida de la concatenación. También que ambas son asociativas a izquierda y que el super y sub índice no son asociativos.

$$\begin{array}{lcl}
 S & \rightarrow & E \\
 E & \rightarrow & E \ / \ A \\
 & | & A \\
 A & \rightarrow & A \ B \\
 & | & B \\
 B & \rightarrow & C \\
 & | & C \ ^ \ C \\
 & | & C \ - \ C \\
 & | & C \ ^ \ C \ - \ C \\
 & | & C \ - \ C \ ^ \ C \\
 & | & (\ C \) \\
 & | & \{ \ C \ } \\
 & | & ID
 \end{array}$$

En el parsing, todos los atributos que definimos son sintetizados. Los atributos definidos son: x, y, tam, h1, h2 y a (ancho). Cuando hacemos el parsing, inicializamos cada atributo en Nil, y construimos un árbol sintáctico para la cadena de entrada. A continuación, recorreremos el árbol 3 veces, rellenando los valores de los atributos. La primera recorrida es top-down, la segunda bottom-up y la última, top-down.