Universidad del Valle de Guatemala
Diseño de Lenguajes de Programación
Catedrático: Tomás Gálvez
Semestre II, 2015

Laboratorio #8

En este laboratorio se añadirán tres aspectos a su generador de compiladores:

- Revisión sintáctica de ***ParserSpecification*****:** deberá extender la revisión sintáctica de Cocol/R para detectar errores en (o aprobar) un programa con producciones, de acuerdo a la gramática de Cocol/R incluida en el temario de fase de proyecto #2
- Computación de conjunto FIRST: programará una función que tome como parámetro una cadena de símbolos arbitrarios. Este función devolverá un conjunto de símbolos terminales aplicando el algoritmo visto para computación de FIRST sobre el conjunto de producciones especificadas en Cocol/R
- Computación de conjunto FOLLOW: programará una función que tome como parámetro un símbolo no-terminal, y que devuelva un conjunto de símbolos terminales aplicando el algoritmo visto para computación de FOLLOW usando la función FIRST y las producciones especificadas en Cocol/R

Aclaración: su objetivo será sólo programar las funciones que computen los conjuntos FIRST y FOLLOW, pero no computar dichos conjuntos para las producciones dadas. Eso se hará en un próximo laboratorio.

Ejemplo de ejecución:

ParserSpecification en Cocol:

```
PRODUCTIONS
  E = T EP .
  EP = + T EP | # .
  T = F TP .
  TP = * F TP | # .
  F = ( E ) | id .
```

Donde # representa ε, y lo que está en negrilla son símbolos terminales.

Ejecución:

```
1. FIRST
2. FOLLOW

>> 1

Ingrese una cadena de símbolos: + T E'

Resultado: {'+'}
```

```
1. FIRST
2. FOLLOW

>> 1

Ingrese una cadena de símbolos: F T'

Resultado: {'(', 'id'}

1. FIRST
2. FOLLOW

>> 2

Ingrese un símbolo no-terminal: T

Resultado: {'+', ')', '$'}

1. FIRST
2. FOLLOW

>> 1

Ingrese una cadena de símbolos: X Y Z

No existe el símbolo 'X' en la gramática
```