INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO

COMPILADORES

GRUPO: 3CM17

Alumno: Mora Guzman Jose Antonio

BOLETA: 2018631591

TAREA 1: JERARQUIA DE CHOMSKY, YACC Y LEX

Jerarquía de Chomsky

La **Jerarquía de Chomsky** tiene cuatro niveles:

- **Gramáticas de tipo 0 sin restricciones**: incluye a todas las gramáticas formales. Estas gramáticas generan todos los lenguajes capaces de ser reconocidos por una máquina de Turing.

Ejemplo: A→ aABC
A → abC
CB → BC
bB → bb
bC → b

L(G) = { an bn, , n ≥ 0 }
Revisar: aaabbb

- **Gramáticas de tipo 1**: **gramáticas sensibles al contexto** generan los lenguajes sensibles al contexto. Los lenguajes descritos por estas gramáticas son exactamente todos aquellos lenguajes reconocidos por una máquina de Turing determinista cuya cinta de memoria está acotada por un cierto número entero de veces sobre la longitud de entrada, también conocidas como autómatas linealmente acotados.

  Ejemplo: AB → AbBc
  A → bcA
  B → b

- **Gramáticas de tipo 2 gramáticas libres del contexto** generan los lenguajes independientes del contexto.Estos lenguajes son aquellos que pueden ser reconocidos por un autómata con pila.

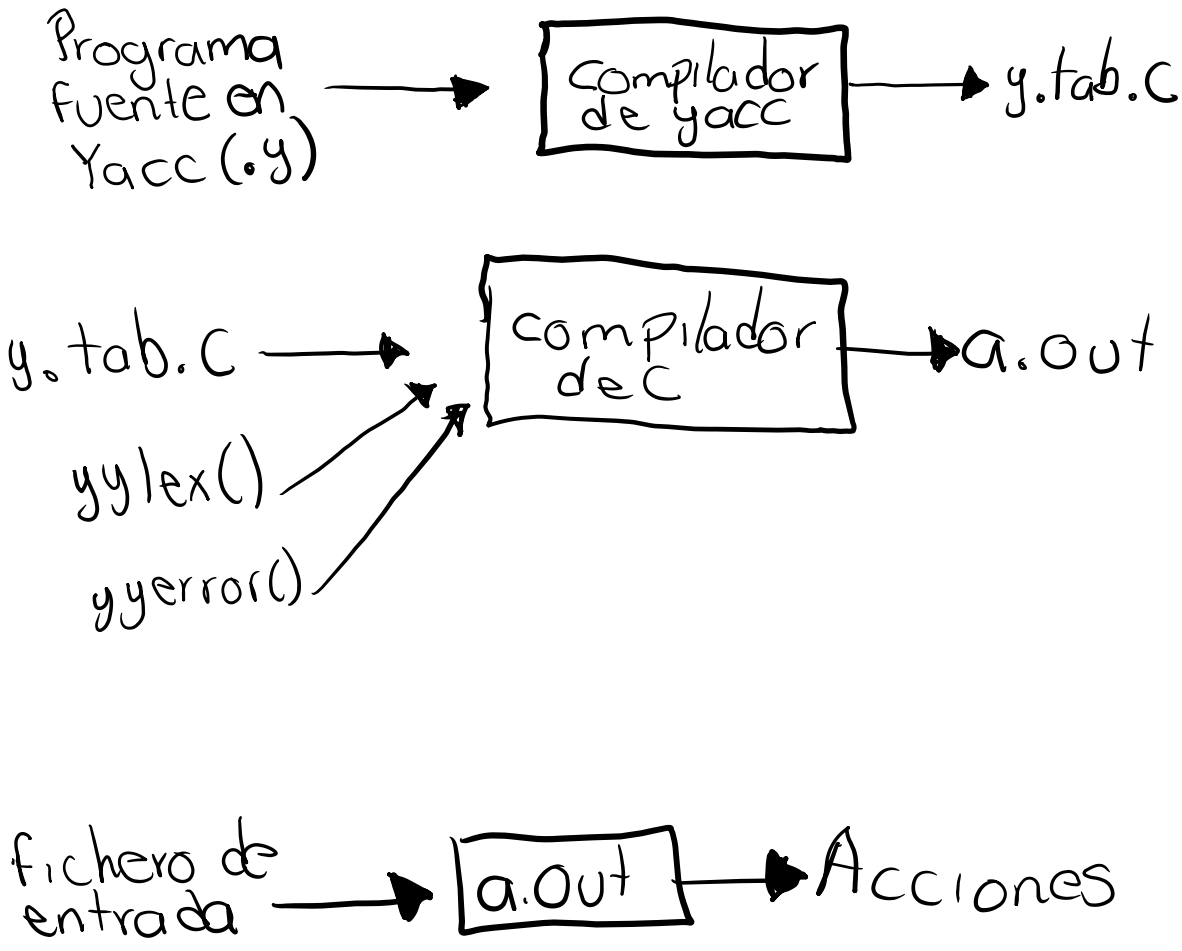
  Ejemplo: S → X a
  X → a
  X → aX
  X → abc
  X → ε

- **Gramáticas de tipo 3 gramáticas regulares** generan los lenguajes regulares. Estas gramáticas se restringen a aquellas reglas que tienen en la parte izquierda un no terminal, y en la parte derecha un solo terminal, posiblemente seguido de un no terminal. Estos lenguajes son aquellos que pueden ser aceptados por un autómata finito. También esta familia de lenguajes pueden ser obtenidas por medio de expresiones regulares.

Ejemplo: X → ε
X → a | aY
Y → b

YACC

Yacc no es directamente un analizador sino un generador de analizadores. A partir de un fichero fuente en yacc, se genera un fichero fuente en C que contiene el analizador sintáctico. Sin embargo, un analizador sintáctico de yacc no puede funcionar por sí solo, sino que necesita un analizador léxico externo para funcionar. Dicho de otra manera, el fuente en C que genera yacc contiene llamadas a una función yylex() que debe estar definida y debe devolver el tipo de lexema encontrado. Además, es necesario incorporar también una función yyerror(), que será invocada cuando el analizador sintáctico encuentre un símbolo que no encaja en la gramática.



LEX

Lex es una herramienta de los sistemas UNIX/Linux que nos va a permitir generar código C que luego podremos compilar y enlazar con nuestro programa. La principal característica de Lex es que nos va a permitir asociar acciones descritas en C, a la localización de las Expresiones Regulares que le hayamos definido. Para ello Lex se apoya en una plantilla que recibe como parámetro, y que deberemos diseñar con cuidado. Internamente Lex va a actuar como un autómata que localizará las expresiones regulares que le describamos, y una vez reconocida la cadena representada por dicha expresión regular, ejecutará el código asociado a esa regla. Externamente podemos ver a Lex como una caja negra con la siguiente estructura: