maria.guerrero@uca.es
Departamento Ingeniería Informática

Practice 2: Syntax-directed translation

Exercise 1

En el siguiente ejemplo se muestra una calculadora que admite muchas expresiones booleanas terminadas en punto y coma, las evalúa y escribe el resultado:

Entrada	**Salida**
`x := true;`	
`print x;`	`Resultado es 1`
`y := false and x;`	
`print not y;`	`Resultado es 1`
`print x and not y;`	`Resultado es 1`
`print not (x and not y);`	`Resultado es 0`
`x := not x;`	
`z := true or not (x and not y);`	

Tenga en cuenta lo siguiente:

- Se admiten sentencias de asignación (:=), además de la sentencia PRINT.
- Dentro de dichas expresiones booleanas, se pueden utilizar:
  - Constantes TRUE y FALSE
  - Identificadores.
  - El operador OR (binario y asociativo por la izquierda).
  - El operador AND (binario, asociativo por la izquierda, y de mayor prioridad que el OR).
  - El operador unario NOT de máxima prioridad.
  - Paréntesis ( ).

Se pide:

1. Diseñe una gramática para un traductor ascendente.
2. Adapte la gramática del apartado anterior para un traductor descendente.
3. Añada el esquema de traducción a la gramática del apartado 2.
4. Implemente, en C (o C++) y FLEX, el traductor descendente recursivo.