

Grupo Jueves 15:00-17:00 semanas A
– Práctica 3 –
Autor: Patricia Briones Yus

Ejercicio 1:

1. Resumen

Al compilar el fichero de bison salta el warning: 16 reduce/reduce conflicts [-Wconflicts-rr]. Una vez hecho varias pruebas sobre calcOrig con diferentes expresiones vemos que para el ejemplo "3+4*2" realiza la cuenta perfectamente, pero para "4*2+3" da syntax error, o para 4+4+4. Al contrario, calcMejor resuelve sin problema todas estas cuentas anteriores, debido a que se ha modificado el fichero de bison de manera que ahora puede sumarle a una expresión un número.

Ejercicio 2:

1. Resumen

Ejercicio 2.1: he creado una función en el apartado de reglas para transformar un número de binario a decimal. En el apartado del código he añadido una línea para que detecte la cadena de cuando un número es binario.

Ejercicio 2.2: he modificado el fichero de flex para que cuando detecte un ";" seguido de un salto de línea realizase las cuentas como solía hacerlo, y si detecta un ";" seguido de una "b" transforme el resultado de las cuentas binario. Esto último lo he hecho modificando el fichero de bison. En una de las opciones de calclist he añadido que detecte cuando el resultado se tiene que devolver en binario (calclist exp BIN), y en el apartado de reglas he creado una función para transformar un número de decimal a binario.

Ejercicio 2.3: en el fichero de flex he creado una condición de arranque inclusiva para detectar luego en el apartado del código cuando se le va a asignar un valor a la variable "acum". En el fichero de bison he añadido un token para saber cuándo es la asignación y otro para que interpretar el valor de "acum". También he añadido una regla "numero" que le da valor NUMBER o ACU (valor en ese momento de la variable "acum").

Ejercicio 3:

1. Resumen

En el fichero de flex he creado tres reglas en el apartado de código para que reconozca cada una de las reglas por las que está formada la gramática y estas devuelvan su correspondiente token. En el fichero de bison he añadido un token para cada una de las letras. También he creado tres reglas para los apartados de la gramática dada, y en cada una de ellas la expresión que reconoce.

El lenguaje reconocido por la gramática:

$$S \rightarrow CxS \mid \epsilon$$

$$B \rightarrow xCy \mid xC$$

$$C \rightarrow xBx \mid z$$

$$\text{Es } L = \{ (x^i z^j y^k x^n)^* \mid i, k \geq 0 \text{ y } j, n \geq 1 \}$$