



Laboratorio n°3

Calculadora Binaria

Integrante: Rodrigo Kobayashi Araya
Profesor: Rodrigo Abarzúa
Fecha de entrega: 15 de septiembre de 2020
Santiago, Chile

Índice de Contenidos

1. Yacc y analizadores sintácticos	1
2. Problema	1
3. Resolución	2
3.1. Compilación	2
3.2. Expresiones regulares	2
3.3. Expresiones y tokens	2
3.4. Funciones	3
4. Conclusiones	3

1. Yacc y analizadores sintácticos

Un analizador sintáctico es una parte de un compilador que genera un árbol de derivación a partir de una entrada. De esta manera se facilita el análisis de la información.

El árbol de derivación toma en cuenta la prioridad de las funciones que separan un nodo en dos ramas distintas. De esta forma se puede poner prioridad, por ejemplo, a la multiplicación por sobre la suma y la resta.

Un analizador léxico crea tokens, los cuales son procesados por el analizador sintáctico para construir la estructura, generalmente un árbol,

Los analizadores sintácticos reconocen generalmente a los lenguajes de libre contextos, que a su vez, son reconocibles por autómatas de pila.

Yacc es una herramienta que permite describir el input para programas de computador. Se describen las estructuras del input para que luego Yacc las procese y aplique las funciones, también descritas dentro del programa.

2. Problema

Desarrollar una implementación de una calculadora para manipular números naturales en representación binaria. La calculadora debe tener funciones para suma, resta, multiplicación y módulo.

3. Resolución

3.1. Compilación

Para compilar la calculadora basta con abrir una terminal de linux, dirigirse a la carpeta donde se encuentran los archivos del programa y escribir:

```
bash cmp.sh
```

Luego de esto el script cmp.sh compilará y ejecutará el programa.

3.2. Expresiones regulares

Las expresiones regulares usadas son para reconocer los números binarios, operaciones y paréntesis.

1. Números: Los números reconocidos son el 1 y el 0, de esta manera solo se reconocen números binarios.
2. Operaciones y paréntesis: se reconocen los símbolos "-", "*", y "%" para suma, resta, multiplicación y módulo respectivamente. además se reconocen los paréntesis "(z)".
3. Espacios y saltos de línea se ignoran.
4. El resto de los caracteres se toman como syntax error, se usa ECHO para imprimir por pantalla el caracter que entra como error.

3.3. Expresiones y tokens

Las expresiones y tokens que se usan en yacc son las siguientes:

1. Se reconoce la suma del tipo ".expresión + expresión".
2. Se reconoce la resta del tipo ".expresión - expresión"
3. Se reconoce la multiplicación del tipo ".expresión * expresión"
4. Se reconoce el módulo del tipo ".expresión % expresión"
5. Se reconocen los paréntesis del tipo "(expresión)"
6. La multiplicación y el módulo tienen prioridad sobre la suma y la resta
7. todas las operaciones binarias se resuelven de izquierda a derecha.
8. se toma como token los números binarios, los cuales se detectan en el lexer.

3.4. Funciones

Para las operaciones binarias existen funciones de suma, resta multiplicación y módulo de números binarios, estas se encuentran en el header `.opsBinarias.h`. Además de las funciones anteriores se usa otra mas, llamada "potencia", la cual toma dos enteros y retorna el primer entero elevado al segundo. Las funciones se describen a continuacion:

1. Suma binaria: se toman dos enteros, los cuales son sumados dígito a dígito, acarreando la suma si es superior al límite establecido por el número binario, al terminar la suma, si hay un acarreo que logra estar por sobre los dígitos sumados, se agrega al final. La salida resultante es la suma de los dos enteros binarios.
2. Resta binaria: se ingresan dos enteros, los cuales son restados dígito a dígito, desde el menor al mayor. si el primer entero entregado es menor que el segundo, la resta no se lleva a cabo. en cada iteración se divide a los números por 10 para eliminar el dígito menor. En caso de necesitar acarrear un uno desde un dígito mayor, se resta 10 del primer número.
3. Multiplicación binaria: se usa la funcion de suma binaria el número de veces indicadas por el segundo entero entregado. Se retorna el primer número sumado las veces indicadas anteriormente.
4. Módulo binario: Se emplea la funcion de resta binaria descrita anteriormente. Se resta el segundo entero entregado al primero hasta que el segundo entero sea mayor al segundo. Finalmente se entrega el resto del primer número.

4. Conclusiones

Luego de hacer la investigación pertinente sobre yacc y sus funciones, se concluye que la herramienta estudiada es eficaz pese a su simpleza, al describir unas cuantas reglas sobre cómo manejar las expresiones, yacc analiza y genera el arbol de derivación sin tener que describir nada mas. además, las forma de introducir funciones cuando se llega a una forma descrita es simple y no requiere mucha investigación extra.

A diferencia del laboratorio anterior, se logra llevar a cabalidad todas las especificaciones entregadas por el profesor, por lo tanto se puede concluir además que se genera un avance de manejo de tiempo y organización.

Finalmente, se concluye que las expresiones matemáticas con operaciones binarias y paréntesis entran dentro de los lenguajes de libre contexto, ya que pueden ser reconocidas por el analizador sintáctico y además, por justificación formal, por autómatas de pila.