Compiladores U-2017
Informe Proyecto
Traductor
C a Bash

Juan Andrés Vivas
Julián Briceño

13 de diciembre de 2017

1. Descripción del problema

Se plantea hacer una traducción simple del lenguaje C a Bash, de lo cual es importante destacar que se está realizando la traducción de un lenguaje fuertemente tipado a uno débilmente tipado.

El objetivo principal es traducir expresiones matemáticas sencillas, lectura de variables, impresiones por pantalla y la estructura de decisión if y de repetición while. Todo esto pasando debidamente por análisis léxico, sintáctico y semántico.

1.1. Alcance

Lograr traducir expresiones matemáticas simples, tales como: sumas, restas, multiplicaciones, divisiones y calculo de modulo. Para el desarrollo de dichas operaciones se acotó a operaciones que contendrán un solo tipo de operación.

Se tendrán expresiones lógicas las cuales sirven para las estructuras if y while, limitadas a un solo operador lógico.

Se leerá una sola variable por la entrada estándar y se acoto a 2 variables la cantidad que se pueden imprimir por pantalla.

El tipo de datos en C a usar son int, float y char, empero, a pesar de reconocer el tipo de dato flotante, al momento de traducir solo se tomará en cuenta su parte entera.

1.2. Especificación del lenguaje fuente

El lenguaje fuente a ser traducido es el lenguaje C, el cual para fines prácticos fue limitado a operaciones matemáticas sencillas y las estructuras de decisión y repetición if y while, respectivamente. También son reconocidas las bibliotecas que contenga el código fuente y la delcaración de la función principal "main", para esta por fines prácticos no son tomados en consideración los parámetro que pueda contener (ejemplo argc o argv).

Son considerados el manejos de los tipos de datos entero, carácter y flotante y mediante ellos implementar la estructura de asignación a variables requeridas.

Para lectura de variables es utilizada la función "scanf", la cual es acotada a la lectura de una sola variable y la impresión por panatalla mediante la función "printf.ªcotandola a una impresión de un máximo de 3 variables a la vez.

Cualquier otra operación, comando, estructura y demás no sera reconocido por nuestra gramática ni traducido al lenguaje objeto.

1.3. Especificación del lenguaje a traducir

El lenguaje de consola Bash (Bourne again shell) interpreta ordenes previamente desarrolladas en código, y ejecutadas mediante un script en una shell de Unix.

A raíz de la traducción proveniente del código en el lenguaje C, se interpretara operaciones matemáticas simples de una sola operación, comparación de variables, lectura y escritura de variables por la entrada y salida estándar, impresión de un máximo de 3 variables mediante pantalla y captura de valores.

2. Solución al problema planteado

Se plantea realizar la traducción de un lenguaje a otro sólo si luego de realizarse el proceso de verificación de correctitud del lenguaje fuente éste se encuentra de manera correcta, de lo contrario no se realiza y se arroja un error adecuado al problema.

Para poder examinar la correctitud del código fuente es necesario pasar a través de las siguientes secciones.

2.1. Análisis Léxico

Encargado de detectar todos los lexemas y dirigirlos al análisis sintáctico como tokens.

Contiene el siguiente conjunto de expresiones regulares:

LETRA [a–zA–Z_]
DIGITO [0 – 9]
{LETRA}({LETRA}|{DIGITO})*
(–)?{DIGITO}+
(–)?{DIGITO}*"."{DIGITO}+

Su funcionaliad radica en hacer "match"(verificar la coincidencia y correlación de tipos) para detectar identificadores, tipos de datos, palabras reservadas y valores numéricos

Adicionalmente detecta el siguiente conjunto de caracteres:

. ; { } ( ) < > ! = + – * / ^ : # &
<= >= != += –= *= /= %= == || &&
++ — /% , ’ "

Cada expresión regular y cada uno de los caracteres previamente descritos envían un token al analizador sintáctico, para próximamente ser utilizado.

El analizador léxico también se encarga de eliminar del lenguaje fuente los espacios en blanco, saltos de linea, comentarios y tabulaciones. Cuenta saltos de linea y rechaza aquellos caracteres que no coincidan con ningún patrón valido definido.

2.2. Análisis Sintáctico

El análisis sintáctico se realiza a través de la siguiente gramática, la cual recibe los tokens obtenidos previamente y los agrupa de acuerdo a las reglas de producción.

```
programa:
        codigo;

codigo:
        cabecera principal | principal;

cabecera:
        cabecera NUMERAL RESERVADA MENOR ID MAYOR
        | NUMERAL RESERVADA MENOR ID MAYOR
        | cabecera NUMERAL RESERVADA COMILLAS TEXTO COMILLAS
        | NUMERAL RESERVADA COMILLAS TEXTO COMILLAS
        | cabecera NUMERAL RESERVADA MENOR ID PUNTO ID MAYOR
        | NUMERAL RESERVADA MENOR ID PUNTO ID MAYOR;

principal:
        TIPO RESERVADA PARENTESISABR PARENTESISCERR LLAVEABR cuerpo LLAVECERR;

cuerpo:
        asignacion cuerpo | asignacion | declaracion cuerpo | declaracion
        | retornar cuerpo | retornar | scan cuerpo | scan
        | print cuerpo | print | estructura cuerpo | estructura
        | estructura LLAVEABR cuerpo LLAVECERR cuerpo
        | estructura LLAVEABR cuerpo LLAVECERR
        | RESERVADA LLAVEABR cuerpo LLAVECERR cuerpo
        | RESERVADA LLAVEABR cuerpo LLAVECERR;

estructura:
        RESERVADA PARENTESISABR condicional PARENTESISCERR;

scan:
        RESERVADA PARENTESISABR COMILLAS PRCVAL COMILLAS COMA
```

```
        AMPERSAND ID PARENTESISCERR PTOCOMA;

print:
        RESERVADA PARENTESISABR COMILLAS TEXTO COMILLAS PARENTESISCERR PTOCOMA
        | RESERVADA PARENTESISABR COMILLAS TEXTO PRCVAL TEXTO COMILLAS COMA
        ID PARENTESISCERR PTOCOMA
        | RESERVADA PARENTESISABR COMILLAS PRCVAL TEXTO COMILLAS COMA
        ID PARENTESISCERR PTOCOMA
        | RESERVADA PARENTESISABR COMILLAS TEXTO PRCVAL COMILLAS COMA
        ID PARENTESISCERR PTOCOMA
        | RESERVADA PARENTESISABR COMILLAS PRCVAL COMILLAS COMA ID
        PARENTESISCERR PTOCOMA
        | RESERVADA PARENTESISABR COMILLAS PRCVAL TEXTO PRCVAL COMILLAS
        COMA ID COMA ID PARENTESISCERR PTOCOMA
        | RESERVADA PARENTESISABR COMILLAS PRCVAL TEXTO PRCVAL TEXTO PRCVAL
        COMILLAS COMA ID COMA ID COMA ID PARENTESISCERR PTOCOMA;

condicional:
        ID IGUALD ID | NUM IGUALD ID | ID IGUALD NUM
        | ID MAYOR ID | ID MAYOR_I ID | ID MENOR ID | ID MENOR_I ID
        | NUM MAYOR ID | NUM MAYOR_I ID | NUM MENOR ID | NUM MENOR_I ID
        | ID MAYOR NUM | ID MAYOR_I NUM | ID MENOR NUM | ID MENOR_I NUM
        ID DIST ID| ID DIST NUM| NUM DIST ID;

retornar:
        RESERVADA NUM PTOCOMA | RESERVADA ID PTOCOMA
        | RESERVADA PARENTESISABR NUM PARENTESISCERR PTOCOMA
        | RESERVADA PARENTESISABR ID PARENTESISCERR PTOCOMA;


declaracion:
        TIPO ID PTOCOMA | TIPO ID IGUAL NUM PTOCOMA
        | TIPO ID IGUAL COMISIMPLE ID COMISIMPLE PTOCOMA
        | TIPO ID IGUAL COMISIMPLE NUM COMISIMPLE PTOCOMA
        | TIPO ID IGUAL ID PTOCOMA;

asignacion:
        ID IGUAL ID PTOCOMA | ID IGUAL NUM PTOCOMA | ID SUM_ASSIGN ID PTOCOMA
        | ID SUM_ASSIGN NUM PTOCOMA | ID SUB_ASSIGN ID PTOCOMA
        | ID SUB_ASSIGN NUM PTOCOMA | ID MUL_ASSIGN ID PTOCOMA
        | ID MUL_ASSIGN NUM PTOCOMA | ID IGUAL suma PTOCOMA
        | ID IGUAL resta PTOCOMA | ID IGUAL multi PTOCOMA
        | ID IGUAL div PTOCOMA | ID IGUAL ID PORCENTAJE ID PTOCOMA
        | ID IGUAL ID PORCENTAJE NUM PTOCOMA
        | ID IGUAL NUM PORCENTAJE ID PTOCOMA
        | ID IGUAL NUM PORCENTAJE NUM PTOCOMA | ID INC PTOCOMA
```

```
        | ID DEC PTOCOMA | INC ID PTOCOMA | DEC ID PTOCOMA;

suma:
        suma SUMA ID | suma SUMA NUM | ID SUMA ID | NUM SUMA NUM
        | ID SUMA NUM | NUM SUMA ID;

resta:
        resta MENOS ID | resta MENOS NUM | ID MENOS ID | NUM MENOS NUM
        | ID MENOS NUM | NUM MENOS ID;

multi:
        multi MULTI ID | multi MULTI NUM | ID MULTI ID | NUM MULTI NUM
        | ID MULTI NUM | NUM MULTI ID;

div:
        div DIV ID | div DIV NUM | ID DIV ID | NUM DIV NUM
        | ID DIV NUM | NUM DIV ID;
```

Para la realización del análisis se implemento la tabla de símbolos mediante un vector dinámico de tuplas, el cual se encarga de agrupar el tipo de dato con su identificador.

2.3. Análisis Semántico

Este se encarga de verificar que las palabras reservadas se encuentren correctamente ubicadas dentro del lenguaje fuente, que las variables estén declaradas antes de ser usadas y al momento de hacer una asignación los tipos de datos de ambas variables sean iguales.

2.4. Manejo de errores

Inicia en el análisis léxico. En el se lleva cuenta de cantidad de lineas que se van leyendo con la finalidad de poder indicar en que linea se ubica un error léxico en el caso de ser detectado, ademas de emitir un mensaje si no se llega a reconocer alguna palabra.

En el análisis semántico ocurre una falla al momento de que no se pueda construir un árbol de derivación correcto a partir de la gramática definida. Se detiene la traducción y se avisa en que linea esta el error sintáctico

Al momento de revisar la semántica, se valida que todas las variables sean declaradas antes de invocarse, al momento de ser asignadas tengan el mismo tipo de dato y por ultimo que las palabras reservadas coincidan con su ubicación y uso dentro del lenguaje, de lo contrario se detiene la traducción y se emite un mensaje de error con la linea donde se encontró y la variable o palabra reservada involucrada.

En el caso de cualquier error, no se generara el código en el lenguaje objeto.

3. Salida

Luego de pasar por cada una de los pasos de verificación sin levantar ninguna alerta se procede a realizar la traducción según la estructura requerida del lenguaje requerido.

En el caso particular de la traducción de C a Bash toda la estructura de la cabecera en C pasa a ser una línea de instrucción, el scanf cambia a read y el printf a echo. El uso de tipos al realizar la declaración no es necesario, los condicionales dentro de las estructuras de decisión y repeticón cambian, así como el denotar el fin de las mismas, usos de llaves caracteres epeciales, entre otras consideraciones que fueron tomadas para lograr que la traducción se realice forma correcta para el lenguaje Bash y el código se encuentre correcto para su ejecución.

Luego de ejecutar el traductor al archivo en lenguaje C, se obtiene un archivo de nombre salida.sh, el cual es un script que puede ser ejecutado desde la terminal.