UNIVERSIDAD SIMÓN BOLÍVAR
INGENIERIA EN COMPUTACIÓN
LENGUAJES III

PRIMERA ENTREGA DE LA IMPLEMENTACIÓN DE

ÇNC

Federico Flaviani
Carnet: 99-31744

Caracas, 4 de junio de 2010

INTRODUCCIÓN

El objetivo de este proyecto es el de desarrollar un compilador del lenguaje ÇNC (definido por Federico Ponte y María Gabriela Valdés) para el procesardor MIPS32.

Dicho compilador esta siendo desarrollado en Java y su analizador lexicografico y sintactico fue hecho con las herramientas javaflex y javacup. En este informe se encontraran solamente detalles del back end, por lo tanto en las siguientes paginas no encontrara información sobre las dos herramientas antes mensionadas.

Todos los programas en MIPS32 resultantes de la compilación de programas en ÇNC usados en este proyecto fueron verificados con el simulador Spim.

ESTADO DEL CÓDIGO

Solo se esta generando código para imprimir la tabla de simbolos global, ninguna otra tabla de simbolos es impresa por el compilador. Tampoco se genera código para las inicilizaciones que son hechas en el entorno global, o sea fuera del bloque main, de modo que si se quiere que el programa compilado corra correctamente en MIPS, por ahora es necesario declarar todas las variables en el entorno global y luego inicializarlas dentro del main.

Hasta el momento esta implementado la generación de codigo de todas las expreciones, exepto las de identificadores con acceso a registro, uniones y arreglos (la generación de codigo para variables de tipo basico si esta implementado).

La generación de código de las intrucciones esta implementado, salvo la de la instrucción switch y la instrucción asignación cuando el L-Value es un acceso a registro, union o arreglo.

Está implementado el cálculo de los tamaños de los tipos, exepto el de los arreglos cuyo tamaño sea declarado con una expresión distinta de constante (el cálculo del tamaño de los arreglos que son declarados con una constante si esta implementado).

No esta implementado la generación de código para expresiones de tipo float,

TECNICAS DE IMPLEMENTACIÓN

Para la implementación del código para expresiones booleanas se usan las dos tecnicas vistas en clases:

1. código con evaluación en registros

   Este tipo de código se genera cuando la expresión booleana se encuentra al lado derecho de una asignación.

2. código con salto a etiquetas predeterminadas

   Este tipo de código se genera cuando la expresión booleana se encuentra dentro de la condición de un while o un if

Las expresiones arítmeticas se compilan con la técnica de evaluación en registros, tal cual como fue visto en clases, salvo que los atributos sintetizados de las clases son metodos de los arboles abstractos y los atributos heredados son parametros que se pasan a los métodos de estos arboles. Por ejemplo:
En clases se vio que

$SUM : AEAritm \rightarrow AEAritmAEAritm$
$AEAritm_1.prox\_reg := AEAritm_0.prox\_reg$
$AEAritm_2.prox\_reg := AEAritm_0.prox\_reg + 1$
$AEAritm_0.reg := AEAritm_0.prox\_regmodN$
$AEAritm_0.codigo := AEAritm_1.codigo +$
$salv +$
$AEAritm_2.codigo +$
$linea(AEAritm_0.reg := AEAritm_1.reg + AEAritm_2.reg)$

Donde como se puede ver $prox\_reg$ es un atributo heredado y $codigo, reg$ son sintetizados.
En nuestro código en java tenemos un método reg y uno llavado codigo, y ambos tienen como parametro el entero $prox\_reg$, de modo que el codigo para la suma seria:

$left.codigo(prox\_reg) +$
$P.salvar(prox\_reg + 1) +$
$right.codigo(prox\_reg + 1) +$
$P.linea("add" + reg + "," + reg + "," + right.reg(prox\_reg + 1)) +$
$P.restaurar(prox\_reg + 1)$

Donde la clase P es donde estan todos los procedimientos estáticos que se usaron en las clases de teoria, por ejemplo salvar, restaurar, newLabel, linea, etc

Los identificadores de la tabla de simbolos global (que es la única a la que se le genera código por ahora), se implementa como etiquetas en la sección de data, los identificadores de las demas tablas se implementaran como direcciones en memoria calculadas por desplazamientos del fp.

PROGRAMAS DE PRUEBA

El compilador CNC.class recibe como entrada un archivo ascii con extención lang y retorna otro archivo ascii punto s para poder ser corrido en Spim.

Los programas de prueba que se usaron estan en la carpeta nuevasPruebas y los resultados de compilar dichos programas estan en la carpeta resultados. Estas pruebas estan divididas en tres carpetas:

1. Expresiones: Pruebas con expresiones aritmeticas y booleanas de las cuales para estas ultimas existen pruebas donde se aplican evaluación en registros y otras donde se aplican saltos a etiquetas predeterminadas.

2. If: Pruebas con if y else anidados con otras instrucciones.

3. While: Pruebas con while anidados con las demas instrucciones.