

PRÁCTICA DEL PROCESADORES DEL LENGUAJE II

CURSO 2020-2021

ENTREGA DE SEPTIEMBRE



NOMBRE Y APELLIDOS: JAVIER OCHOA PÉREZ

DNI: 50556176-Z

CENTRO ASOCIADO: MADRID – LAS TABLAS

TELÉFONO DE CONTACTO: 639 57 13 09

CORREO UNED: jochoa80@alumno.uned.es

ÍNDICE

1.	ANALIZADOR SEMÁNTICO Y COMPROBACIÓN DE TIPOS.....	3
2.	GENERACIÓN DE CÓDIGO INTERMEDIO.....	4
3.	GENERACIÓN DE CÓDIGO FINAL.....	5
4.	INDICACIONES Y EXPLICACIONES.....	6

1. ANALIZADOR SEMÁNTICO Y COMPROBACIÓN DE TIPOS.

Para la realización de esta parte, se han creado las clases AxiomImp, Constante, Expresion, SecuenciaVariables, Sentencia, TipoVector, Var y Vect que extienden de la clase NonTerminal.

Todas ellas cuentan con métodos getters y setters de acceso a los campos.

Durante el análisis semántico se controla la correcta estructura de las expresiones.

También se realiza la comprobación de tipos, lanzando errores y finalizando la ejecución cuando se produce un error en los tipos.

Los tipos básicos del lenguaje son ENTERO y LOGICO, además de los que se creen como por ejemplo los de tipo vector.

Cuando se crea un tipo se agrega a la tabla de tipos.

Igualmente, cuando se crea un símbolo del tipo correspondiente, se crea un símbolo asociado de esa clase, que se incluye en la tabla de símbolos.

Antes de añadir un símbolo o un tipo se comprueba la validez del mismo, por ejemplo, en los tipos en caso de crear un vector que sea de tipo entero o lógico.

Igualmente, en los símbolos se comprueba que en nombre no este en uso en el ámbito correspondiente.

2. GENERACIÓN DE CÓDIGO INTERMEDIO

Para la creación de código intermedio se ha seguido la estructura de Cuadruplas facilitada por el equipo docente. El código para las mismas ha sido:

- HALT: final del programa.
- VARGLOBAL: inicialización de las variables globales.
- PRINTE: imprimir un entero.
- PRINTC: imprimir una cadena.
- PRINTCV: imprimir la cadena vacía.
- ADD: suma.
- AND: conjunción lógica.
- NOT: negación lógica.
- EQ: Igualdad
- LS: menor que.
- MUL: multiplicación.
- BRF, BR, CMP, BN: comparadores y saltos.
- MV, MVA, MVP: mover valores o referencias.
- STP: asignaciones.
- INC: incremento.
- CADENA: crea los datos de la cadena guardada en lista*

*Las cadenas de texto generadas se guardan en una lista de cadenas, que luego se recuperan en el código final para trabajar sobre ellas asociándolas a etiquetas.

3. GENERACIÓN DE CÓDIGO FINAL

Para el código final se traducen las Cuadruplas mencionadas anteriormente, en base a las operaciones sobre las mismas dependiendo si el operando es Variable, Value, Temporal o Label. Hay la operación normal y la complementaria (operación y operacionComp).

A partir de estas Cuadruplas, con los operandos y el resultado, ayudándome del método append, se van creando las instrucciones para la máquina en código final.

En el ejemplo se muestra el código final para el testCase08:

```
|;Quadruple - [VARGLOBAL i, 0, null]
MOVE #0,/65535
;Quadruple - [MV i, T_0, null]
MOVE /65534,/65535
;Quadruple - [INL L_0, null, null]
L_0:
NOP
;Quadruple - [MV T_1, 10, null]
MOVE #10,/65533
;Quadruple - [CMP T_1, i, null]
CMP /65533,/65535
;Quadruple - [BN L_1, null, null]
BN /L_1
;Quadruple - [MVP T_2, i, null]
MOVE /65535, /65532
;Quadruple - [PRINTE T_2, null, null]
WRINT /65532
WRCHAR #10
;Quadruple - [INC i, null, null]
INC /65535
;Quadruple - [BR L_0, null, null]
BR /L_0
;Quadruple - [INL L_1, null, null]
L_1:
NOP
;Quadruple - [HALT null, null, null]
HALT
```

4. INDICACIONES Y EXPLICACIONES

La práctica está realizada implementando el análisis semántico, código intermedio y código final no realizando la parte optativa.

Quiero comentar, que con los vectores he tenido problemas y no están correctamente implementados en el código final.

Aunque entiendo que no es justificación ni mucho menos, he tenido en el últimos mes problemas familiares que me han impedido acabar la práctica como deseaba

Entiendo que los plazos son iguales para todos y por ello no he solicitado la posibilidad de entregar la práctica más tarde. Igualmente si tras la corrección se considera que es posible realizar modificaciones de cara a mejorar la práctica o la nota, estoy totalmente dispuesto a trabajar en ello e implementarlos. Simplemente quero avisar de cara a la previa realización de los test por parte del equipo docente.

El resto de test se han probado con modificaciones tanto la parte semántica, sintáctica, como de código intermedio y final.

La práctica aunque muy larga, me ha parecido muy interesante de cara a comprender como funciona un compilador y como se construyen y piensan los diferentes lenguajes de programación y pienso que es muy útil para cualquier estudiante del grado.