Compilador MiniLeng

Procesadores de Lenguajes

3er curso, Grado de Ingeniería en Informática, 2019-2020

Patricia Briones Yus, 735576

La ejecución del compilador se realiza de la siguiente manera:

"java MiniLeng nomFichero [-v]"

- 'nomFichero': nombre del fichero que contiene el código a compilar
- '-v' muestra un resumen de los datos contenidos en el fichero y su uso es opcional.

Características:

El formato de los ficheros a compilar puede ser tanto lenguaje MiniLeng (.ml) como ficheros de texto (txt).

El flag '-v' (verbose), una vez el compilador haya terminado, el programa muestra por pantalla una tabla como la que aparece abajo, indicando el número de apariciones por cada token detectado y su clasificación según la función que realiza.

Palabras reservadas	Comentarios: 3 Programa: 1 Accion: 0 Val: 0 Ref: 0
Funciones	Escribir(): 0 Leer(): 0 Entacar(): 0 Caraent(): 0
Tipos de datos	Entero: 0 Booleano: 1 Caracter: 3 Var: 0
Agrupaciones	Principio: 0 Fin: 0 Si: 0 Ent: 0 Si_no: 0 Fsi: 0 Mq: 0 Fmq: 0
Separadores	';': 4 ',': 4
Operaciones aritmeticas	'+': 0 '-': 0 '*': 0 '/': 0 '(': 0 ')': 0 'Mod': 0 'Div': 0
Operaciones logicas	'>': 0 '<': 0 '=': 0 '>=': 0 '<=': 0 '<>': 0 ':=': 0 'And': 0 'Or': 0 'Not': 0

Léxico:

No se hace distinción entre mayúsculas y minúsculas mediante 'ignore_case = true;' y el número máximo de tokens de preanálisis es 1 'lookahead = 1;'.

También se permite el uso de comentarios mediante el símbolo % hasta detectar un salto de línea.

En caso de existir algún error léxico se mostrará por pantalla lo siguiente:
ERROR LÉXICO (<línea, columna>): símbolo no reconocido: <símbolo>

Sintáctico:

Se permite la declaración de acciones dentro de otras, es decir, acciones anidadas y estas pueden tener o no parámetros, siendo en este último caso opcional el uso de paréntesis '()'. En caso de existir parámetros estos podrán pasar se por valor (val) o por referencia (ref) y separados por ';'.
'accion ejemplo (val caracter a; ref entero b);'

Al igual que las acciones, se permite la anidación de los bloques condicionales 'SI' y bucles 'mq'.

Los bloque de sentencias deben contener al menos una sentencia.
No se permite la declaración de variables dentro del bloque 'principio' 'fin'.
Las variables 'globales' solo se pueden declarar, no se pueden asignarles valores.

En caso de existir algún error sintáctico se mostrará por pantalla lo siguiente:
ERROR SINTÁCTICO (<línea, columna>): <Símbolo reconocido: 'símbolo'>
<Se esperaba: 'texto del error'>

o

ERROR SINTÁCTICO (<línea, columna>): <Símbolo no esperado: 'símbolo'>

Semántico:

Solo se permiten asignaciones que sean del mismo tipo y operaciones donde os dos operadores sean del mismo tipo (la operación puede devolver un tipo diferente). Los parámetros por referencia y las variables pueden ser asignables y evaluados, a diferencia de los parámetros por valor que solo

pueden ser evaluados. Se permite la escritura de cualquier expresión y cadenas constantes. Sólo se permite la lectura de tipos asignables que sean caracteres o enteros.

En caso de existir algún error sintáctico se mostrará por pantalla lo siguiente: "ERROR SEMÁNTICO (<línea, columna>): <Mensaje de error>"

Generación de código:

Se genera un código intermedio ejecutable por la Máquina P. Dado un fichero de entrada con formato '.ml', genera un fichero de código intermedio con extensión '.code'. Si ha habido algún error en el proceso de compilación, no se genera el fichero de salida. Todos los tipos de datos ocupan 16 bits, es decir, 2 Bytes (incluidas las referencias).