Instituto Tecnológico de Costa Rica
Ingeniería en Computación
Compiladores e Intérpretes
Verano, 2023
Profesor: Allan Rodríguez Dávila

TAREA #1

Gramática BNF

Lenguaje de programación Imperativo

Introducción

Un grupo de desarrolladores desea crear un nuevo lenguaje imperativo, ligero, que le permita realizar operaciones básicas para la configuración de chips, ya que esta es una industria que sigue creciendo constantemente, y cada vez estos chips necesitan ser configurados por lenguajes más ligeros y potentes. Es por esto que este grupo de desarrolladores requiere desarrollar su propio lenguaje para el desarrollo de sistemas empotrados, y como primer paso necesitan desarrollar una gramática simple y poderosa.

Requerimientos a desarrollar

Se requiere desarrollar una Gramática BNF (con tipado fuerte y explítico) para un lenguaje imperativo con las siguientes características:

a. Permitir la creación de funciones, y dentro de ellas, estructuras de control, bloques de código ({}) y sentencias de código.
b. Manejar los tipos de variables enteras, flotantes, booleanas, caracteres, cadenas de caracteres (string) y arreglo estático.
c. Se permite crear arreglos de tipo entero o char. Además, se permite obtener y modificar sus elementos, y ser utilizados en expresiones.
d. Permitir sentencias para creación de variables, creación y asignación de expresiones y asignación de expresiones a variables, y algunos casos, sólo expresiones sin asignación. La asignación se hará por medio de “<=”.
e. Las expresiones permiten combinar literales, variables y/o funciones, de los tipos reconocidos en la gramática.
f. Debe permitir operadores y operandos, respetando precedencia (usual matemática) y permitiendo el uso de paréntesis.
g. Permitir expresiones aritméticas binarias de suma (+), resta (-), división (/) –entera o decimal según el tipo--, multiplicación (*), módulo (~) y potencia (**). Para enteros o flotantes.
h. Permitir expresiones aritméticas unarias de negativo (-), ++, -- , antes del operando (preorden); esto para enteros, el negativo adicionalmente se puede aplicar a flotantes. El negativo a literales y el ++ y -- a variables.
i. Permitir expresiones relacionales (sobre enteros y flotantes) de menor, menor o igual, mayor, mayor o igual, igual y diferente. Los operadores igual y diferente

permiten adicionalmente tipo booleano. Sólo permiten un operando y 2 operadores.

j. Permitir expresiones lógicas de conjunción (^), disyunción (#) y negación (esta debe ser de tipo caracter (!)).
k. Debe permitir sentencias de código para las diferentes expresiones mencionadas anteriormente y su combinación, el delimitador de final de expresión será el carácter pipe (|). Además, dichas expresiones pueden usarse en las condicionales y bloques de las siguientes estructuras de control.
l. Debe permitir el uso de tipos y la combinación de expresiones aritméticas (binarias y unarias), relacionales y lógicas, según las reglas gramaticales, aritméticas, relacionales y lógicas del Paradigma Imperativo, por ejemplo, tomando como referencia el lenguaje C.
m. Debe permitir las estructuras de control if-[elif]-[else], do-until y for, además, permitir return y break. Las expresiones de las condiciones deberán ser valores booleanos combinando expresiones aritméticas, lógicas y relacionales.
n. Debe permitir las funciones de leer (enteros y flotantes) y escribir en la salida estándar (cadena carácter, enteros y flotantes), se pueden escribir literales o variables, se lee a identificadores.
o. Debe permitir la creación y utilización de funciones, definición clásica, estos deben retornar valores (entero, flotantes, char o booleanos) y recibir parámetros (con tipo).
p. Debe existir un único procedimiento inicial main, por medio de la cual se inicia la ejecución de los programas.
q. Además, debe permitir comentarios de una línea (@) o múltiples líneas (/_ _/).

Consideraciones especiales de sintaxis:

- La creación de funciones deberá ir precedida por la palabra reserva `function`.
- La creación de variables deberá ir precedida por la palabra reserva `local`.

Consideraciones especiales de gramática:

- El símbolo inicial se llamará navidad.
- Todas las producciones tendrán un nombre alusivo a la navidad, por ejemplo:
  - La producción para funciones podría llamarse `bolsanavideña`.
  - La producción para expresiones podría llamarse `regalo`.
  - La producción para expresiones aritméticas podría llamarse `regaloprin`.
  - La producción para expresiones relacionales podría llamarse `regalocomprado`.
  - La producción para expresiones lógicas podría llamarse `regalomanual`.

Aspectos técnicos

1. La tarea es en parejas.

2. La gramática debe ser desarrollada utilizando la notación BNF.

3. Toda gramática libre de contexto BNF debe tener un símbolo inicial, que a partir de ella se generan todas las producciones.

Documentación

La documentación externa deberá incluir:

a. Portada.
b. Descripción del problema.
c. Diseño del programa: Lista de terminales, lista de no terminales, símbolo inicial y producciones.
d. Análisis de resultados: lecciones aprendidas, objetivos alcanzados, objetivos no logrados.
e. Bitácora (blog con control de usuario, hora y fecha): adjuntar evidencia en las entradas. Utilizar git.

Evaluación

La tarea tiene un valor de **10%** de la nota final, en el rubro de Tareas.
Desglose de la evaluación de la tarea:

1. Documentación externa 5 ptos.
2. Funcionalidad 95 ptos (ver detalle en la sección de requerimientos a desarrollar)

Aspectos administrativos

Debe crear un archivo **.zip** ("TC1_Estudiante1_Estudiante2.zip") que contenga únicamente un archivo **info.txt** y 1 carpeta llamada **documentacion**, que deberá incluir el documento de ***Word* o txt** solicitado en la documentación y un archivo con las producciones de la gramática. El archivo **info.txt** debe contener la siguiente información (cualidades):

a. Nombre del curso
b. Número de semestre y año lectivo
c. Nombre de los Estudiantes
d. Número de carnet de los estudiantes
e. Número de tarea
f. Fecha de entrega
g. Estatus de la entrega (debe ser **CONGRUENTE** con la solución entregada): [Deplorable|Regular|Buena|MuyBuena|Excelente|Superior]

Entrega

Deberá subir el archivo antes mencionado al TEC Digital en el curso de COMPILADORES E INTÉRPRETES GR 60, en la asignación llamada “T1” debajo del rubro de “Tareas”. Fecha y hora de entrega

a. A más tardar 11:55:55 **PM** de martes 19 de diciembre de 2023.
   Después de este punto, **NO SE ACEPTARÁN** más trabajos.

El archivo será revisado en el sistema de Control de Plagio del TEC Digital. **Todo el contenido debe ser 100% original y en caso de detectar plagio se asignará nota cero a la tarea para todos los estudiantes del grupo y lo indicado en el artículo 75 del RREA.**