

Objetivo de la práctica

Analizar una gramática y diseñar e implementar un analizador sintáctico descendente.

Fecha límite de entrega

Viernes 11 de Noviembre de 2011.

Descripción del problema

ESeq es un lenguaje que permite escribir una secuencia de expresiones lógicas, relacionales y aritméticas. El alumno deberá diseñar e implementar un analizador sintáctico para el lenguaje *ESeq*, cuya gramática¹ se muestra en la Fig. 1. Para ello deberá probar primero si la gramática que se presenta es de la clase $\mathcal{LL}(1)$ o no. En su defecto deberán hacerse los cambios necesarios para obtener otra gramática equivalente que sí sea de esta clase, y pueda entonces implementarse el analizador predictivo correspondiente.

<i>body</i>	→	<i>begin eseq end</i>
<i>eseq</i>	→	<i>eseq ; assign</i>
<i>assign</i>	→	id := <i>quest</i>
<i>quest</i>	→	<i>prop ? quest : quest prop</i>
<i>prop</i>	→	<i>prop or clause clause</i>
<i>clause</i>	→	<i>clause and rel rel</i>
<i>rel</i>	→	<i>exp < exp exp > exp exp <= exp exp >= exp </i> <i>exp == exp exp != exp exp</i>
<i>exp</i>	→	<i>exp + term exp - term term</i>
<i>term</i>	→	<i>term * fact term / fact fact</i>
<i>fact</i>	→	<i>- fact not fact (quest) num true false</i>

Figura 1. Especificación del lenguaje *ESeq*.

Su aplicación deberá leer un archivo de texto, el cual contendrá un programa en el lenguaje *ESeq*. Como resultado, su aplicación deberá informar si el programa analizado es sintácticamente correcto o no, de acuerdo con la especificación del lenguaje. En el caso de localizar algún error sintáctico en su programa, deberá parar el análisis y generarse un mensaje de error, señalando el número de la línea donde éste ha sido encontrado.

Para los símbolos **id** (*identificadores*) y **num** (*números*), use las especificaciones que se usaron en la práctica número 1, de esta manera no será necesario modificar sus correspondientes reconocedores. Similarmemente, use la especificación de la práctica anterior para los comentarios del lenguaje *ESeq*.

¹ En esta notación, los símbolos escritos en letras *cursivas* denotan símbolos generadores. Los símbolos terminales se escriben con una tipografía *monoespaciada* cuando denotan lexemas ó bien en **negritas** cuando denotan familias de lexemas (**id** para identificadores y **num** para las literales numéricas).

Lineamientos generales para la realización de su solución

1. **Análisis de la gramática.** Deberá de analizarse la gramática de la Fig. 1 y determinar, de acuerdo con lo estudiado en el curso, si se trata de una gramática $\mathcal{L}(1)$ o no. En el caso de se determine que la gramática no cumple con este requisito deberán hacerse los ajustes necesarios, justificando cada uno de sus pasos, hasta asegurarse que se cuenta con una gramática analizable por el método predictivo.

Una vez que se cuente con una gramática correcta deberá reportar cuáles son 1) los símbolos terminales, 2) los símbolos generadores, 3) la inicialización de los conjuntos *first* (*primero*), 4) los conjuntos *follow* (*siguiente*), 5) los conjuntos *predict* (*predictivos*) y 6) el controlador correspondiente.
2. **Diseño de la implementación.** Una vez que se cuente con un controlador, el alumno deberá implementar el *algoritmo general* de análisis sintáctico $\mathcal{L}(1)$.
 - Para generar la secuencia de *tokens* que requiere su analizador sintáctico deberá usar el analizador léxico que fue implementado en la práctica anterior, haciendo las modificaciones que resulten necesarias para su correcto acoplamiento. Los reconocedores que no resulten útiles simplemente pueden cancelarse.
 - Se sugiere utilizar un diseño modular, ya que esto facilitará el mantenimiento de su proyecto para futuras aplicaciones.
 - Para las clases **num** e **id** de la gramática de *ESeq*, use las definiciones que se implementaron en la práctica anterior (es decir que los números *naturales*, *octales*, *hexadecimales* y *reales* serán todos elementos de la clase **num** de *ESeq*).
3. **Lenguaje de implementación.** Se exhorta al uso del lenguaje ANSI C/C++ como lenguaje de desarrollo. Atienda los siguientes puntos:
 - a) El uso de interfaces gráficas y/o entornos de desarrollo será opcional y se hará bajo responsabilidad del alumno. Cualquier inconveniente ocurrido durante la revisión que originado por su uso que impida la revisión del proyecto será causa de la anulación de la misma.
 - b) No podrá hacerse uso de herramientas auxiliares para la implementación de su programa que sean ajenas al contenido del curso o que suplanten las técnicas estudiadas. Cualquier solución alterna deberá ser señalada y justificada, y no deberá alterar en ningún caso los aspectos a evaluar. El abuso del no apego a este punto podrá ser causa de anulación de su proyecto.
3. **Asesoría para su elaboración.** El alumno podrá asesorarse en todo momento con el profesor durante el desarrollo de su proyecto, empleando para ello preferentemente los horarios destinados a este fin.

Requerimientos para la evaluación de la práctica

Los siguientes requerimientos serán solicitados para la evaluación de esta práctica:

1. **Verificación de la aplicación.** Se hará una corrida de prueba, durante la cual se proporcionará un archivo de texto a su programa, teniendo esta un programa escrito en el lenguaje *ESeq* posiblemente con errores de sintaxis. Se hará la verificación de los resultados de acuerdo a lo expuesto en la descripción del problema. Se verificará además que la implementación corresponda con el controlador que fue calculado.
2. **Comprensión del problema.** El alumno deberá tener una clara comprensión técnica y teórica del problema, la cual podrá ejercitar durante el desarrollo de esta práctica. Este punto será determinado mediante una evaluación que el profesor hará durante la verificación de la aplicación así como la revisión del código fuente y de la documentación entregada.
3. **Documentación del proyecto.** Deberá elaborarse un reporte de práctica, apegándose al esquema que se expone en el *Manual para la elaboración de los Reportes de Práctica*. Para esta práctica deberá informar:
 - a) Análisis de la gramática. Se deberán incluir los 6 puntos mencionados en la exposición de lineamientos generales.
 - b) Acoplamiento del analizador léxico.
 - c) Detalles relevantes en la implementación del algoritmo general de análisis sintáctico LL(1) como, por ejemplo, la codificación de los símbolos gramaticales, la implementación de estructuras de datos, estrategias de seguimiento del análisis, etc.

Recuerde que la entrega de la documentación impresa es un requisito necesario para dar inicio a la revisión de su proyecto.

Ejemplo

El siguiente programa del lenguaje *ESeq* puede ayudarle a probar su implementación:

```
begin
  n := 3.4 * 5;
  c1 := (3 < n) or (n * 3 >= 4);
  c2 := n > 20 ? true : false
end
```