**Compiladores**

Práctica 4 – Analizadores sintáctico descendente recursivo predictivo (1era parte)

**Objetivo:**

- Construir un analizador sintáctico descendente predictivo.

- Modelar una gramática.

- Escribir una tabla de análisis sintáctico

- Escribir el algoritmo de reconocimiento de una cadena utilizando el analizador.

**Modalidad:**

Individual o grupo de hasta 2 personas.

**Indicaciones:**

1. **Estableciendo una gramática**

Especificar los constructores y sobrecarga de operadores para las siguientes clases:

|  |
| --- |
| **class Produccion{**  **public:**  **string izq;**  **vector<string> der;**  **};**  **class Gramatica{**  **public:**  **vector<Production> production;**  **set<string> terminales;**  **set<string> noterminales;**  **};** |

1. **Establecer métodos de acceso a los elementos de las producciones:**

|  |
| --- |
| **vector<vector<string>> Gramatica::getProduccion(string izq);**  **void Gramatica::read(string texto);** |

1. **Gramática a modelar:**

|  |
| --- |
| **E := T Ep**  **Ep := + T Ep**  **Ep := - T Ep**  **Ep := lambda**  **T := F Tp**  **Tp := \* F Tp | / F Tp | lambda**  **F := ( E ) | num | id** |

1. **Transformar una gramática (en texto) a una estructura Gramatica.**
2. **Crear una tabla sintáctica de predicción:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **E** | **+** | **-** | **\*** | **/** | **(** | **)** | **num** | **id** | **$** |
| **Ep** |  |  |  |  |  |  |  |  |  |
| **T** |  |  |  |  |  |  |  |  |  |
| **Tp** |  |  |  |  |  |  |  |  |  |
| **F** |  |  |  |  |  |  |  |  |  |