

Procesadores del Lenguaje

Segunda Sesión

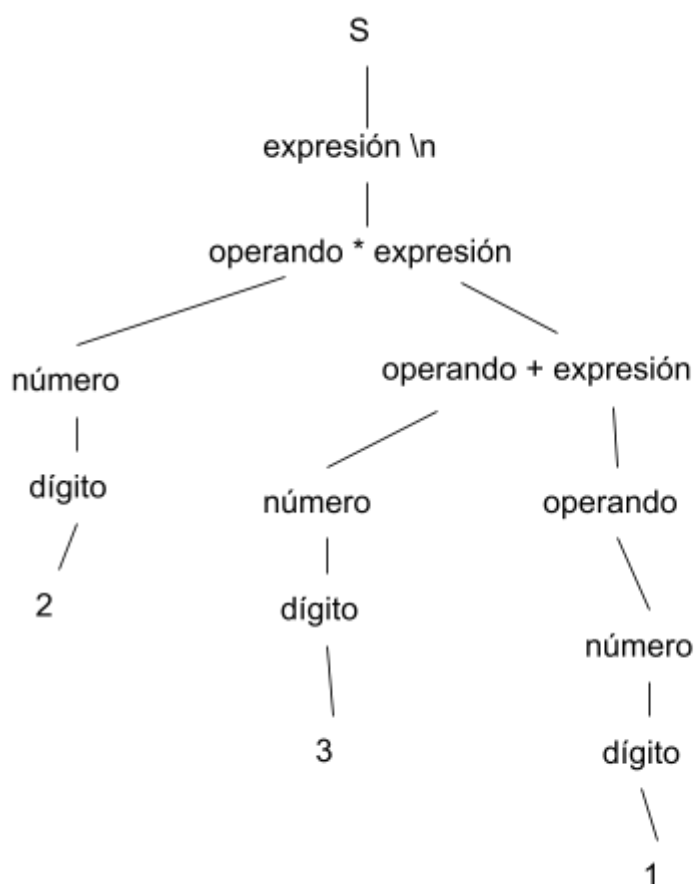
Autores: Álvaro Marco Pérez (100383382), David Rico Menéndez (100384036)

Datos: Grupo de trabajo 2

2.1. ¿Sabes determinar a qué se deben los resultados obtenidos?

La prioridad de las operaciones está definida por la recursividad de la gramática. Al tener recursividad por la derecha, la primera operación que se ejecuta es la que está más a la derecha, dando lugar a que operaciones equivalentes siguiendo el método PEMDSR.

Podemos ver un ejemplo del árbol de derivación de la operación $2*3+1$ que, en lugar de seguir la prioridad de operadores que equivaldría a $(2*3)+1$, hace $2*(3+1)$:



2.2. ¿A qué se debe esto? ¿Qué soluciones se te ocurren? Analiza el código del analizador léxico (yylex).

El analizador léxico se encarga de determinar qué parte de la entrada se utiliza como entrada para la gramática. En la condición del bucle while se contemplaba que cuando la variable 'c' fuera igual al espacio ' ', se ignorase ese carácter. Este comportamiento no es correcto, ya que da lugar a casos equivalentes como '123'='1 23'='12 3'='1 2 3'.

Para solucionar este comportamiento es suficiente con cambiar la condición del bucle while, pasando de " c == ' '" a " c == NULL ".

2.3. ¿Sabes en qué casos aparece? ¿Y a qué es debido? Prueba diversos tipos de expresiones.

| Operación | Resultado |
|-----------|--------------|
| 1+2 | 3 |
| 1-2 | -1 |
| 1*2 | 2 |
| 1/2 | 0.5 |
| (1+2) | 3 |
| (2) | 2 |
| 1+(2*3) | 7 |
| (2+(2*3)) | 8 |
| ((1+2)*3) | syntax error |
| (2*3)+1 | syntax error |
| 1+(2*3)+1 | syntax error |

Este problema surge a raíz de la estructuración de la gramática, para evitar la recursividad por la izquierda se evita las reglas de tipo:

`expresión ::= expresión + expresión`

sustituyéndolas por:

`expresión ::= operando + expresión`

Esto hace que no se pueda intercalar operaciones con y sin paréntesis de una manera en la que los paréntesis no queden anidados en el término de la derecha de la expresión.

Para solucionar este problema se ha sustituido operando por:

```
expresión ::= r_oper + expresión
```

```
r_oper ::= '(' expresión ')' | operando
```

De esta manera se permite que se puedan escribir de cualquier manera los paréntesis en la entrada. Sin embargo, la prioridad de las operaciones definidas en el estándar PEMDSR se sigue sin cumplimentar.

2.4. ¿Sabes decir por qué vale en el primer caso y no en el segundo?

En el caso de las reglas asociadas a dígito, la indeterminación se resuelve antes del campo de la acción, por lo que la gramática funciona sin problemas. Si se moviera a la derecha del campo de acción, el caso sería equivalente al correspondiente al axioma, dando lugar a una indeterminación que no se puede resolver antes del campo de la acción.