

**UNIVERSIDAD CARLOS III DE MADRID**

**PROCESADORES DEL LENGUAJE**

**PRÁCTICA 1**

**PRÁCTICA DESCENDENTE RECURSIVO**

Pedro del Moral Galán -100383350

David Rico Menéndez -100384036

# **Gramática original**

Para el desarrollo de las tareas pedidas partimos de la siguiente gramática:

Expression ::= Term + Expression |

Term - Expression |

Term \* Expression |

Term / Expression |

Term

Term ::= *Number |* (Expression)

# **Gramática modificada**

Con objetivo de cumplir todos los requisitos pedidos para la tarea de los alumnos, llegamos a la siguiente gramática:

Expression ::= Form + Expression |

Form - Expression |

Form

Form ::= Term \* Form |

Term / Form |

Term

Term ::= *Number |* (Expression) |

+Term | -Term

Como podemos ver, para asignar mayor precedencia en el producto y la división se incluye un nivel por debajo de Expression haciendo que de esta solo se produzcan la suma y la resta, mientras que de Form se produce el producto y la división. Para la adición de los signos unarios se añaden las dos últimas producciones de Term que permiten encadenar signos + y - antes de los numeros o paréntesis.

Con esta gramática, es obvio que no podemos crear un parser descendente recursivo; por tanto, para conseguir una gramática que sí lo permita se incluyen las siguientes modificaciones:

Expression ::= Form Expression’

Expression’ ::= Form + Expression |

Form - Expression |

lambda

Form ::= Term Form’

Form’::= Term \* Form |

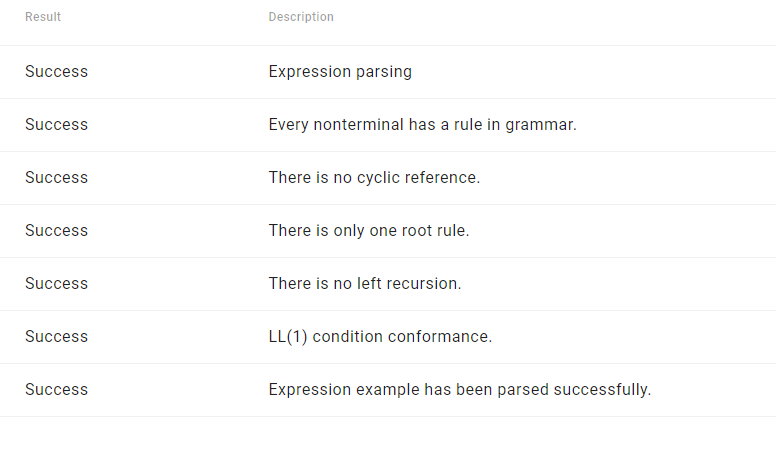
Term / Form |

lambda

Term ::= *Number |* (Expression) |

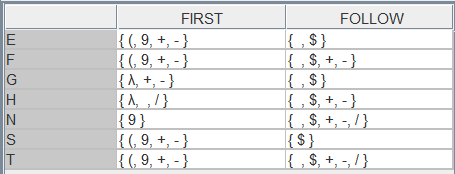
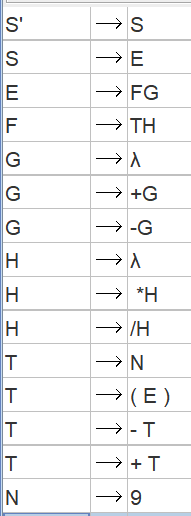
+Term | -Term

Con una herramienta que hemos encontrado en internet, hemos comprobado que efectivamente es una gramática que permite su transformación a descendente recursivo. La herramienta está en [este enlace](https://planetcalc.com/6385/). Y se puede ver que la gramática cumple todos los requisitos en la siguiente captura de pantalla:



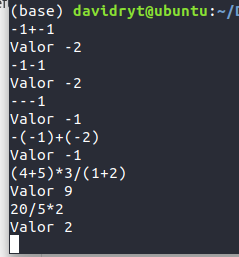
# **Conjunto Primero y Siguiente.**

Conjuntos obtenidos con JFLAP:



**Pruebas realizadas**

Estas son las pruebas realizadas para comprobar el funcionamiento del programa. Se añaden las capturas de pantalla de ejecución para comprobar el funcionamiento:



* -1 + -1
* -1 -1
* - - - 1
* -(-1)+(-2)
* (4+5)\*3/(1+2)
* 20 / 5 \* 2