

# Procesadores de Lenguaje

GRADO EN INGENIERÍA INFORMÁTICA

## Práctica Guiada 6: Grupo 3

Curso 2020/2021

Jorge Rodríguez Fraile, 100405951, Grupo 83, [100405951@alumnos.uc3m.es](mailto:100405951@alumnos.uc3m.es)  
Carlos Rubio Olivares, 100405834, Grupo 83, [100405834@alumnos.uc3m.es](mailto:100405834@alumnos.uc3m.es)

# Índice

Gramática:

3

## Gramática:

E	→	TR
R	→	$\lambda$
R	→	+E
R	→	-E
R	→	*E
R	→	/E
T	→	9
R	→	$\wedge$ E
T	→	(E)
T	→	-T
T	→	+T

  

Parse table complete. Press "parse" to use it.		
	FIRST	FOLLOW
E	{ (, 9, +, - }	{ \$, ) }
R	{ $\lambda$ , *, +, -, $\wedge$ , / }	{ \$, ) }
T	{ (, 9, +, - }	{ \$, ), *, +, -, $\wedge$ , / }

  

	(	)	*	+	-	/	9	$\wedge$	\$
E	TR			TR	TR		TR		
R		$\lambda$	*E	+E	-E	/E		$\wedge$ E	$\lambda$
T	(E)			+T	-T		9		

Para esta gramática nos hemos limitado a crear los signos unarios y las operaciones por potencias, por ahora no se muestra ningún elemento de precedencia.

Empezamos generando el axioma E que da lugar a T (que genera un número y su signo) y R, que puede ser vacío para evitar recursividad o en su defecto una nueva llamada al axioma junto las operaciones +, -, \*, /,  $\wedge$ . De esta manera conseguimos que se puedan generar operaciones de todos los tipos especificados recursivamente junto con el signo de los operandos, aparte de obviamente permitir paréntesis en las operaciones.