

Gramática de Contexto Libre del lenguaje JavaScript-PDL del curso 2021/22

/ Para descendente hay que factorizar y quitar recursividad por la izqda. (y comprobar si la resultante es LL). Para ascendente hay que comprobar si es LR */*

$P \rightarrow B P \mid F P \mid \text{eof}$ */* B corresponde a declaraciones de variables y sentencias. */*
/ Dentro de una función puede haber declaraciones de variables y sentencias, pero no declaración de otras funciones (por eso necesitamos separar B y F). */*
/ eof es el token de fin de fichero; si no se está generando ese token, se tendrá la regla $P \rightarrow \lambda$ */*

$B \rightarrow \text{let } T \text{ id}; \mid \text{if } (E) S \mid S \mid \text{while } (E) \{ C \}$ */* en el cuerpo del if simple solo puede haber una de las denominadas sentencias simples del lenguaje (S) */*
/ en vez del while, cada grupo ha de poner "su sentencia compuesta" */*

$T \rightarrow \text{int} \mid \text{boolean} \mid \text{string}$

$S \rightarrow \text{id} = E; \mid \text{return } X; \mid \text{id } (L); \mid \text{print } (E); \mid \text{input } (\text{id});$ */* estas son las sentencias simples */*
/ la asignación con operación (+ = ...) iría también aquí, como la asignación) */*

$X \rightarrow E \mid \lambda$

$C \rightarrow B C \mid \lambda$ */* el cuerpo (C) de un while o de una función puede tener declaraciones de variables y sentencias, pero no declaración de funciones */*

$L \rightarrow E Q \mid \lambda$ */* L y Q derivan los parámetros actuales */*

$Q \rightarrow , E Q \mid \lambda$

$F \rightarrow \text{function id } H (A) \{ C \}$ */* declaración de función; H es λ si la función no devuelve nada (procedimiento)*/*

$H \rightarrow T \mid \lambda$

$A \rightarrow T \text{ id } K \mid \lambda$ */* A y K derivan los parámetros formales */*

$K \rightarrow , T \text{ id } K \mid \lambda$

$E \rightarrow E \&\& R \mid R$ */* las reglas que siguen son para la generación de expresiones (respetando la precedencia y la asociatividad de los operadores). Cada grupo pondrá sus operadores */*

$R \rightarrow R > U \mid U$

$U \rightarrow U + V \mid V$

$V \rightarrow \text{id} \mid (E) \mid \text{id } (L) \mid \text{entero} \mid \text{cadena}$ */*Para el que tenga ++id, la regla iría aquí */*