

Ejercicio 1: Elimina la recursión por la izquierda en la siguiente gramática:

$S \rightarrow Sa \mid Sb \mid aXb$

$X \rightarrow Xb \mid bb$

Old Grammar

Grammar

```
S → S a
  | S b
  | a X b.
X → X b
  | b b.
```

New (Transformed) Grammar

Grammar

```
S → a X b S1 .
S1 → a S1
    | b S1
    | .
X → b b X1 .
X1 → b X1
    | .
```

Performed the 'removing left recursion' transformation.

Unfortunately the grammar is still not LL(1).

Ejercicio 2: Factoriza por la izquierda la siguiente gramática:

$S \rightarrow aX \mid aY \mid aaX \mid b$

$X \rightarrow bX \mid bY \mid \epsilon$

$Y \rightarrow aX \mid bY \mid a \mid bb$

Old Grammar

Grammar

```
S → a X
  | a Y
  | a a X
  | b .
X → b X
  | b Y
  | .
Y → a X
  | b Y
  | a
  | b b .
```

New (Transformed) Grammar

Grammar

```
S → a Fa
  | b .
Fa → X
    | Y
    | a X .
X → b Fb
  | .
Fb → X
    | Y .
Y → a Fa1
  | b Fb1 .
Fa1 → X
    | .
Fb1 → Y
    | b .
```

Performed the 'removing left factors' transformation.

Unfortunately the grammar is still not LL(1).

Ejercicio 3: Calcula los conjuntos de predicción PRIMERO y SIGUIENTE para todos los no terminales de la siguiente gramática:

$X \rightarrow aXb \mid bZa \mid Y \mid \epsilon$

$Y \rightarrow aY \mid XYZ \mid ab$

$Z \rightarrow bX$

JFLAP : <untitled1>

File Input Test Convert Help

Editor Build LL(1) Parse

Do Selected Do Step Do All Next Parse

Parse table complete, but has ambiguity.

	FIRST	FOLLOW
X	{ λ , a, b }	{ a, b, \$ }
Y	{ a, b }	{ a, b, \$ }
Z	{ b }	{ a, b, \$ }

	a	b	\$
X	λ Y aXb	λ Y bZa	λ
Y	XYZ aY ab	XYZ	
Z		bX	

Ejercicio 4: Comprueba si la siguiente gramática es LL(1):

$S \rightarrow PS'$

$S' \rightarrow +S \mid \epsilon$

$P \rightarrow RP'$

$P' \rightarrow *P \mid \epsilon$

$R \rightarrow (S) \mid \text{num}$

Grammar

```
S → P S' .
S' → + S
    | .
P → R P' .
P' → * P
    | .
R → ( S )
    | num .
```

Some sentences generated by this grammar: {num, (num), num + num, num * num, ((num)), num * (num), (num) + num, (num * num), num + (num), (num) * num, (num + num), num + num + num, num * num + num, num + num * num, num * num * num, (num) * num * num, (num * num) * num, num + num * num * num, num * num + num * num, num * num * num * num }

- All nonterminals are reachable and realizable.
- The nullable nonterminals are: S' P'.
- The endable nonterminals are: P' R S S' P.
- No cycles.

nonterminal	first set	follow set	nullable	endable
S	(num))	no	yes
S'	+)	yes	yes
P'	*) +	yes	yes
P	(num)	+	no	yes
R	(num)	* +	no	yes

The grammar is LL(1).

- attempt to transform the grammar (to LL(1))
- generate LL(1) parsing table
- generate LR(0)/SLR(1) automaton
- generate LALR(1) automaton
- generate LR(1) automaton

Return home to [enter a new grammar](#).

Ejercicio 5: Construye la tabla predictiva LL(1) para la siguiente gramática. Usa la tabla para analizar la frase abab:

$X \rightarrow aXb \mid bYa$

$Y \rightarrow Zb \mid b$

$Z \rightarrow a$

JFLAP : <untitled1>

File

Input

Test

Convert

Help

Editor

Build LL(1) Parse

LL(1) Parsing

Table Text Size

	a	b	\$
X	aXb	bYa	
Y	Zb	b	
Z	a		

Start

Step

Noninverted Tree

Input

abab

Input Remaining

\$

Stack

ab

LHS		RHS
X	→	aXb
X	→	bYa
Y	→	Zb
Y	→	b
Z	→	a

String rejected.