

## LALR(1) generado manualmente

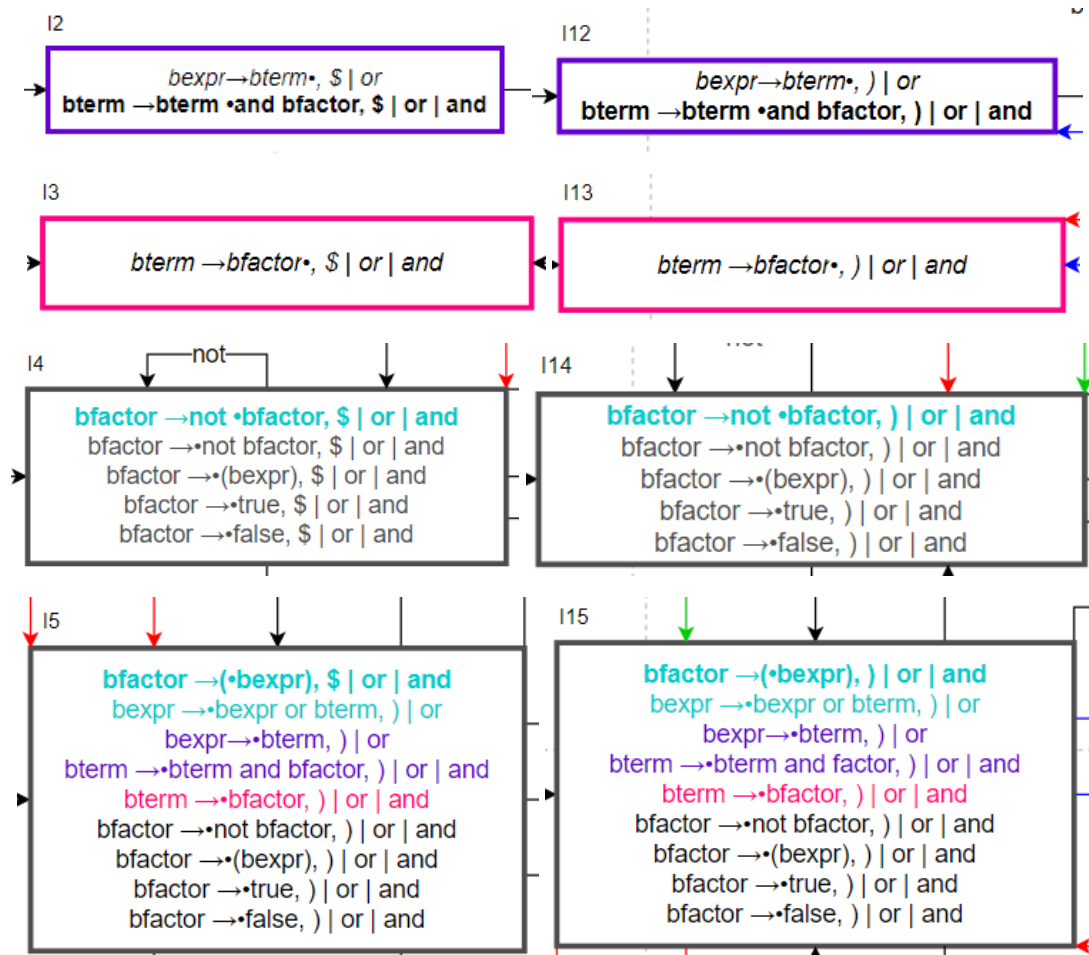
Gramática:

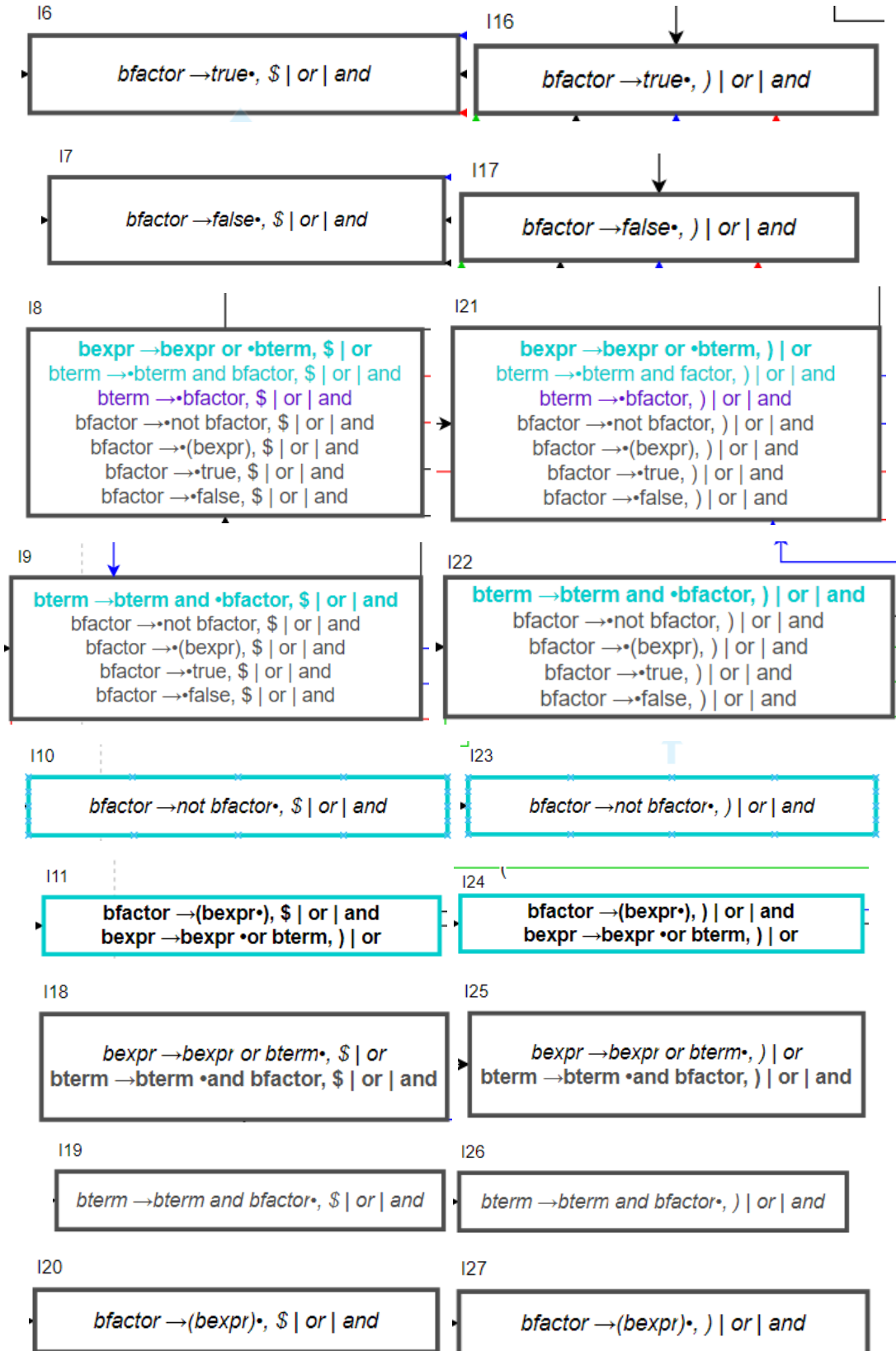
1.  $bexpr \rightarrow bexpr \text{ or } bterm$
2.  $bexpr \rightarrow bterm$
3.  $bterm \rightarrow bterm \text{ and } bfactor$
4.  $bterm \rightarrow bfactor$
5.  $bfactor \rightarrow \text{not } bfactor$
6.  $bfactor \rightarrow (bexpr)$
7.  $bfactor \rightarrow \text{true}$
8.  $bfactor \rightarrow \text{false}$

Grafo CLR(1):

Adjunto en la misma carpeta que este archivo como "Lab2-Compis.drawio.png"

Estados con mismo CORE:





**Tabla LALR(1):**

	Action								Goto		
	or	and	not	(	)	false	true	\$	bexpr	bterm	bfactor
0			S4-14	S5-15		S7-17	S6-16		1	2-12	3-13
1	S8-21							accept			
2-12	R2	S9-22			R2			R2			
3-13	R4	R4			R4			R4			
4-14			S4-14	S5-15		S7-17	S6-16				10-23
5-15			S4-14	S5-15		S7-17	S6-16		11-24	2-12	3-13
6-16	R7	R7			R7			R7			
7-17	R8	R8			R8			R8			
8-21			S4-14	S5-15		S7-17	S6-16			18-25	3-13
9-22			S4-14	S5-15		S7-17	S6-16				19-26
10-23	R5	R5			R5			R5			
11-24	S8-21				S20-27						
18-25	R1	S9-22			R1			R1			
19-26	R3	R3			R3			R3			
20-27	R6	R6			R6			R6			

## Comparación con Analizador YACC/FLEX

**Tabla LALR(1) generada con *grammar.output*:**

Para generar *grammar.output* se prescindió de las reglas generadas por el No terminal “lines” para obtener estados más parecidos a los generados manualmente.

```

lines : lines bexpr '\n' {
    if( $2 == 1 )
    {
        printf("true (1)\n");
    }
    else
    {
        printf("false (0)\n");
    }
}
| lines '\n'
| /* empty */
;

```

Las reglas de “lines” son utilizadas para tomar en cuenta los saltos de línea entre expresiones booleanas. Si estas reglas son eliminadas también al momento de generar *grammar.y*, el analizador no reconoce las expresiones de forma correcta.

Se realizó una tabla LALR(1) basada en *grammar.output* y se colocó el número de cada estado generado por *bison* en la fila de la tabla que ocupa su estado equivalente en la tabla del analizador generado manualmente.

	Action								Goto		
	or	and	not	(	)	false	true	\$	bexpr	bterm	bfactor
0			S1	S4		S3	S2		5	6	7
5	S11							S10			
6	R2	S12			R2			R2			
7	R4	R4			R4			R4			
1			S1	S4		S3	S2				8
4			S1	S4		S3	S2		9	6	7
2	R7	R7			R7			R7			
3	R8	R8			R8			R8			
11			S1	S4		S3	S2			14	7
12			S1	S4		S3	S2				15
8	R5	R5			R5			R5			
9	S11				S13						
14	R1	S12			R1			R1			
15	R3	R3			R3			R3			
13	R6	R6			R6			R6			
10								accept			

Se puede observar que tanto las Acciones como los Goto son los mismos, ya que van al mismo estado y reducen las mismas reglas.

La única diferencia es que en el analizador generado por *bison* se utiliza un estado más (110) para llegar a **accept**.