

Exemplo: Gramática para impressões ou testes condicionais

```
PROGRAMA → CMD EOF
CMD       → if E then { CMD }
           | if E then { CMD } else { CMD } | print T
E         → E OP E | T
OP        → < | <= | == | != | > | >= | + | -
T         → id | num
```

Primeiro, foi eliminada a precedência dos operadores:

A regra $E \rightarrow E \text{ OP } E$ foi tratada como as regras:

```
E → E < E | E <= E | E == E | E != E | E >= E | E > E
    | E + E | E - E
```

Logo:

```
E → E < F | E <= F | E == F | E != F | E >= F | F
F → F + T | F - T | T
T → id | num
```

Em seguida, foi feita uma fatoração à esquerda para as regra de CMD, e após, foram eliminadas as recursões à esquerda das regras que contém os operadores. Então, obteve-se:

```
PROGRAMA → CMD EOF 1
CMD       → if E then { CMD } CMD' 2 | print T 3
CMD'      → else { CMD } 4 | ε 5
E         → F E' 6
E'        → < F E' 7 | <= F E' 8 | == F E' 9 | != F E' 10
           | > F E' 11 | >= F E' 12 | ε 13
F         → T F' 14
F'        → + T F' 15 | - T F' 16 | ε 17
T         → id 18 | num 19
```

Cálculo do FIRST e FOLLOW:

não-terminal	FIRST	FOLLOW
PROGRAMA	if, print	EOF
CMD	if, print	EOF, }
CMD'	else, ε	EOF, }
E	id, num	then
E'	<, <=, >, >=, ==, !=, ε	then
F	id, num	<, <=, >, >=, ==, !=, then
F'	+, -, ε	<, <=, >, >=, ==, !=, then
T	id, num	+, -, <, <=, >, >=, ==, !=, then

Cálculo da Tabela Preditiva:

não-terminal	if	then	else	print	{	}	<	<=	>	>=	==	!=	+	-	id	num	\$
PROGRAMA	1			1													synch
CMD	2			3		synch											synch
CMD'			4			5											5
E		synch													6	6	
E'		13					7	8	11	12	9	10					
F		synch					synch	synch	synch	synch _h	synch _h	synch			14	14	
F'		17					17	17	17	17	17	17	15	16			
T		synch					Synch _h	Synch	Synch	Synch _h	Synch _h	Synch	Synch _h	synch	18	19	

Exemplo: Programa válido para essa linguagem:

```

if var - 1 < 100 then {
    print var
}
else {
    if var + 1 >= 100 then {
        print var
    }
    else {
        print 100
    }
}

```