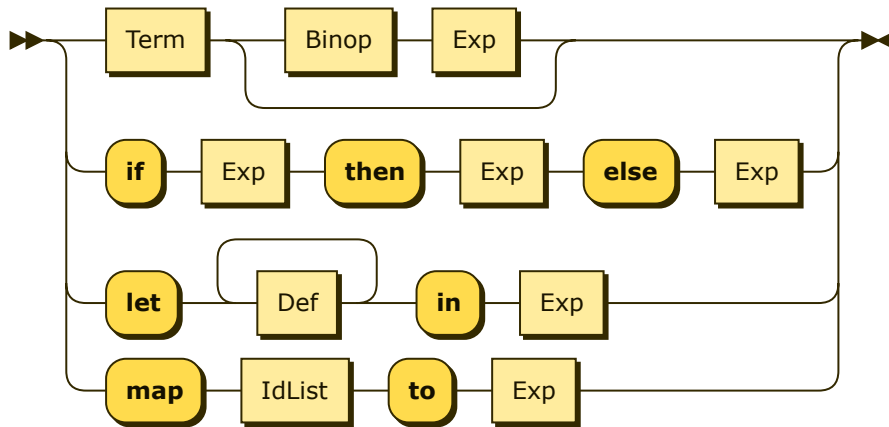


Exp:

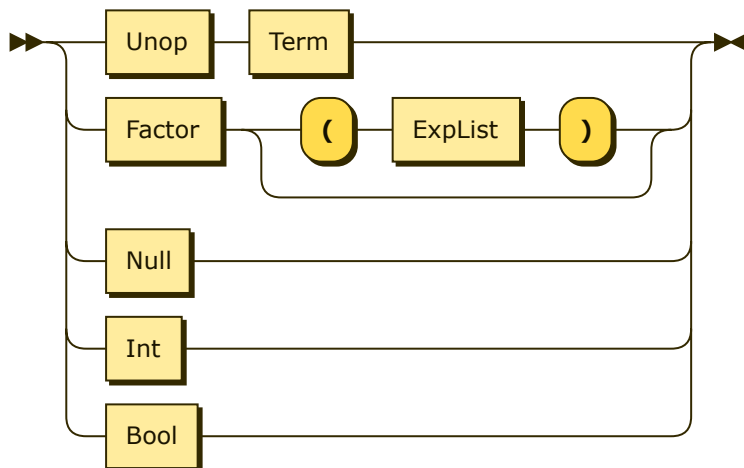


```
Exp ::= Term ( Binop Exp )?  
      | 'if' Exp 'then' Exp 'else' Exp  
      | 'let' Def+ 'in' Exp  
      | 'map' IdList 'to' Exp
```

referenced by:

- Exp
- ExpList
- Factor

Term:



```
Term ::= Unop Term  
       | Factor ( '(' ExpList ')' )?  
       | Null  
       | Int  
       | Bool
```

referenced by:

- Exp
- Term

Def:

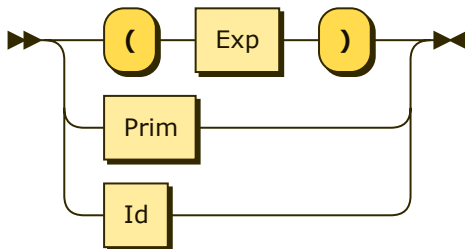


Def ::= Id ':= ' Expr ';'

referenced by:

- Exp

Factor:

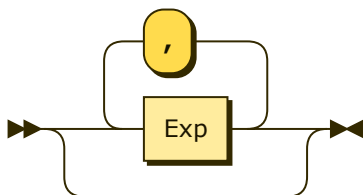


Factor ::= '(' Exp ')'
 | Prim
 | Id

referenced by:

- Term

ExpList:

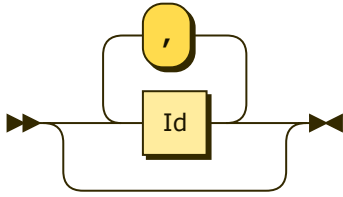


ExpList ::= (Exp (',' Exp)*)?

referenced by:

- Term

IdList:



`IdList ::= (Id (',' Id)*)?`

referenced by:

- Exp