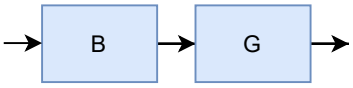
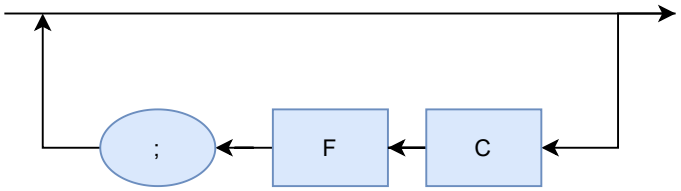
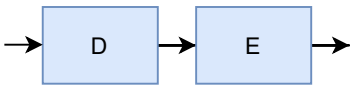
A -> B G



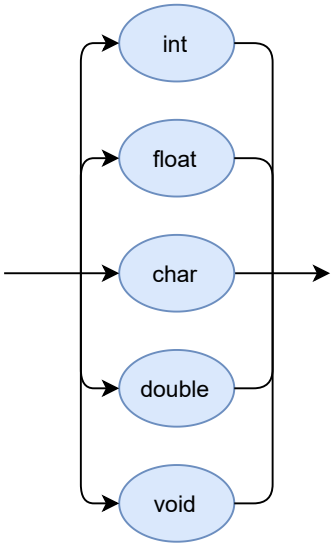
B -> CF;B | ε



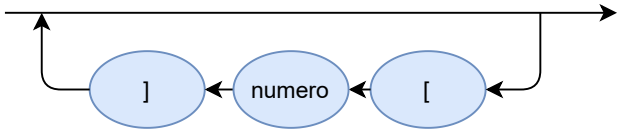
C -> D E



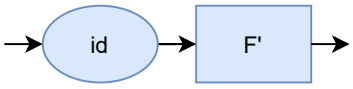
D -> int | float | char | double | void



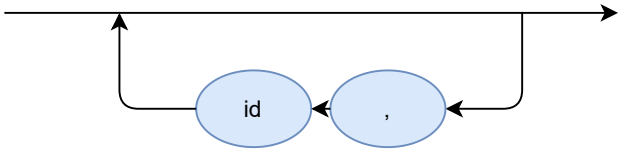
E -> [numero] E | ε



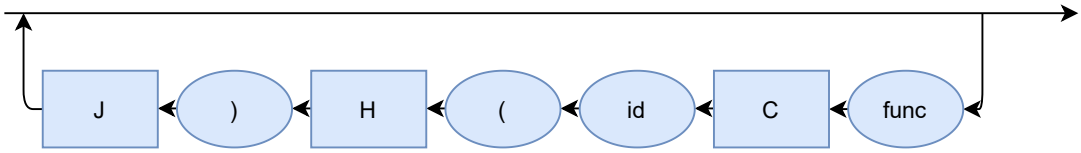
F -> id F'



F' -> , id F' | ε



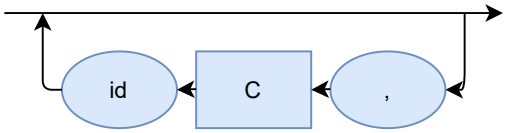
G -> func C id(H) J G | ε



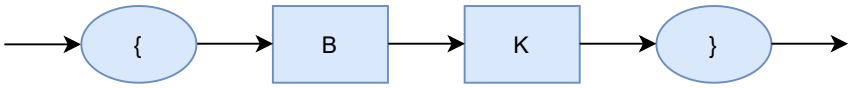
H -> I | ε

I -> C id I' | ε

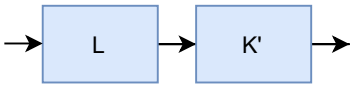
I' -> ,C id I' | ε



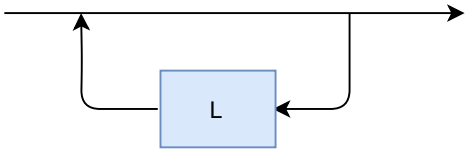
J -> {B K}



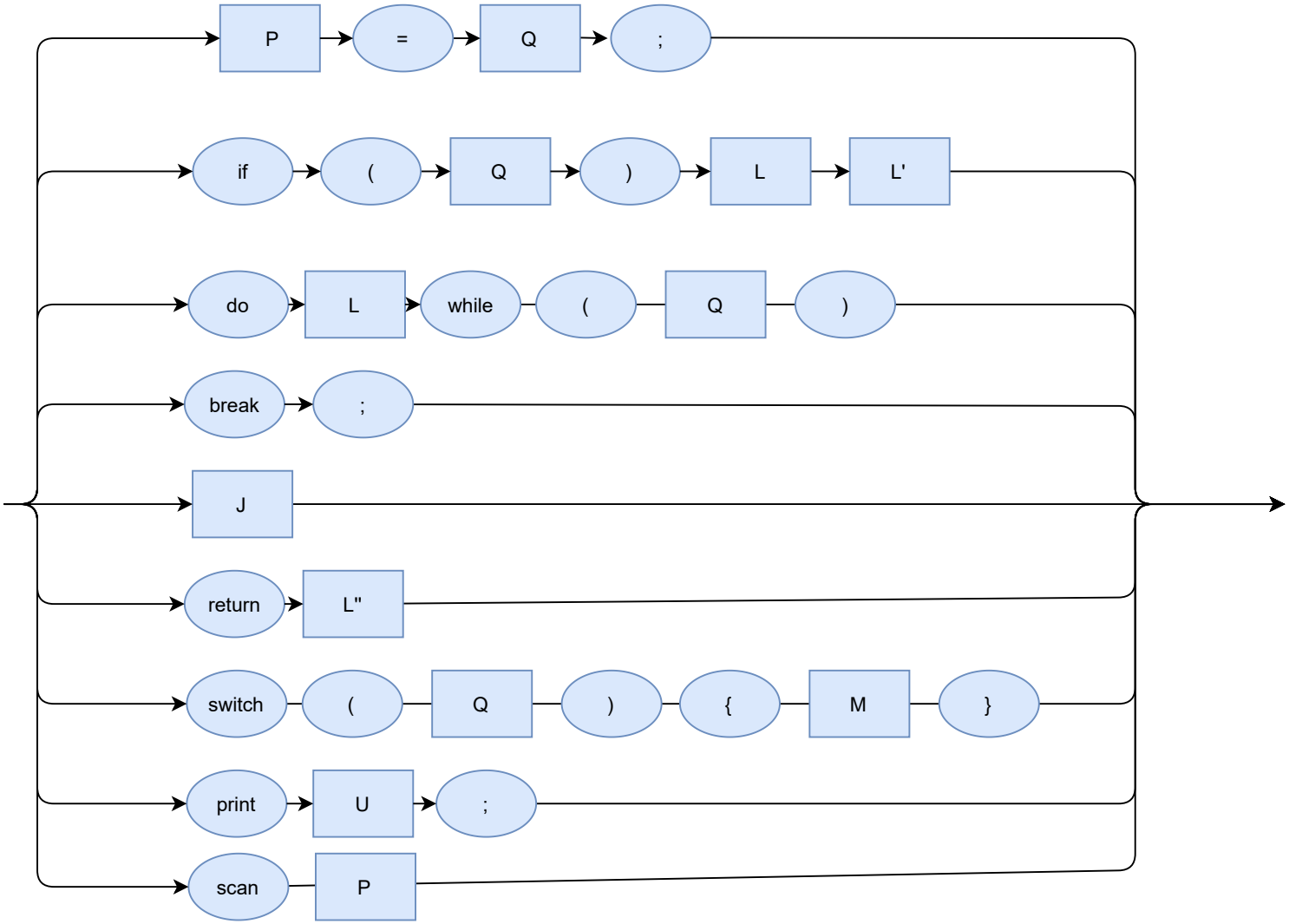
K -> L K'



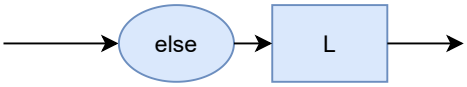
K' -> L K' | ε



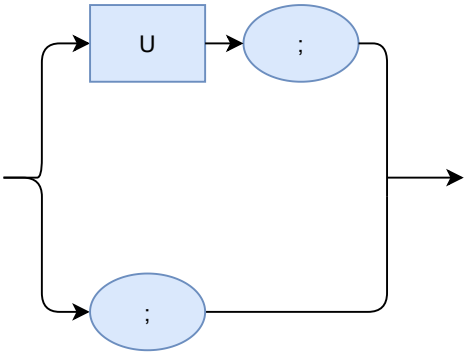
L -> P = Q; | if(Q) L L' | while(Q) L | do L while(Q) | break; | J | return L" | switch(Q) {M} | print U; | scan P



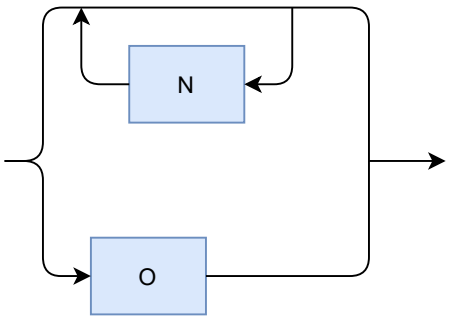
L' -> else L | ε



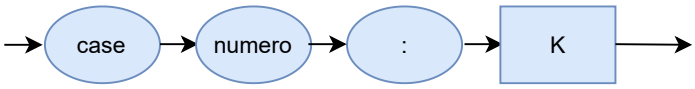
L" -> U; | ;



M -> N M | O | ε

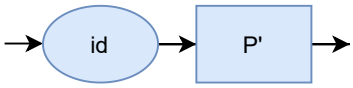


N -> case numero: K

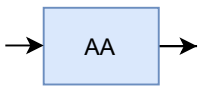


O -> default: K

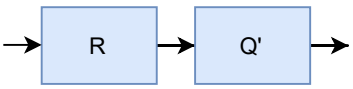
P -> id P'



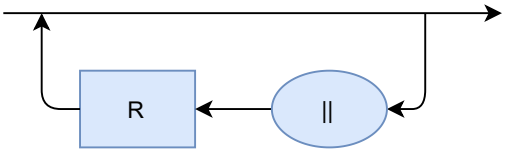
P' -> AA | ε



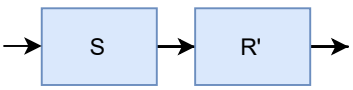
Q -> R Q'



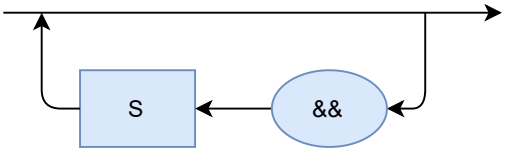
Q' -> || R Q' | ε



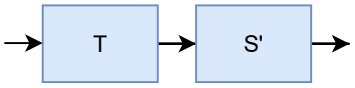
R -> S R'



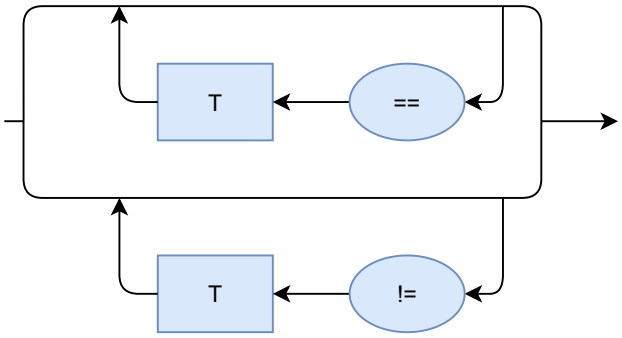
R' -> && S R' | ε



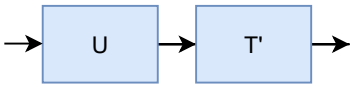
S -> TS'



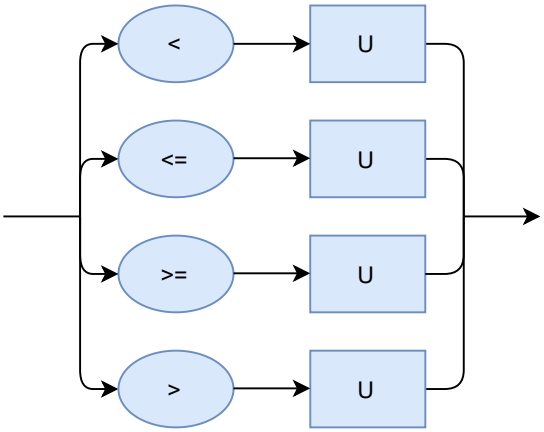
S' -> == TS' | != TS' | ε



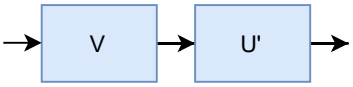
T -> U T'



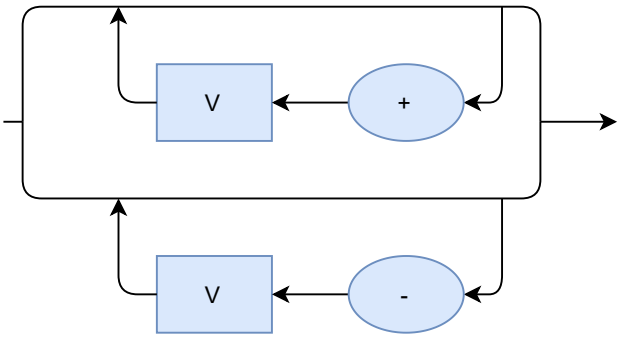
T' -> < U | <= U | >= U | > U | ε



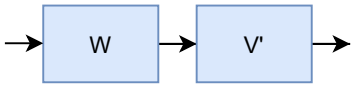
U -> V U'



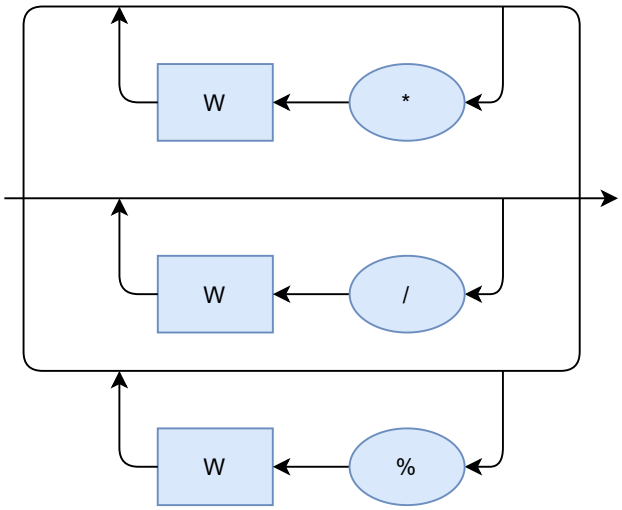
U' -> + V U' | - V U' | ε



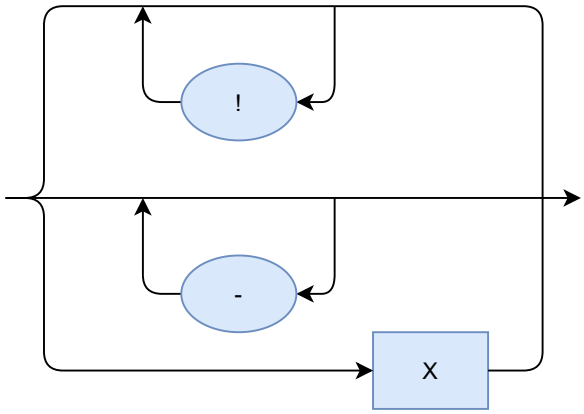
V -> W V'



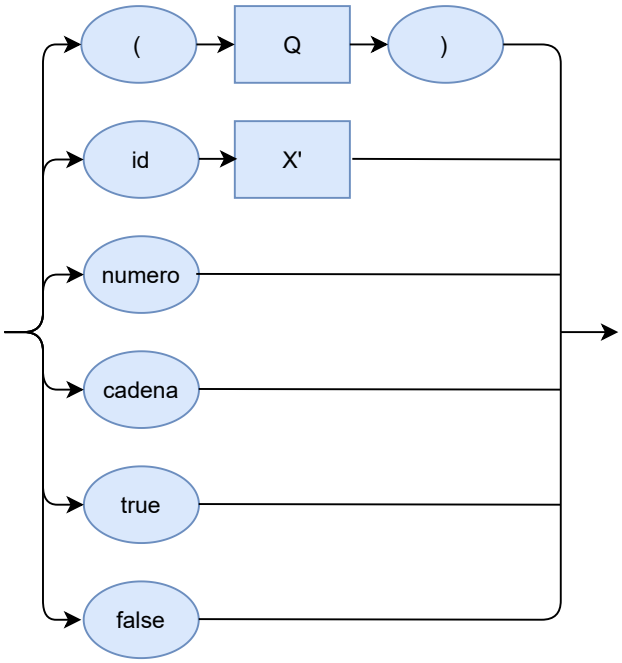
V' -> * W V' | / W V' | % W V' |ε



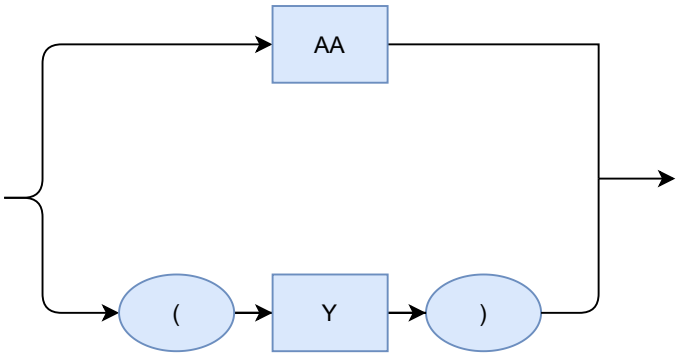
W -> !W | -W | X



X -> (Q) | id X' | numero | cadena | true | false

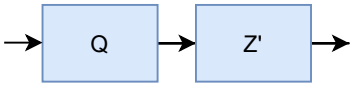


X'-> AA | (Y) | ε

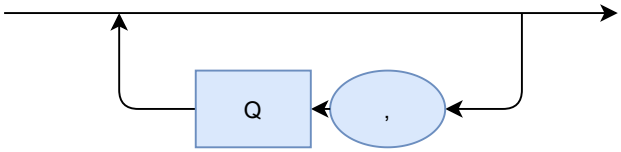


Y -> Z | ε

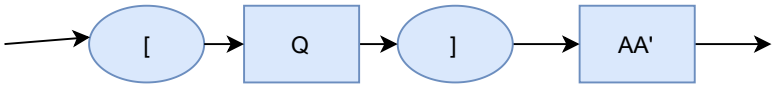
Z -> QZ'



Z' -> ,QZ' | ε



AA -> [Q] AA'



AA' -> [Q] AA'