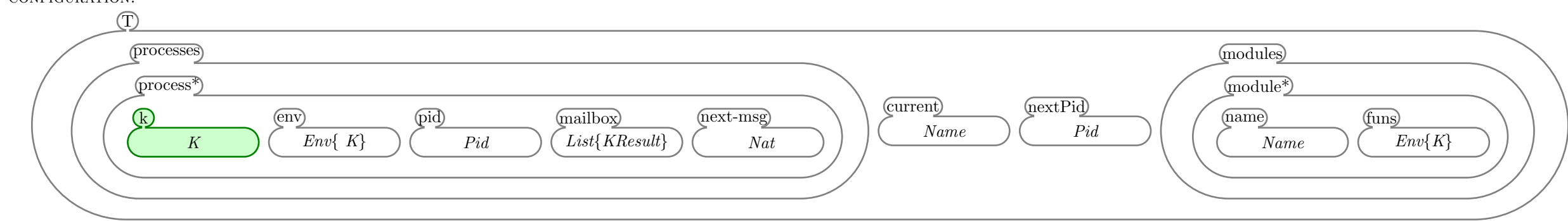


```

MODULE ERTISST
IMPORTS GENERIC-EXP-SEMANTICS
SYNTACTIC CONSTRUCTS:
  Attr ::=
    | Attr Attr [renameTo ↗.]
    | -export([ Fun ])
    | -import( #Name, Fun ),
  CaseClauses ::=
    | CaseClauses ; CaseClauses
    | Pattern :: Exp
  Cases ::=
    | Cases ; Cases
    | AnyVar( Pattern , Eps )
  Decl ::=
    | FunCases .
    | Decl Decl [renameTo ↗.]
  ETerm ::= #Bool[ #Int] #Name[ #String]
    | []
  Exp ::= Pattern[ #Bool] #Term[ ] [strict]
    |
    | set[]
    | #Name[ ]
    | [ Eps ] [aux]
    | Eps [strict]
    | receive Cases end
    | length( Exp ) [strict]
    | tuple-size( Exp ) [strict]
    | Exp , Exp [strict]
    | Pattern = Exp [strict(2)]
    | #Name ( Exp )
    | [ Exp - Exp ] [strict]
    | case Pattern of CaseClauses end [strict(1)]
    | element( Exp , Exp ) [strict]
    | #Name : #Name ( Exp ) [strict(3)]
    | setelement( Exp , Exp , Exp ) [strict]
    | spawn( Exp , Exp , Exp ) [strict]
  Eps ::= Exp Patterns
    | Eps ; Eps [strict(1)]
    | Eps , Eps [aux]
    | Eps , Eps [strict]
  Fun ::=
    | Fun , Fun
    | #Name // #Int
  FunCases ::=
    | FunCases ; FunCases [renameTo ↗.]
    | #Name [ := ] Eps
    | AnyVar( #Name , Pattern , Eps )
  Mod ::=
    | Mod Mod [renameTo ↗.]
    | -module( #Name ). Attr
    | -module( #Name ). Decl
    | -module( #Name ). Attr Decl
  Pattern ::= ETerm
    | Patterns
    | [ pat( #Name )
    | [ Pattern - Pattern ] [strict]
  Patterns ::= Pattern
    | Patterns ; Patterns
    | Patterns , Patterns [aux]
  Pgm ::= Mod Stmt
    | Mod Stmt [renameTo ↗.]
  Stmt ::=
    | Exp , [strict]
    | Stmt Stmt [renameTo ↗.]
    | match( Pattern , Exp ) [strict(2)]
SEMANTIC CONSTRUCTS:
  Config ::=
    | run( Pgm )
  K ::=
    | match-failure
  KResult ::=
    | List( KResult ) at Nat
  List( KResult ) ::=
    | del List( KResult ) at Nat
CONFIGURATION:

```



K EQUATIONS AND RULES:

EQUATION:

$$\frac{}{[]}$$

EQUATION:

$$-$$

EQUATION:

$$\frac{}{V.}$$

EQUATION:

$$\frac{P = V}{\text{match}(P, V) \curvearrowright V}$$

EQUATION:

$$\frac{}{\emptyset.(\cdot)}$$

EQUATION:

$$\frac{}{V..V'}$$

EQUATION:

$$\frac{}{[V \dashv V']}$$

EQUATION:

$$\frac{}{V \dashv V'}$$

EQUATION:

$$\frac{}{\text{length}([])}$$

EQUATION:

$$\frac{}{\text{length}([V \dashv V'])}$$

EQUATION:

$$\frac{}{\text{match}(V..V')}$$

EQUATION:

$$\frac{}{\text{match}([], [])}$$

EQUATION:

$$\frac{}{\text{match}([H \dashv T], [VH \dashv VT])}$$

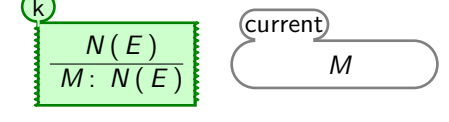
EQUATION:

$$\frac{}{\text{match}(\text{pat}(M), \text{pat}(x))}$$

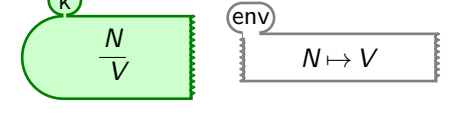
EQUATION:

$$\frac{}{\text{match}(V..V') \curvearrowright \text{Rest}}$$

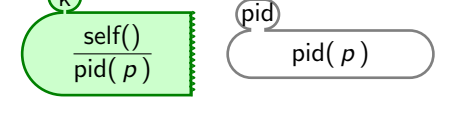
EQUATION:



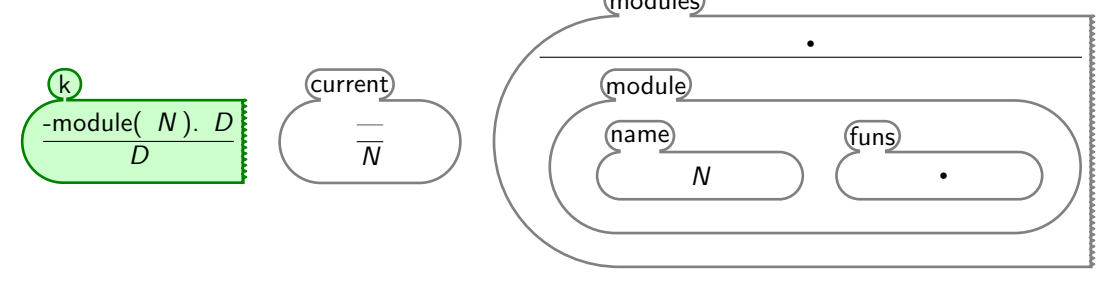
EQUATION:



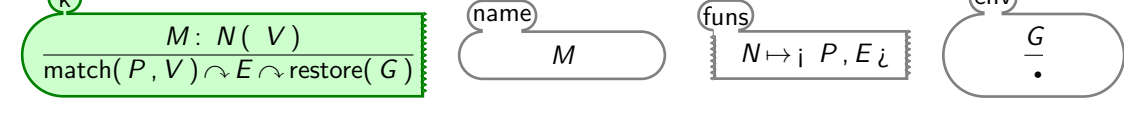
EQUATION:



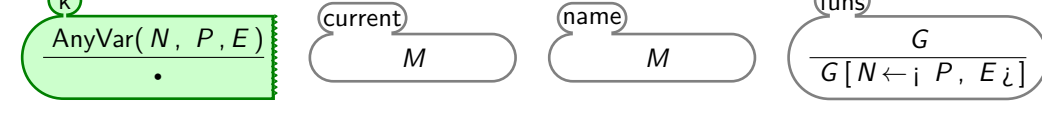
EQUATION:



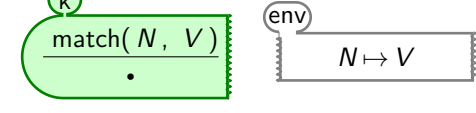
EQUATION:



EQUATION:



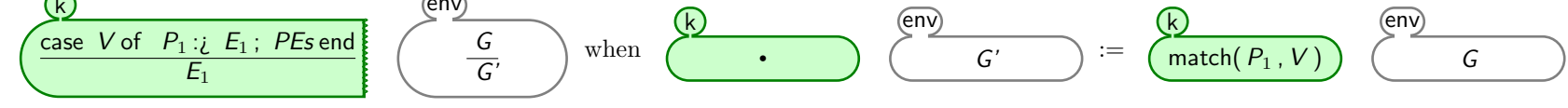
EQUATION:



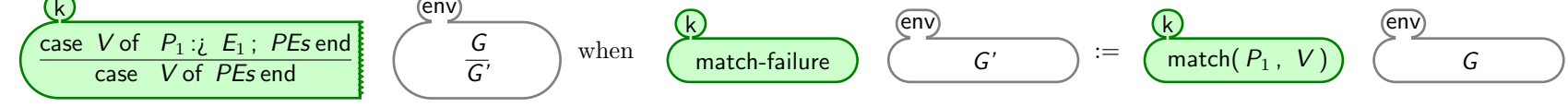
EQUATION:



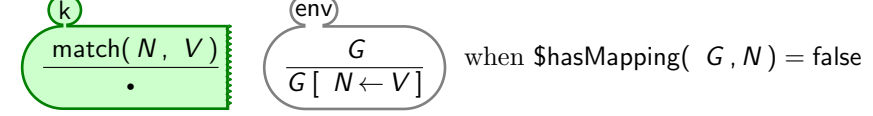
EQUATION:



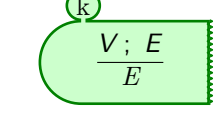
EQUATION:



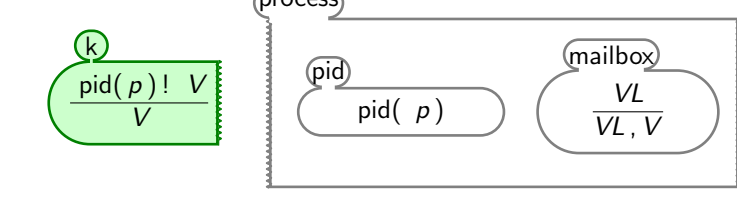
EQUATION:



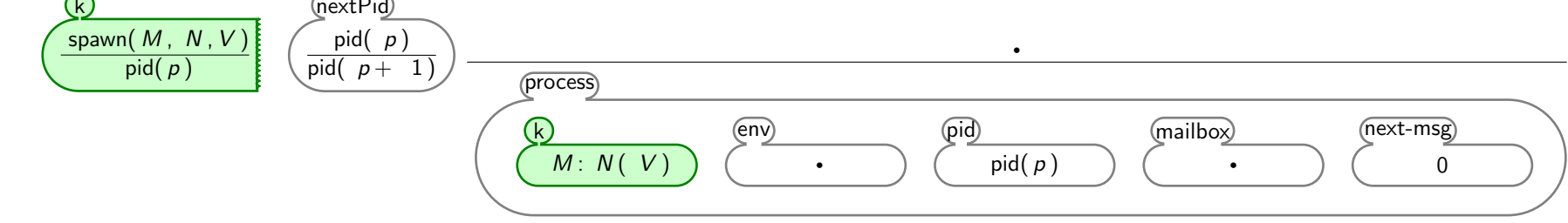
RULE:



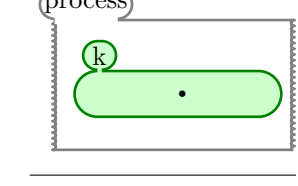
RULE:



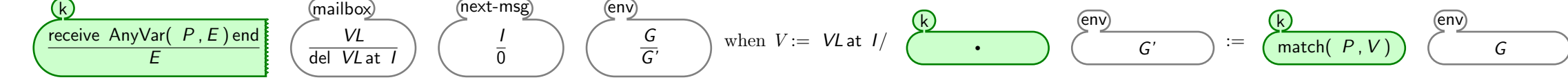
RULE:



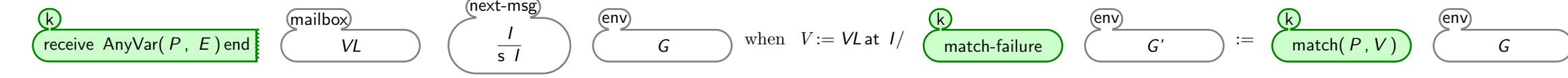
RULE:



RULE:



RULE:



EQUATION:  $N() = N[[]]$

EQUATION:  $N[\cdot] = T = \text{AnyVar}(N, [], T)$

EQUATION:  $ES, ES' = ES, ES'$

EQUATION:  $PS, PS' = PS, PS'$

EQUATION:  $V, VL \text{ at } 0 = V$

EQUATION:  $V, VL \text{ at } s \neq VL \text{ at } I$

EQUATION:  $[T] = [T - \{\cdot\}]$

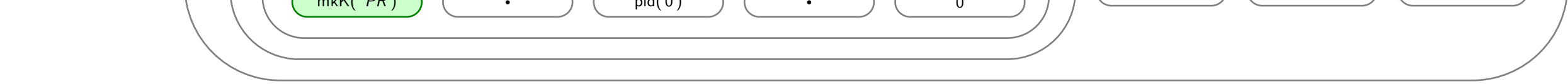
EQUATION:  $[T, T'] = [T - \{\cdot\}, T']$

EQUATION:  $ES, ES' = ES, ES'$

EQUATION:  $\text{del } \cdot \text{ at } I = \cdot$

EQUATION:  $\text{del } V, VL \text{ at } 0 = VL$

EQUATION:  $\text{del } V, VL \text{ at } s \neq V, \text{del } VL \text{ at } I$



END MODULE