Module Ertest

imports GENERIC-EXP-SEMANTICS

syntactic constructs:

Attr ::=
| *Attr Attr* [renameTo _↷_]
| -export([*Fun*]).
| -import(*#Name* ,[*Fun*]).

Cases ::=
| *Cases* ; *Cases*
| *Pattern* ⇝ *Exps*

Decl ::=
| *FunCases* .
| *Decl Decl* [renameTo _↷_]

ETerm ::= *#Bool*|*#Int*| *#Name*|*#String*
| []
| [*ETerm* — *ETerm*] [strict]

Exp ::= *Pattern*|*#Bool*| *#Int*|*#Name*
|
| self()
| *#Name* ()
| [*Exps*] [aux]
| *Exps* [strict]
| iff *GuardExps* end
| receive *Cases* end
| length(*Exp*) [strict]
| tuple-size(*Exp*) [strict]
| *Exp* ! *Exp* [strict]
| *Pattern* = *Exp* [strict(2)]
| *#Name* (*Exp*)
| [*Exp* — *Exp*] [strict]
| case *Exp* of *Cases* end [strict(1)]
| element(*Exp* , *Exp*) [strict]
| *#Name* : *#Name* (*Exp*) [strict(3)]
| setelement(*Exp* , *Exp* , *Exp*) [strict]
| spawn(*Exp* , *Exp* , *Exp*) [strict]

Exps ::= *Exp*|*Patterns*
| *Exps* ; *Exps* [strict(1)]
| *Exps* , *Exps* [aux]
| *Exps* , , *Exps* [strict]

Fun ::=
| *Fun* , *Fun*
| *#Name* // *#Int*

FunCases ::=
| *FunCases* ; *FunCases* [renameTo _↷_]
| *#Name* ()⇒ *Exps*
| *#Name* (*Pattern*)⇒ *Exps*

GuardExps ::=
| *Guard* :> *Exps* [strict(1)]
| *GuardExps* ; *GuardExps*

Mod ::=
| *Mod Mod* [renameTo _↷_]
| -module(*#Name*). *Attr*
| -module(*#Name*). *Decl*
| -module(*#Name*). *Attr Decl*

Pattern ::= *ETerm*
| *Patterns*
| pid(*#Name*)
| [*Pattern* — *Pattern*] [strict]

Patterns ::= *Pattern*
| *Patterns* ; *Patterns*
| *Patterns* , *Patterns* [aux]

Prgm ::= *Mod*|*Stmt*
| *Mod Stmt* [renameTo _↷_]

Stmt ::=
| *Exp* . [strict]
| *Stmt Stmt* [renameTo _↷_]
| match(*Pattern* , *Exp*) [strict(2)]

semantic constructs:

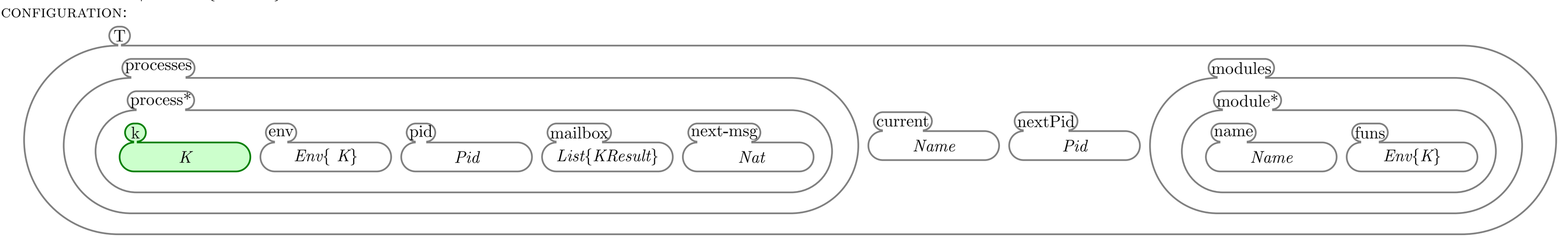
Config ::=
| run(*Prgm*)

K ::=
| match-failure
| check-msg(*K*)

KResult ::=
| *List*{ *KResult*} at *Nat*

List{*KResult*} ::=
| del *List*{*KResult*} at *Nat*

configuration:



K equations and rules:

equation:

k: $$\frac{[]}{[]}$$

equation:

k: $$-$$

equation:

k: $$\frac{V\ .}{\bullet}$$

equation:

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

equation:

k: $$\frac{@.(\bullet)}{\bullet}$$

equation:

k: $$\frac{V\ ,,\ V'}{V\ ,,\ V'}$$

equation:

k: $$\frac{[\ V —\ V']}{[\ V —\ V']}$$

equation:

k: $$\frac{V}{V}$$

equation:

k: $$\frac{\text{length}(\ []\ )}{0}$$

equation:

k: $$\frac{\text{length}(\ [\ V —\ V']\ )}{\text{length}(\ V'\ ) +\ 1}$$

equation:

k: $$\frac{\text{match}(\ V\ , V\ )}{\bullet}$$

equation:

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

equation:

k: $$\frac{\text{match}(\ [H —\ T\ ],\ [VH —\ VT\ ]\ )}{\text{match}(\ H\ , VH\ ) \curvearrowright \text{match}(\ T\ , VT\ )}$$

equation:

k: $$\frac{\text{match}(\ \text{pid}(\ N\ ), \text{pid}(\ p\ )\ )}{\text{match}(\ N\ ,\ \text{pid}(\ p\ )\ )}$$

equation:

k: $$\frac{\text{receive}\ \ PEs\ \text{end}}{\text{check-msg}(\ PEs\ ) \curvearrowright \text{receive}\ \ PEs\ \text{end}}$$

EQUATION:

k: $\dfrac{\text{match}(V, V') \curvearrowright Rest}{\text{match-failure}}$

EQUATION:

k: $\dfrac{N(E)}{M:\ N(E)}$ current: M

EQUATION:

k: $\dfrac{N}{V}$ env: $N \mapsto V$

EQUATION:

k: $\dfrac{\text{self}()}{\text{pid}(p)}$ pid: $\text{pid}(p)$

EQUATION:

k: $\dfrac{\text{-module}(N).\ D}{D}$ current: $\dfrac{—}{N}$ modules: $\dfrac{\bullet}{\text{module: (name: } N \text{) (funs: } \bullet \text{)}}$

EQUATION:

k: $\dfrac{M:\ N(V)}{\text{match}(P, V) \curvearrowright E \curvearrowright \text{restore}(G)}$ name: M funs: $N \mapsto ¡\ P, E\ ¿$ env: $\dfrac{G}{\bullet}$

EQUATION:

k: $\dfrac{N(P) \Rightarrow E}{\bullet}$ current: M name: M funs: $\dfrac{G}{G[N \leftarrow ¡\ P, E\ ¿]}$

EQUATION:

k: $\dfrac{\text{match}(N, V)}{\bullet}$ env: $N \mapsto V$

EQUATION:

process: (k: $\dfrac{\text{iff } Gr :> E \text{ end}}{E}$ $Cfgs$) when process: (k: true $Cfgs$) := process: (k: Gr $Cfgs$)

EQUATION:

process: (k: $\dfrac{\text{iff } Gr :> E;\ GEs \text{ end}}{E}$ $Cfgs$) when process: (k: true $Cfgs$) := process: (k: Gr $Cfgs$)

EQUATION:

process: (k: $\dfrac{\text{iff } Gr :> E;\ GEs \text{ end}}{\text{iff } GEs \text{ end}}$ $Cfgs$) when process: (k: false $Cfgs$) = process: (k: Gr $Cfgs$)

EQUATION:

k: $\dfrac{\text{case } V \text{ of } P_1 \rightsquigarrow E_1;\ PEs \text{ end}}{E_1}$ env: $\dfrac{G}{G'}$ when k: $\bullet$ env: G' := k: $\text{match}(P_1, V)$ env: G

EQUATION:

k: $\dfrac{\text{case } V \text{ of } P_1 \rightsquigarrow E_1;\ PEs \text{ end}}{\text{case } V \text{ of } PEs \text{ end}}$ env: $\dfrac{G}{G'}$ when k: match-failure env: G' := k: $\text{match}(P_1, V)$ env: G

EQUATION:

k: $\dfrac{\text{case } V \text{ of } P_1 \rightsquigarrow E_1 \text{ end}}{E_1}$ env: $\dfrac{G}{G'}$ when k: $\bullet$ env: G' := k: $\text{match}(P_1, V)$ env: G

EQUATION:

k: $\dfrac{\text{match}(N, V)}{\bullet}$ env: $\dfrac{G}{G[N \leftarrow V]}$ when \$hasMapping$(G, N) = \text{false}$

RULE:

k: $\dfrac{V;\ E}{E}$

RULE:

k: $\dfrac{\text{pid}(p)!\ V}{V}$ process: (pid: $\text{pid}(p)$ mailbox: $\dfrac{VL}{VL, V}$)

RULE:

k: $\dfrac{\text{spawn}(M, N, V)}{\text{pid}(p)}$ nextPid: $\dfrac{\text{pid}(p)}{\text{pid}(p + 1)}$ $\dfrac{\bullet}{\text{process: (k: } M:\ N(V) \text{) (env: } \bullet \text{) (pid: pid}(p)\text{) (mailbox: } \bullet \text{) (next-msg: } 0\text{)}}$

RULE:

$\dfrac{\text{process: (k: } \bullet \text{)}}{\bullet}$

RULE:

k: $\dfrac{\text{check-msg}(P \rightsquigarrow E;\ PEs') \curvearrowright \text{receive } PEs \text{ end}}{E}$ mailbox: $\dfrac{VL}{\text{del } VL \text{ at } I}$ next-msg: $\dfrac{I}{0}$ env: $\dfrac{G}{G'}$ when $V := VL \text{ at } I /$ k: $\bullet$ env: G' := k: $\text{match}(P, V)$ env: G

RULE:

k: $\dfrac{\text{check-msg}(P \rightsquigarrow E) \curvearrowright \text{receive } PEs \text{ end}}{E}$ mailbox: $\dfrac{VL}{\text{del } VL \text{ at } I}$ next-msg: $\dfrac{I}{0}$ env: $\dfrac{G}{G'}$ when $V := VL \text{ at } I /$ k: $\bullet$ env: G' := k: $\text{match}(P, V)$ env: G

RULE:

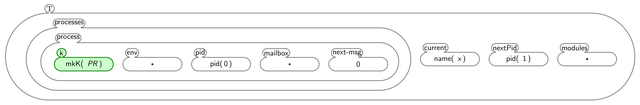
k: $\dfrac{\text{check-msg}(P \rightsquigarrow E;\ PEs)}{\text{check-msg}(PEs)}$ mailbox: VL next-msg: I env: G when $V := VL \text{ at } I /$ k: match-failure env: G' := k: $\text{match}(P, V)$ env: G

RULE:

k: $\dfrac{\text{check-msg}(P \rightsquigarrow E)}{\bullet}$ mailbox: VL next-msg: $\dfrac{I}{\text{s } I}$ env: G when $V := VL \text{ at } I /$ k: match-failure env: G' := k: $\text{match}(P, V)$ env: G

EQUATION: $N() = N([\,])$
EQUATION: $N() \Rightarrow T = N([]) \Rightarrow T$
EQUATION: $ES, ES' = ES; ES'$
EQUATION: $PS, PS' = PS; PS'$
EQUATION: $V, VL \text{ at } 0 = V$
EQUATION: $V, VL \text{ at s } I = VL \text{ at } I$
EQUATION: $[T] = [T — [\,]]$
EQUATION: $[T; T'] = [T — [T']]$
EQUATION: $ES; ES' = ES ,, ES'$
EQUATION: $\text{del } \bullet \text{ at } I = \bullet$
EQUATION: $\text{del } V, VL \text{ at } 0 = VL$
EQUATION: $\text{del } V, VL \text{ at s } I = V, \text{del } VL \text{ at } I$

EQUATION: run(PR) =



END MODULE