

AGENT

MODULE BASIC-EXP-SYNTAX

SYNTAX $Exp ::= (Exp) \text{ [strict]}$

END MODULE

MODULE VAL

SYNTAX $Exp ::= Val$

SYNTAX $KResult ::= Val$

END MODULE

MODULE BOOL-EXP-SYNTAX

SYNTAX $Exp ::= Bool$

END MODULE

MODULE BOOL-EXP

SYNTAX $Val ::= Bool$

END MODULE

MODULE INT-EXP-SYNTAX

SYNTAX $Exp ::= Int$

END MODULE

MODULE INT-EXP

SYNTAX $Val ::= Int$

END MODULE

MODULE EXP-SYNTAX

SYNTAX $Exp ::= Exp * Exp \text{ [mul, strict]}$
 $Exp / Exp \text{ [div, strict]}$
 $Exp + Exp \text{ [plus, strict]}$
 $Exp \leq Exp \text{ [eq, isstrict]}$
 $Exp == Exp \text{ [eq, strict]}$
 $\text{not } Exp \text{ [not, strict]}$
 $Exp \text{ and } Exp \text{ [and, strict(1)]}$

END MODULE

MODULE EXP

RULE
$$\frac{I1 * I2}{I1 *_{mul} I2}$$

RULE
$$\frac{I1 + I2}{I1 +_{plus} I2}$$

RULE
$$\frac{I1 / I2}{I1 /_{div} I2} \text{ requires } I2 \neq_{int} 0$$

RULE
$$\frac{I1 \leq I2}{I1 \leq_{leq} I2}$$

RULE
$$\frac{V1 == V2}{V1 ==_N V2}$$

RULE
$$\frac{\text{not } T}{\neg_{Bool} T}$$

RULE
$$\frac{\text{true and } E}{E}$$

RULE
$$\frac{\text{false and } E}{\text{false}}$$

END MODULE

MODULE IF-SYNTAX

SYNTAX $Exp ::= \text{if } Exp \text{ then } Exp \text{ else } Exp \text{ [if, strict(1)]}$

END MODULE

MODULE IF

RULE
$$\frac{\text{if true then } E \text{ else } _}{E}$$

RULE
$$\frac{\text{if false then } _ \text{ else } E}{E}$$

END MODULE

MODULE ID-EXP-SYNTAX

SYNTAX $Exp ::= Id$

END MODULE

MODULE LAMBDA-SYNTAX

SYNTAX $Lambda ::= \lambda Id. Exp \text{ [lam, binder]}$
SYNTAX $Exp ::= Exp \text{ Exp [app, strict]}$
 $Lambda$

END MODULE

MODULE LAMBDA

SYNTAX $Val ::= Id$
 $Lambda$

RULE
$$\frac{(\lambda X. E) V}{E[V / X]}$$

END MODULE

MODULE MU-SYNTAX

SYNTAX $Exp ::= \mu Id. Exp \text{ [mu, binder]}$

END MODULE

MODULE MU

RULE
$$\frac{(\mu X. E)}{E[(\mu X. E) / X]}$$

END MODULE

MODULE CALLCC-SYNTAX

SYNTAX $Exp ::= \text{callcc } Exp \text{ [callCC, strict]}$

END MODULE

MODULE CALLCC

SYNTAX $Val ::= cc(K)$

RULE
$$\frac{\text{callcc } (V) \wedge K}{(V \text{ cc}(K))}$$

RULE
$$\frac{(\text{cc}(K) V) \wedge _}{V \wedge K}$$

END MODULE

MODULE HALT-SYNTAX

SYNTAX $Exp ::= \text{halt } Exp \text{ [strict]}$

END MODULE

MODULE HALT

RULE
$$\frac{(\text{halt } V) \wedge _}{V}$$

END MODULE

MODULE SEQ-SYNTAX

SYNTAX $Exp ::= \text{skip}$
 $Exp ; Exp \text{ [seq, strict(1)]}$

END MODULE

MODULE SEQ

SYNTAX $Val ::= \text{skip}$

RULE
$$\frac{V ; S}{S}$$

[structural]

END MODULE

MODULE IO-SYNTAX

SYNTAX $Exp ::= \text{read} \text{ [read]}$
 $\text{print } Exp \text{ [pm, strict]}$

END MODULE

MODULE IO

CONFIGURATION:

RULE

RULE

END MODULE

MODULE REF-SYNTAX

SYNTAX $Exp ::= \text{ref } Exp \text{ [ref, strict]}$
 $* Exp \text{ [dref, strict]}$
 $Exp := Exp \text{ [assign, strict(2)]}$

END MODULE

MODULE REF

CONFIGURATION:

CONTEXT $* \square := _$

RULE

RULE

[transition]

RULE

[transition]

END MODULE

MODULE WHILE-SYNTAX

SYNTAX $Exp ::= \text{while } Exp \text{ do } Exp \text{ [while]}$

END MODULE

MODULE WHILE

RULE
$$\frac{\text{while } E \text{ do } S}{\text{if } E \text{ then } (S ; \text{while } E \text{ do } S) \text{ else skip}}$$

END MODULE

MODULE THREADS-SYNTAX

SYNTAX $Exp ::= \text{acquire } Exp \text{ [acq, strict]}$
 $\text{release } Exp \text{ [rel, strict]}$
 $\text{rendezvous } Exp \text{ [rmdv, strict]}$
 $\text{spawn } Exp \text{ [spwn]}$

END MODULE

MODULE THREADS

CONFIGURATION:

RULE

[transition]

RULE

[transition]

RULE

[transition]

RULE

RULE

[transition]

END MODULE

MODULE AGENTS-SYNTAX

SYNTAX $Exp ::= \text{newAgent } Exp \text{ [newAg]}$
 me [me]
 parent [parent]
 receive [rcv]
 $\text{receiveFrom } Exp \text{ [rcvFr, strict]}$
 $\text{send } Exp \text{ to } Exp \text{ [sndTo, strict]}$
 $\text{sendSynch } Exp \text{ to } Exp \text{ [sndSyn, strict]}$
 barrier [bar]
 $\text{broadcast } Exp \text{ [bcast, strict]}$
 $\text{haltAgent [haltAg]}$

END MODULE

MODULE AGENTS

CONFIGURATION:

RULE

RULE

[transition]

RULE

RULE

RULE

RULE

[transition]

RULE

[transition]

RULE

RULE

[structural]

RULE

[transition]

RULE

[transition]

RULE

[transition]

RULE

[transition]

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE

RULE