

AGENT

MODULE EXP-SYNTAX

SYNTAX $K ::= Bool$
 | Int
 | $[K]$
 | $K \wedge K$ [strict]
 | $K \vee K$ [strict]
 | $K \rightarrow K$ [strict]
 | $K \leq K$ [sequint]
 | $K == K$ [strict]
 | $not\ K$ [strict]
 | $K\ and\ K$ [strict1]

END MODULE

MODULE EXP

SYNTAX $KResult ::= Bool$
 | Int

RULE $\frac{I1 + I2}{I1 +_{res} I2}$
RULE $\frac{I1 + I2}{I1 +_{val} I2}$
RULE $\frac{I1 \neq I2}{I1 \neq_{res} I2}$ when $I1 \neq_{val} I2 = 0$
RULE $\frac{I1 \leq I2}{I1 \leq_{res} I2}$
RULE $\frac{V1 == V2}{V1 ==_{res} V2}$
RULE $\frac{not\ T}{T ==_{res} F}$
RULE $\frac{true\ and\ E}{E}$
RULE $\frac{false\ and\ E}{false}$

END MODULE

MODULE IF-SYNTAX

SYNTAX $K ::= Bool$
 | $if\ K\ then\ K\ else\ K$ [strict1]

END MODULE

MODULE IF

SYNTAX $KResult ::= Bool$

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

END MODULE

MODULE LAMBDA-SYNTAX

SYNTAX $K ::= \lambda x. E$
 | $[K]$
 | $K\ K$ [strict]
 | $MId.K$ [binder]

END MODULE

MODULE LAMBDA

SYNTAX $KResult ::= Id$
 | $MId.K$ [binder]

RULE $\frac{\lambda x. E\ E\ V}{E[V/x]}$

END MODULE

MODULE MU-SYNTAX

SYNTAX $K ::= \mu Id. K$ [binder]

END MODULE

MODULE MU

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

END MODULE

MODULE CALLCC-SYNTAX

SYNTAX $K ::= callcc\ K$ [strict]
 | $\alpha(K)$

END MODULE

MODULE CALLCC

SYNTAX $KResult ::= \alpha(K)$

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

END MODULE

MODULE HALT-SYNTAX

SYNTAX $K ::= halt\ K$ [strict]

END MODULE

MODULE HALT

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

END MODULE

MODULE SEQ-SYNTAX

SYNTAX $K ::= skip$
 | $K ; K$ [strict1]

END MODULE

MODULE SEQ

SYNTAX $KResult ::= skip$

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

END MODULE

MODULE IO-SYNTAX

SYNTAX $K ::= Int$
 | $read$
 | $print\ K$ [strict]

END MODULE

MODULE IO

CONFIGURATION:

RULE $\frac{\text{read } V}{V}$

RULE $\frac{\text{print } V\ skip}{V}$

END MODULE

MODULE REF-SYNTAX

SYNTAX $K ::= Nil$
 | $ref\ K$ [strict]
 | $*K$ [strict]
 | $K := K$ [strict2]

END MODULE

MODULE REF

CONFIGURATION:

CONTEXT $* \square := _$

RULE $\frac{\text{ref } V}{N}$

RULE $\frac{*N}{V}$

RULE $\frac{*N := V\ skip}{V}$

END MODULE

MODULE WHILE-SYNTAX

SYNTAX $K ::= while\ K\ do\ K$

END MODULE

MODULE WHILE

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

END MODULE

MODULE THREADS-SYNTAX

SYNTAX $K ::= Nil$
 | $acquire\ re\ K$ [strict]
 | $release\ K$ [strict]
 | $rendezvous\ K$ [strict]
 | $spawn\ K$

END MODULE

MODULE THREADS

CONFIGURATION:

RULE $\frac{\text{spawn } S\ skip}{S}$

RULE $\frac{\text{acquire } V\ skip}{V := 0}$

RULE $\frac{\text{release } V\ skip}{V := N}$

RULE $\frac{\text{rendezvous } V\ skip}{rendezvous\ V}$

MODULE AGENTS-SYNTAX

SYNTAX $K ::= Int$
 | $Bool$
 | $newAgent\ K$
 | me
 | $parent$
 | $receive$
 | $receiveFrom\ K$ [strict]
 | $send\ K\ to\ K$ [strict]
 | $sendsynch\ K\ to\ K$ [strict]
 | $barrier$
 | $broadcast\ K$ [strict]
 | $haltAgent$

END MODULE

MODULE AGENTS

CONFIGURATION:

RULE $\frac{\text{newAgent } S}{N2}$

RULE $\frac{\text{me } N}{N}$

RULE $\frac{\text{parent } N}{N}$

RULE $\frac{\text{send } V\ to\ N2\ skip}{N2}$

RULE $\frac{\text{receive } V}{N2}$

RULE $\frac{\text{receiveFrom } N2\ V}{N2}$

RULE $\frac{\text{broadcast } V\ skip}{W}$

RULE $\frac{\text{barrier } true}{W}$

RULE $\frac{\text{barrier } false}{W}$

RULE $\frac{\text{barrier } skip}{W}$

RULE $\frac{\text{barrier } true}{W}$

MODULE QUOTE-UNQUOTE-SYNTAX

SYNTAX $K ::= quote\ K$
 | $unquote\ K$
 | $lift\ K$ [strict]
 | $eval\ K$ [strict]

END MODULE

MODULE QUOTE-UNQUOTE

SYNTAX $K ::= isQuote\ (K)$ [labelInQuote]

RULE $\frac{isQuote\ (quote\ K)}{true}$

RULE $\frac{isQuote\ (unquote\ K)}{true}$

SYNTAX $K ::= quote\ (K, N2)$ [labelInQuote]

RULE $\frac{quote\ (K, N2)}{apply\ quote_{N2}\ in\ K\ if\ isQuote}$

SYNTAX $KLabel ::= quote_{N2}$

RULE $\frac{quote\ K}{quote\ (K, 0)}$

RULE $\frac{quote_{N2}\ (quote\ K)}{quote_{N2}\ (quote\ (K, N2 +_{res} 1))}$

RULE $\frac{quote_{N2}\ (unquote\ K)}{K}$

RULE $\frac{quote_{N2}\ (unquote\ K)}{unquote_{N2}\ (quote\ (K, N2 +_{res} 1))}$ when $N2 >_{res} 0$

RULE $\frac{lift\ V}{V}$

RULE $\frac{eval\ K}{K}$

END MODULE

MODULE AGENT-SYNTAX

END MODULE

MODULE AGENT

CONFIGURATION:

RULE $\frac{\text{SPICAM}}{M}$

END MODULE