Thesis Notation

March 28, 2011

Table 1: Types

Macro	Output	Description
`\type{0}`	$\mathsf{0}$	Type macro
`\sst{0}{1}`	0_1	Subscripted type
`\Talpha`	Σ	Alphabet
`\Tholes`	X	Hole Alphabet
`\Thvars`	Θ	Hole Variable Alphabet
`\Tpvars`	P	Propositional variables
`\Taddr`	Addr	Heap addresses
`\Tval`	Val	Values
`\Tralph`	Υ	Ranked alphabet
`\Tcalph`	Ω	$\Sigma \cup \mathrm{X}$
`\Ttree`	Tree	Trees
`\Tseq`	Seq	Sequences
`\Theap`	Heap	Heaps
`\Tterm`	Term	Terms
`\Tutree`	UTree	Uniquely-labelled trees
`\Tword`	Ω^*	Words
`\Tfor`	$\mathsf{Forest}_{\Sigma,\mathrm{X}}$	Forests
`\Tlist`	Lst	Lists
`\Tlsto`	LStore	List stores
`\Tdata`	D	Abstract states
`\Tcont`	C	Contexts
`\Ttcont`	$\mathsf{C}_{\mathsf{Tree}}$	Tree Contexts
`\Tscont`	$\mathsf{C}_{\mathsf{Seq}}$	Sequence Contexts
`\Ttercont`	$\mathsf{C}_{\mathsf{Term}}$	Term Contexts
`\Tutcont`	$\mathsf{C}_{\mathsf{UTree}}$	Uniquely-labelled tree contexts
`\Tmcont`	C^m	Multi-holed Contexts
`\Tmtcont`	$\mathsf{C}^m_{\mathsf{Tree}}$	Multi-holed Tree Contexts
`\Tmscont`	$\mathsf{C}^m_{\mathsf{Seq}}$	Multi-holed Sequence Contexts
`\Tmtercont`	$\mathsf{C}^m_{\mathsf{Term}}$	Multi-holed Term Contexts

Table 1: Types

Macro	Output	Description
\TLEnv	LEnv	Logical environments
\TInterp	Interp	Propositional interpretations
\Tworld	World	Worlds
\Tsort	Sort	Sorts
\Tformula	$\mathsf{Formula}$	Formulae
\Trank	Rank	Ranks
\DW	DW	Duplicator-winning games
\SW	SW	Spoiler-winning games
Mathematics		
\Tnat	$\mathbb{N}$	Natural numbers (includes 0)
\Tint	$\mathbb{Z}$	Integers
\Tbool	Bool	Booleans
Formulae		
\TF	F	(Propositional) formulae
\TFd	P	Data formulae
\TFc	K	Context Formulae
\TFdt	$\mathsf{P}_{\mathsf{Tree}}$	Tree
\TFct	$\mathsf{K}_{\mathsf{Tree}}$	Tree context
\TFsd	P^{s}	Specifically single holed
\TFsc	K^{s}	
\TFsdt	$\mathsf{P}^{s}_{\mathsf{Tree}}$	
\TFsct	$\mathsf{K}^{s}_{\mathsf{Tree}}$	
\TFcd	P^{c}	With composition
\TFcc	K^{c}	
\TFcdt	$\mathsf{P}^{c}_{\mathsf{Tree}}$	
\TFcct	$\mathsf{K}^{c}_{\mathsf{Tree}}$	
\TFm	K^{m}	Multi-holed
\TFmt	$\mathsf{K}^{m}_{\mathsf{Tree}}$	
\TFms	$\mathsf{K}^{m}_{\mathsf{Seq}}$	
\TFmter	$\mathsf{K}^{m}_{\mathsf{Term}}$	
Programs		
\Tprog	Prog	Programs
\Tcommand	Cmd	Basic command
\Tpgvar	Var	Program variables
\Tscope	Scope	Variable scope
\Tstack	Stack	Variable stack
\Tstore	Store	Data store
\Tstate	State	Program state
\Toutcome	$\mathsf{Outcome}$	Outcomes
\Tpname	PName	Procedure names

Table 1: Types

Macro	Output	Description
\Tpdef	PDef	Procedure definition environment
\Texpr	Expr	Expressions
\Tbexp	BExp	Boolean expressions
CAP		
\TLVar	LVar	Logical variables
\TLState	LState	Logical state
\TSState	SState	Shared state
\TPerm	Perm	Permission
\TTok	Token	Token
\TRid	RID	Region identifier
\TAName	AName	Action name
\TAction	Action	Action
\TAMod	AMod	Action model
\TAPName	APName	Abstract predicate name
\Tassn	Assn	Assertion
\Tbassn	BAssn	
\Tiassn	IAssn	
\Tfassn	FAssn	
\Tinterp	Interp	
\Tpenv	PEnv	
\Tfenv	FEnv	

Table 2: Variables

Macro	Output	Description
\la	$\mathbf{a}$	Letters of Σ
\lb	$\mathbf{b}$	
\lc	$\mathbf{c}$	
\hx	x	Holes of X
\hy	y	
\hz	z	
\hxP	$\acute{x}$	
\hyP	$\acute{y}$	
\hzP	$\acute{z}$	
\hva	α	Variables of Θ
\hvb	β	
\hvc	γ	
\hpa	a	Heap addresses
\va	v	Values

Table 2: Variables

Macro	Output	Description
\vb	w	
\vc	u	
\pvp	p	Propositional variables
\pvq	q	
\tr	t	Trees
\sq	s	Sequences
\hp	h	Heaps
\tm	r	Terms
\lst	l	Lists
\lstc	lc	
\ls	ls	
\lsc	lsc	
\word	w	Words (confusing with worlds?)
\state	s	Abstract state
\cont	c	Contexts
\contd	d	
\cidc	i	Element of $\mathbf{I}$
\lenv	σ	Logical Environments of LEnv
\interp	ι	Propositional Interpretation of Interp
\ca	$\mathcal{A}$	Context Algebra
\sor	ς	Sort
\rank	r	Rank
\rn	m	Non-adjunct component
\rs	s	Adjunct component
\rL	L	Label component
\rV	V	Variable component
\world	w	World
\swor	w^{S}	Spoiler's world
\dwor	w^{D}	Duplicator's world
\opvar	$\circledast$	Operator
\tv	b	Truth value
Programs		
\cmd	$\mathbb{C}$	Program
\bcmd	φ	Basic command
\pdef	μ	Procedure definition
\penv	γ	Procedure definition environment

Table 2: Variables

Macro	Output	Description
`\stack`	s	Stack
`\scope`	ρ	Scope
`\stor`	χ	Store
`\expr`	E	Expression
`\bexp`	B	Boolean expression
`\oc`	o	Outcome
`\pse`	Γ	Procedure specification environment
Logical formulae		
`\Fp`	F	Propositional formula
`\Fdv`	P	Data formula
`\Fdva`	Q	
`\Fdvb`	R	
`\Fcv`	K	Context formula
`\Fo`	P	Arbitrary formula
`\FSd`	S_{P}	Specific data formula
`\FSc`	S_{K}	
`\FS`	S	
Abstract modules		
`\ama`	$\mathbb{A}$	Abstract module
`\amb`	$\mathbb{B}$	
CAP		
`\lstate`	l	Logical state
`\sstate`	s	Shared state
`\rid`	r	Region identifier
`\aname`	γ	Action name
`\perm`	ϕ	Permission
`\amod`	ζ	Action model
`\tok`	t	Token
`\action`	a	Action
`\apname`	α	Abstract predicate name
`\ias`	I	Interference assertion
`\ridexp`	R	Region expression
`\perexp`	π	Permission expression
`\apenv`	δ	Abstract Predicate environment

Table 3: Constants

Macro	Output	Description
`\nullref`	nil	null value
`\empt`	$\varnothing$	Empty tree
`\hole`	$-$	Context hole
`\truth`	$\mathbf{T}$	Truth
`\falsity`	$\mathbf{F}$	

Table 4: Operators

Macro	Output	Description
`\tpar`	$\otimes$	Tree parallel
`\tlab{0}{1}`	$0[1]$	Tree label
`\scat`	$\cdot$	Sequence concatenation
`\cell{0}{1}`	$0 \mapsto 1$	Heap cell
`\hsep`	$*$	Heap combination
`\hemp`	emp	Empty heap
`\lcell{0}{1}`	$0 \Mapsto 1$	List cell
`\lcellc{0}{1}`	$0 \Mapsto [1]$	
`\app`	$\bullet$	Context application
`\comp`	$\circ$	Context composition
`\mcc{0}`	$\textcircled{0}$	Multi-holed composition
`\subs{0}{1}`	$[0/1]$	Substitution
`\cid`	$\mathbf{I}$	Identity of a context algebra
`\czero`	$\mathbf{0}$	Zero of a context algebra
`\holes`	holes	Holes function
`\dom`	dom	
`\range`	range	
`\freevars`	fv	
`\fresh`	$\sharp$	Fresh
Mathematics		
`\pfun`	$\rightharpoonup$	Partial function
`\fpfun`	$\rightharpoonup_{\text{fin}}$	Finite partial function
`\fupd{0}{1}{2}`	$0[1 \mapsto 2]$	(Partial) function update
`\powset{0}`	$\mathcal{P}(0)$	Powerset
`\fpowset{0}`	$\mathcal{P}_{\text{fin}}(0)$	
`\Set{0}`	$\{0\}$	Set
`\Setb{0}{1}`	$\{0 \mid 1\}$	Set builder
`\union`	$\cup$	Union
`\intersect`	$\cap$	Intersection
`\Card{0}`	$\lvert 0 \rvert$	Cardinality

Table 4: Operators

Macro	Output	Description
`\disjunion`	$\uplus$	Disjoint union
`\Family{0}{1}{2}`	$\{0_2\}_{2\in 1}$	Family
`\family{0}{1}{2}`	$\{0\}_{1\in 2}$	Alternate macro
`\randthen`	$\fatsemi$	Relation composition
Logical formulae		
`\Ftzero`	$\mathbf{0}$	Tree zero
`\Ftpar`	$\otimes$	Tree parallel
`\Ftlab{0}{1}`	$0[1]$	Tree label
`\Fscat`	$\cdot$	Sequence concatenation
`\Fapp`	$\bullet$	Application
`\Fappr`	$\bullet\!\!-$	Application adjoint (tree formula)
`\Frapp`	$-\!\!\bullet$	Application adjoint (context formula)
`\FapprE`	$\bullet\!\!-^{\exists}$	Existential versions of adjoints
`\FrappE`	$-\!\!\bullet^{\exists}$	
`\Fid`	$\mathbf{I}$	Context identity
`\Fcomp`	$\circ$	Composition
`\Fcompr`	$\circ\!\!-$	Composition adjoints
`\Frcomp`	$-\!\!\circ$	
`\FcomprE`	$\circ\!\!-^{\exists}$	Existential versions of adjoints
`\FrcompE`	$-\!\!\circ^{\exists}$	
`\Fimp`	$\rightarrow$	Material implication
`\Fand`	$\wedge$	Conjunction
`\For`	$\vee$	Disjunction
`\Fnot`	$\neg$	Negation
`\Fiff`	$\leftrightarrow$	Material equivalence
`\Fdtt`	true	Data true
`\Fdff`	false	Data false
`\Fctt`	True	Context true
`\Fcff`	False	Context false
`\Ftt`	$\top$	Propositional true
`\Fff`	$\bot$	Propositional false
`\new`	$\mathsf{И}$	Fresh quantification
`\Fcell{0}{1}`	$0 \mapsto 1$	Heap cell
`\Femp`	emp	
`\Fsep`	$*$	Separating conjunction
`\Fwand`	$-\!\!*$	Magic wand
`\FwandE`	$-\!\!*^{\exists}$	
`\Fstarprod{0}`	$\prod_0^*$	Star Product
`\bigoast`	$\circledast$	

Table 4: Operators

Macro	Output	Description
`\Fstarquant{0}`	$\circledast 0.$	Star Quantification
`\Fnoh{0}`	$\circledcirc$	No hole
`\Fahole`	$\mapsto$	Just a hole
`\Fswsc`	$\textcircled{\scriptsize R}$	Strongest weaker stable
`\Fswsb`	$\textcircled{\scriptsize R}$	
`\Fwssb`	$\boxed{\mathrm{R}}$	Weakest stronger stable
`\Fgdiab`	$\textcircled{\scriptsize G}$	
`\Fgdia`	$\textcircled{\scriptsize G}$	
`\Fsws`	$\textcircled{\scriptsize R}$	
`\Fwss`	$\boxed{\mathbf{R}}$	
`\shared{0}`	$\boxed{0}$	
`\Fshare{0}{1}{2}`	$\boxed{0}^1_2$	
`\shareds{0}`	$\boxed{0}$	
`\Fshares{0}{1}{2}`	$\boxed{0}^1_2$	
`\actquant`	$\exists$	
`\actto`	$\rightsquigarrow$	
`\actap{0}{1}{2}{3}{4}`	$0(1):\ \exists 2.\,(3 \rightsquigarrow 4)$	
`\acta{0}{1}{2}{3}`	$0(1):\ 2 \rightsquigarrow 3$	
`\actan{0}{1}{2}`	$0:\ 1 \rightsquigarrow 2$	
`\actanp{0}{1}{2}{3}`	$0:\ \exists 1.\,(2 \rightsquigarrow 3)$	
`\Fall`	all	
`\Fperm{0}{1}{2}{3}`	$[0(1)]^2_3$	
`\Fpermn{0}{1}{2}`	$[0]^1_2$	
`\upimp`	$\Longrightarrow$	
`\Fupimp{0}{1}`	$\Longrightarrow^{\{0\}\{1\}}$	
`\Fup{0}{1}`	$\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right\rangle$	
`\charac{0}{1}`	D^0_1	Rank-r characteristic of a world

Table 5: Relations

Macro	Output	Description
`\mods`	$\models$	Models
`\nmods`	$\not\models$	Doesn't model
`\entails`	$\subseteq$	Entailment?
`\lequiv`	$\equiv$	Equivalence
`\nlequiv`	$\not\equiv$	

Table 5: Relations

Macro	Output	Description
`\rankwfr`	$\lhd$	Rank well-founded relation
`\rankd{0}`	$\mathfrak{R}_0$	Rank constructor relation
`\move{0}`	$\mathfrak{M}_0$	Game move relation
`\spo`	$\leq$	Simulation preorder

Table 6: Misc

Macro	Output	Description
`\Gdef`	$::=$	BNF notation
`\Gbar`	$\mid$	
`\eqdef`	$\stackrel{\text{def}}{=}$	'is defined to be'
`\vect{0}`	$\overrightarrow{0}$	
`\sem{0}`	$[\![0]\!]$	Semantics
Meta syntax		
`\IFF`	$\iff$	If and only if
`\IMPLIES`	$\implies$	Implies
`\OR`	or	Or
`\AND`	and	And
`\NOT`	not	Not
`\EXISTS{0}`	there exists 0 s.t.	There exists
`\EXIST{0}`	there exist 0 s.t.	There exist
`\FORALL{0}`	for all 0,	For all
`\undef`	*undefined*	Undefined
`\blank`	$(\cdot)$	Blank
Logic notations		
`\CL`	CL	
`\CLs`	CL^s	
`\CLc`	CL^c	
`\CLm`	CL^m	
Text macros		
`\ie`	*i.e.*	
`\etc`	*etc.*	
`\etal`	*et al.*	
`\eg`	*e.g.*	
Games stuff		
`\Spoiler`	Spoiler	
`\Duplicator`	Duplicator	

Table 6: Misc

Macro	Output	Description
Sorts		
`\mksort{0}`	0	
`\Sdata`	d	
`\Scont`	c	

Table 7: Automata

Macro	Output	Description
`\Lang`	$\mathcal{L}$	Language
`\Langi{0}`	$\mathcal{L}_0$	
`\nil`	ε	Epsilon
`\lbl`	a	Label
`\lbm`	b	
`\aut`	$\mathcal{A}$	Automaton
`\States`	Q	State set
`\Accs`	A	Accepting states
`\st`	q	State
`\inist`	e	Initial state
`\tra{0}`	f^0	Transition relation
`\nilclose`	ε-closure	ε-closure
`\auti{0}`	$\mathcal{A}_0$	Automaton
`\Statesi{0}`	Q_0	State set
`\Accsi{0}`	A_0	Accepting states
`\sti{0}`	q_0	State
`\inisti{0}`	e_0	Initial state
`\trai{0}{1}`	f_1^0	Transition relation
`\nilclosei{0}`	ε-closure$_0$	ε-closure
`\hataut`	$\hat{\mathcal{A}}$	Partial automata
`\hatauti{0}`	$\hat{\mathcal{A}}_0$	
`\aint{0}`	$[\![0]\!]_{\mathcal{A}}$	Induced mappings
`\aintw{0}{1}`	$[\![0]\!]_1$	
`\afint{0}`	$(\!\|0\|\!)_{\mathcal{A}}$	
`\afintw{0}{1}`	$(\!\|0\|\!)_1$	
`\acompl{0}`	$\overline{0}$	Complementation
`\nlsub`	$\oslash$	Non-deterministic linear substitution
`\nllef`	$\oslash\!\!-^{\exists}$	
`\nlrig`	$-\!\!\oslash^{\exists}$	
`\usub`	$\odot$	Uniform substitution
`\ulef`	$\odot\!\!-^{\exists}$	
`\urig`	$-\!\!\odot^{\exists}$	

Table 7: Automata

Macro	Output	Description
\nusub	$\circledcirc$	
\nurig	$-\circledcirc^{\exists}$	
\reach	reachable	Reachability

Table 8: Programs

Macro	Output	Description
\psyntax{0}	0	Program syntax
\pvs{0}	*0*	Program variable
\pname{0}	0	Procedure name
Expressions		
\eP	+	
\eM	-	
\eT	*	
\eLT	<	
\eEQ	=	
\eIMP	=>	
\eFALSE	false	
Programs		
\pskip	skip	Skip
\pas{0}{1}	0 := 1	Assign
\seq	;	Sequential composition
\pifelse{0}{1}{2}	if 0 then 1 else 2	If-then-else

\pifelsev{0}{1}{2}	if 0 then
	1
	else
	2	

\pifv{0}{1}	if 0 then 1	
\pwhile{0}{1}	while 0 do 1	While-do
\pwhilev{0}{1}	while 0 do 1	
\procdef{0}{1}{2}{3}	0 := 1(2) {3}	Procdef

\procdefv{0}{1}{2}{3}	0 := 1(2) {
	3
	}	

\procdefvb{0}{1}{2}{3}	*0* := 1(2) {
	3
	}	

\procdefvc{0}{1}{2}	0(1) {
	2
	}	

Table 8: Programs

Macro	Output	Description
`\procs{0}{1}`	$\texttt{procs}\ 0\ \texttt{in}\ 1$	Procedures
`\procsv{0}{1}`	$\begin{array}{l}\texttt{procs}\\ \quad 0\\ \texttt{in}\\ \quad 1\end{array}$	
`\pcall{0}{1}{2}`	$\texttt{call}\ 0 := 1(2)$	Procedure call
`\pcallo{0}{1}`	$\texttt{call}\ 0(1)$	
`\plocal{0}{1}`	$\texttt{local}\ 0\ \texttt{in}\ 1$	Local variable declaration
`\plocalv{0}{1}`	$\begin{array}{l}\texttt{local}\ 0\ \texttt{in}\\ \quad 1\end{array}$	
`\alloc{0}{1}`	$0 := \texttt{alloc}(1)$	Allocate
`\dispose{0}{1}`	$\texttt{dispose}(0, 1)$	Dispose
`\sto{0}{1}`	$[0] := 1$	Store
`\fet{0}{1}`	$0 := [1]$	Fetch
Scope		
`\sceq`	$\Rrightarrow$	Scope value
`\scval{0}{1}`	$0 \Rrightarrow 1$	
`\scc`	$*$	Scope combination
`\scemp`	$\varnothing$	Empty scope
Semantics		
`\tosco{0}`	$\lfloor 0 \rfloor$	$\mathsf{Stack} \to \mathsf{Scope}$
`\esem{0}`	$\mathcal{E}\,[\![0]\!]$	$\mathsf{Expr} \to (\mathsf{Scope} \to \mathsf{Val})$
`\psem{0}`	$\mathcal{P}\,[\![0]\!]$	$\mathsf{BExp} \to \mathcal{P}(\mathsf{Scope})$
`\bsem{0}`	$\mathcal{B}\,[\![0]\!]$	$\mathsf{BExp} \to (\mathsf{Scope} \to 2)$
`\csem{0}`	$\mathcal{C}\,[\![0]\!]$	$\mathsf{Cmd} \to (\mathsf{Scope} \to \mathcal{P}(\mathsf{Scope}) \cup \{\lightning\})$
`\bigto`	$\rightsquigarrow$	Big step
`\nbigto`	$\not\rightsquigarrow$	
`\fault`	$\lightning$	Fault
`\writeVar`	writeVar	
`\writeVars`	writeVars	
`\lookup`	lookup	
`\axioms{0}`	$\mathrm{Ax}\,[\![0]\!]$	
Abstract modules		
`\am{0}`	$\nvdash$	
`\amH`	$\mathbb{H}$	Heap module
`\amT`	$\mathbb{T}$	Tree module
`\amL`	$\mathbb{L}$	List module
Module translations		
`\trl`	τ	Translation

Table 8: Programs

Macro	Output	Description
`\ar`	α	Abstraction relation
`\imp{0}`	$[\![0]\!]$	Implementation
`\prtr{0}`	$[\![0]\!]$	Predicate translation
`\TIF`	$\mathcal{I}$	Interface set
`\TIFi`	$\mathcal{I}_{\text{in}}$	
`\TIFo`	$\mathcal{I}_{\text{out}}$	
`\intf`	I	Interface
`\intfi`	in	
`\intfo`	out	
`\TCP`	$\mathcal{F}$	Crust parameter set
`\crup`	F	Crust parameter
`\drep{0}{1}`	$\langle\!\langle 0 \rangle\!\rangle^{1}$	Data rep
`\crep{0}{1}{2}`	$\langle\!\langle 0 \rangle\!\rangle^{1}_{2}$	Context rep
`\crust{0}{1}`	$\Cap^{1}_{0}$	Crust
`\rsem{0}`	$(\!\lvert 0 \rvert\!)$	
`\idpt{0}{1}`	$(\!\lvert 0 \rvert\!)^{1}$	Intermediate Data Translation
`\icpt{0}{1}{2}`	$(\!\lvert 0 \rvert\!)^{1}_{2}$	Intermediate Context Translation
Predicates		
`\vsafe`	$vsafe$	
`\bsafe`	$bsafe$	
Hoare logic		
`\triple{0}{1}{2}`	$\{0\}\ 1\ \{2\}$	Hoare triple
`\vtriple{0}{1}{2}`	$\begin{array}{l}\{0\}\\ 1\\ \{2\}\end{array}$	
`\pspec{0}{1}{2}`	$0 : 1 \rightarrowtail 2$	
Hoare logic rules		
`\mkrule{0}`	0	
`\raxiom`	Axiom	Axiom
`\rframe`	Frame	Frame
`\rcons`	Cons	Consequence
`\rdisj`	Disj	Disjunction
`\rskip`	Skip	Skip
`\rseq`	Seq	Sequencing
`\rif`	If	If
`\rwhile`	While	While
`\rassgn`	Assgn	Assignment
`\rlocal`	Local	Local variable
`\rpdef`	PDef	Procedure definition
`\rpcall`	PCall	Procedure call
`\rpwk`	PWk	Procedure weakening

Table 8: Programs

Macro	Output	Description
`\rconj`	$\textsc{Conj}$	Conjunction

Table 9: Cap

Macro	Output	Description
`\wf`	wf	Well-formedness
`\stable{0}`	$stable\,(0)$	Stability
`\locp{0}`	0_{L}	Local portion
`\shap{0}`	0_{S}	Shared portion
`\hpp{0}`	0_{H}	Heap portion
`\pmp{0}`	0_{P}	Permission portion
`\stap{0}`	0_{R}	State portion
`\actp{0}`	0_{A}	Action portion
`\amplus`	$\sqcup$	Action model combination
`\psep`	$\oplus$	Permission combination
`\zperm`	$\mathbf{0}_{\mathsf{Perm}}$	Zero permission
`\lssep`	$\odot$	Logical state combination
`\wsep`	$\bullet$	World combination
`\zworld`	$\mathbf{0}_{\mathsf{World}}$	Zero world
`\act{0}`	0	Action name
`\abp{0}`	$\mathsf{0}$	Abstract predicate
`\lcol{0}`	$\lfloor 0 \rfloor$	LState collapse
`\hcol{0}`	$\lfloor 0 \rfloor_{\mathrm{H}}$	Heap collapse
`\Guar`	G	Guarantee relation
`\Guarc`	G_c	Construction operation
`\Guarr`	$\overline{\mathrm{G}}$	Repartitioning guarantee
`\Guars`	$\widehat{\mathrm{G}}$	Step-close guarantee
`\Rely`	R	Rely relation
`\Relyc`	R_c	Construction
`\PRED`	P	Predicates
`\PREDq`	Q	
`\PREDr`	R	
`\pred`	p	
`\predq`	q	
`\spredq`	$\mathcal{Q}$	
`\pde`	Δ	Abstract predicate axioims
`\len`	len	Vector length

Table 9: Cap

Macro	Output	Description
`\fpex{0}{1}{2}{3}`	$0 \Uparrow 1, 2, 3$	Fixed-point Existence
Programming language		
`\atomic{0}`	$\langle 0 \rangle$	
`\ppar`	$\parallel$	
`\letin{0}{1}`	`let 0 in 1`	
`\pstmts`	c	
`\safe{0}{1}`	$safe_0\,(1)$	
Proof rules		
`\ratomic`	Atomic	Atomic
`\rprim`	Prim	Primitive
`\rpar`	Par	Parallel
`\rguarl`	Guar-L	Guarantee left
`\rguarr`	Guar-R	Guarantee right
`\rpredi`	Pred-I	Predicate introduction
`\rprede`	Pred-E	Predicate elimination
`\rlet`	Let	Let
`\rcall`	Call	Call
`\rloop`	Loop	Loop
`\rchoice`	Choice	Choice
`\rexist`	Exist	Existential