Functions

$f(x)$	`f(x)`	Application
$(\lambda\, x{:}T \mid P \bullet E)$	`(\lambda ...)`	Lambda-expression
$X ⇸ Y$	`X \pfun Y`	Partial functions
$X \longrightarrow Y$	`X \fun Y`	Total functions
$X ⤔ Y$	`X \pinj Y`	Partial injections
$X \rightarrowtail Y$	`X \inj Y`	Total injections
$X ⤀ Y$	`X \psurj Y`	Partial surjections
$X \twoheadrightarrow Y$	`X \surj Y`	Total surjections
$X ⤖ Y$	`X \bij Y`	Bijections
$X ⇻ Y$	`X \ffun Y`	Finite functions
$X ⤕ Y$	`X \finj Y`	Finite injections

Numbers and arithmetic

$\mathbb{N}$	`\nat`	Natural numbers
$\mathbb{Z}$	`\num`	Integers
$+ - *$ div mod	`+ - * \div \mod`	Operations
$< \leq \geq >$	`< \leq \geq >`	Comparisons
$\mathbb{N}_1$	`\nat_1`	Integers > 0
succ	`succ`	Successor function
$m \mathrel{..} n$	`m \upto n`	Number range
$\#S$	`\# S`	Size of a set
min S	`min~S`	Minimum of a set
max S	`max~S`	Maximum of a set

Sequences

seq X	`\seq X`	Finite sequences
$\mathrm{seq}_1\ X$	`\seq_1 X`	Sequences $\neq \langle\rangle$
iseq X	`\iseq X`	Injective sequences
$\langle x_1, \ldots, x_n \rangle$	`\langle \rangle`	Sequence display
$s \frown t$	`s \cat t`	Concatenation
rev s	`rev~s`	Reverse
head s	`head~s`	Head of sequence
last s	`last~s`	Last element
tail s	`tail~s`	Tail of sequence
front s	`front~s`	All but last element
$U \upharpoonleft s$	`U \extract S`	Extraction
$s \upharpoonright V$	`s \filter V`	Filtering
squash f	`squash~f`	Compaction
s prefix t	`s \prefix t`	Prefix relation
s suffix t	`s \suffix t`	Suffix relation
s in t	`s \inseq t`	Segment relation
$\frown/ss$	`\dcat ss`	Distributed concat.
disjoint SS	`\disjoint SS`	Disjointness
SS partition T	`SS \partition T`	Partition relation

Bags

bag X	`\bag X`	Bags
$[\![x_1, \ldots, x_n]\!]$	`\lbag ... \rbag`	Bag display
count B x	`count~B~x`	Count of element
$B \sharp x$	`B \bcount x`	Infix count operator
$n \otimes B$	`n \otimes B`	Bag scaling
$x ⋿ B$	`x \inbag B`	Bag membership
$B \sqsubseteq C$	`B \subbageq C`	Sub-bag relation
$B \uplus C$	`B \uplus C`	Bag union
$B ⩁ C$	`B \uminus C`	Bag difference
items s	`items~s`	Items in a sequence

*f*uzz flags

Usage: `fuzz` *[*`-aqstv`*] [*`-p` *prelude] [file ...]*

`-a`	Don't use type abbreviations
`-p` *prelude*	Use *prelude* in place of the standard one
`-q`	Implicit quantifiers
`-d`	Dependency analysis
`-s`	Syntax check only
`-t`	Report types of global definitions
`-v`	Echo formal text as it is parsed

Z Reference Card

Mike Spivey
The Spivey Partnership

Specifications

Schema box

Name[Params]
Declarations
Predicates

```
\begin{schema}{Name}[Params]
  Declarations
\where
  Predicates
\end{schema}
```

Axiomatic description

Declarations
Predicates

```
\begin{axdef}
  Declarations
\where
  Predicates
\end{axdef}
```

Generic definition

[Params]
Declarations
Predicates

```
\begin{gendef}[Params]
  Declarations
\where
  Predicates
\end{gendef}
```

`\begin{zed} ...`

Basic type definition

$[NAME, DATE]$	`[NAME, DATE]`

Abbreviation definition

$DOC == \mathrm{seq}\, CHAR$	`DOC == \seq CHAR`

Constraint

$n\_disks < 5$	`n\_disks < 5`

Schema definition

$Point \mathrel{\widehat{=}} [\, x, y : \mathbb{Z} \,]$	`Point \defs [~x, y: \num~]`

Free type definition

$Ans ::= ok\langle\!\langle \mathbb{Z} \rangle\!\rangle \mid error$	`Ans ::= ok \ldata\num\rdata`
	`| error`

`... \end{zed}`

Logic and schema calculus

$true, false$	`true, false`	Logical constants
$\neg P$	`\lnot P`	Negation
$P \land Q$	`P \land Q`	Conjunction
$P \lor Q$	`P \lor Q`	Disjunction
$P \Rightarrow Q$	`P \implies Q`	Implication
$P \Leftrightarrow Q$	`P \iff Q`	Equivalence
$\forall x : T \mid P \bullet Q$	`\forall ...`	Universal quantifier
$\exists x : T \mid P \bullet Q$	`\exists ...`	Existential quant.
$\exists_1 x : T \mid P \bullet Q$	`\exists_1 ...`	Unique quantifier

Special schema operators

$S[y_1/x_1, y_2/x_2]$	`S[y1/x1, y2/x2]`	Renaming
$S \setminus (x_1, x_2)$	`S \hide (x1, x2)`	Hiding
$S \upharpoonright T$	`S \project T`	Projection
$\mathrm{pre}\; Op$	`\pre Op`	Pre-condition
$Op1 \fatsemi Op2$	`Op1 \semi Op2`	Sequential comp.
$Op1 \gg Op2$	`Op1 \pipe Op2`	Piping

Basic expressions

$x = y$	`x = y`	Equality
$x \neq y$	`x \neq y`	Inequality

$\mathbf{if}\ P\ \mathbf{then}\ E_1$
$\mathbf{else}\ E_2$
	`\IF P \THEN E_1`
	`\ELSE E_2`
		Conditional
		expression

θS	`\theta S`	Theta-expression
$E.x$	`E.x`	Selection
$(\mu\, x{:}T \mid P \bullet E)$	`(\mu x:T | P @ E)`	Mu-expression
$(\mathbf{let}\ x{==}E_1 \bullet E_2)$	`(\LET x==E1 @ E2)`	Let-expression

Sets

$x \in S$	`x \in S`	Membership
$x \notin S$	`x \notin S`	Non-membership

$\{x_1, \ldots, x_n\}$	`\{x_1,...,x_n\}`	Set display
$\{\, x{:}T \mid P \bullet E \,\}$	`\{~x:T | P @ E~\}`	Set comprehension
$\varnothing$	`\emptyset`	Empty set
$S \subseteq T$	`S \subseteq T`	Subset relation
$S \subset T$	`S \subset T`	Proper subset
$\mathbb{P}\, S$	`\power S`	Power set
$\mathbb{P}_1\, S$	`\power_1 S`	Non-empty subsets
$S \times T$	`S \cross T`	Cartesian product
(x, y, z)	`(x, y, z)`	Tuple
$first\ p$	`first~p`	First of pair
$second\ p$	`second~p`	Second of pair
$S \cup T$	`S \cup T`	Set union
$S \cap T$	`S \cap T`	Set intersection
$S \setminus T$	`S \setminus T`	Set difference
$\bigcup A$	`\bigcup A`	Generalized union
$\bigcap A$	`\bigcap A`	Gen. intersection
$\mathbb{F}\, X$	`\finset X`	Finite sets
$\mathbb{F}_1\, X$	`\finset_1 X`	Finite sets $\neq \varnothing$

Relations

$X \leftrightarrow Y$	`X \rel Y`	Binary relations
$x \mapsto y$	`x \mapsto y`	Maplet
$\mathrm{dom}\, R$	`\dom R`	Domain
$\mathrm{ran}\, R$	`\ran R`	Range
$\mathrm{id}\, X$	`\id X`	Identity relation
$Q \fatsemi R$	`Q \comp R`	Composition
$Q \circ R$	`Q \circ R`	Backwards comp.
$S \lhd R$	`S \dres R`	Domain restriction
$R \rhd S$	`R \rres S`	Range restriction
$S ⩤ R$	`S \ndres R`	Domain anti-res.
$R ⩥ S$	`R \nrres S`	Range anti-restrict.
$R^{\sim}$	`R \inv`	Relational inverse
$R(\!\vert S \vert\!)$	`R \limg S\rimg`	Relational image
$Q \oplus R$	`Q \oplus R`	Overriding
R^k	`R^{k}`	Iteration
R^+	`R \plus`	Transitive closure
R^*	`R \star`	Refl.–trans. closure