Z Reference Card

Mike Spivey

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 quantifier
$\exists_1 x : T \mid P \bullet Q$	`\exists_1 ...`	Unique quantifier

Special schema operators

$S[y_1/x_1, y_2/x_2]$	`S[y_1/x_1, y_2/x_2]`	Renaming
$S \setminus (x_1, x_2)$	`S \hide (x_1, x_2)`	Hiding
$S1 \upharpoonright S2$	`S1 \project S2`	Projection
$\mathrm{pre}\ Op$	`\pre Op`	Pre-condition
$Op1 \fatsemi Op2$	`Op1 \semi Op2`	Sequential composition
$Op1 \gg Op2$	`Op1 \pipe Op2`	Piping

Basic expressions

<table>
<tr><td>$x = y$</td><td><code>x = y</code></td><td>Equality</td></tr>
<tr><td>$x \neq y$</td><td><code>x \neq y</code></td><td>Inequality</td></tr>
<tr><td>$\mathbf{if}\ P\ \mathbf{then}\ E_1$</td><td><code>\IF P \THEN E_1</code></td><td>Conditional</td></tr>
<tr><td>$\quad\mathbf{else}\ E_2$</td><td><code> \ELSE E_2</code></td><td>Expression</td></tr>
<tr><td>θS</td><td><code>\theta S</code></td><td>Theta-expression</td></tr>
<tr><td>$E.x$</td><td><code>E.x</code></td><td>Selection</td></tr>
<tr><td>$(\mu\, x : T \mid P \bullet E)$</td><td><code>(\mu x: T | P @ E)</code></td><td>Mu-expression</td></tr>
<tr><td>$(\mathbf{let}\ x{=}{=}E1 \bullet E2)$</td><td><code>(\LET x == E1 @ E2)</code></td><td>Let-expression</td></tr>
</table>

Sets

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

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 composition
$S \lhd R$	`S \dres R`	Domain restriction
$R \rhd S$	`R \rres S`	Range restriction
$S \ndres R$	`S \ndres R`	Domain anti-restriction
$R \nrres S$	`R \nrres S`	Range anti-restriction
$R^{\sim}$	`R \inv`	Relational inverse
$R \limg S \rimg$	`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`	Reflexive–trans. closure

Functions

$f(x)$	`f(x)`	Function application
$(\lambda x : T \mid P \bullet E)$	`(\lambda ...)`	Lambda-expression
$X \pfun Y$	`X \pfun Y`	Partial functions
$X \longrightarrow Y$	`X \fun Y`	Total functions
$X \pinj Y$	`X \pinj Y`	Partial injections
$X \rightarrowtail Y$	`X \inj Y`	Total injections
$X \psurj Y$	`X \psurj Y`	Partial surjections
$X \twoheadrightarrow Y$	`X \surj Y`	Total surjections
$X \bij Y$	`X \bij Y`	Bijections
$X \ffun Y$	`X \ffun Y`	Finite partial functions
$X \finj Y$	`X \finj Y`	Finite partial injections

Numbers and arithmetic

$\mathbb{N}$	`\nat`	Natural numbers
$\mathbb{Z}$	`\num`	Integers
$+ - * \ \mathsf{div}\ \mathsf{mod}$	`+ - * \div \mod`	Arithmetic operations
$< \leq \geq >$	`< \leq \geq >`	Arithmetic comparisons
$\mathbb{N}_1$	`\nat_1`	Strictly positive integers
$succ$	`succ`	Successor function
$m \mathinner{\ldotp\ldotp} n$	`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

$\mathrm{seq}\, X$	`\seq X`	Finite sequences
$\mathrm{seq}_1\, X$	`\seq_1 X`	Non-empty sequences
$\mathrm{iseq}\, X$	`\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 of sequence
$tail\ s$	`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
$s\ \mathsf{prefix}\ t$	`s \prefix t`	Prefix relation
$s\ \mathsf{suffix}\ t$	`s \suffix t`	Suffix relation
$s\ \mathsf{in}\ t$	`s \inseq t`	Segment relation
$\frown/ss$	`\dcat ss`	Distributed concat.
$\mathsf{disjoint}\ SS$	`\disjoint SS`	Disjointness
$SS\ \mathsf{partition}\ T$	`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 an element
$B \sharp x$	`B \bcount x`	Infix count operator
$n \otimes B$	`n \otimes B`	Bag scaling
$x \sqsubset\!\!\in B$	`x \inbag B`	Bag membership
$B \sqsubseteq C$	`B \subbageq C`	Sub-bag relation
$B \uplus C$	`B \uplus C`	Bag union
$B \cup\!\!\!- C$	`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` *predude*	Use *prelude* in place of the standard one
`-q`	Assume implicit quantifiers for undeclared variables
`-d`	Dependency analysis
`-s`	Syntax check only
`-t`	Report types of global definitions
`-v`	Echo formal text as it is parsed