Isabelle/UTP Syntax Reference

Simon Foster Frank Zeyda

June 20, 2017

1 Syntax Overview

The Isabelle jEdit interface support a form of autocompletion. When typing mathematical syntax it will often present a list of suggested symbols. When this appears and the right symbol is in scope press the TAB key to insert it. This process will be necessary during most editing in Isabelle. On the whole, Isabelle provides LaTeX-like symbols for the majority of its operators.

2 Expression Operators

2.1 Arithmetic Operators

The arithmetic operators employ the Isabelle/HOL type classe hierarchy for groups, rings, fields, and orders. Consequently, UTP enjoys direct syntax for many of these operators.

Math Operator	**Description**	**Isabelle/UTP**	**Isabelle code(s)**
$0, 1, 17, 3.147$	numerals	$0, 1, 17, 3.147$	`0, 1, 17, 3.147`
$x + y$	addition	$x + y$	`x + y`
$x - y$	subtraction	$x - y$	`x - y`
$x \cdot y$	multiplication	$x * y$	`x * y`
x/y	division	x/y	`x / y`
$x \,\mathbf{div}\, y$	integer division	$x \,\mathbf{div}\, y$	`x div y`
$x \,\mathbf{mod}\, y$	integer modulo	$x \,\mathbf{mod}\, y$	`x mod y`
$\lceil x \rceil$	numeric ceiling	$\lceil x \rceil_u$	`\lceil x \rceil \sub u`
$\lfloor x \rfloor$	numeric floor	$\lfloor x \rfloor_u$	`\lfloor x \rfloor \sub u`
$x \leq y$	less-than-or-equal	$x \leq_u y$	`x \le \sub u y`
$x < y$	less-than	$x <_u y$	`x < \sub u y`
$\min(x, y)$	minimum value	$\min_u(x, y)$	`min \sub u (x, y)`
$\max(x, y)$	maximum value	$\max_u(x, y)$	`max \sub u (x, y)`

2.2 Polymorphic Operators

The following operators are overloaded to various different expression types. Notably the functional operators, such as application, update, and domain, can be applied to a variety of HOL types including, functions, finite maps, and relations.

Math Operator	Description	Isabelle/UTP	Isabelle code(s)
$P = Q$	equals	$P =_u Q$	`P =\sub u Q`
$P \neq Q$	not equals	$P \neq_u Q$	`P \noteq\sub u Q`
$\lambda\, x \bullet P(x)$	λ-abstraction	$\lambda\, x \bullet P(x)$	`\lambda x \bullet P(x)`
$(x, y, \cdots, z)$	tuple	$(x, y, \cdots, z)_u$	`(x, y, ..., z)\sub u`
$\pi_1(x)$	tuple project first	$\pi_1(x)$	`\pi \sub 1 (x)`
$\pi_2(x)$	tuple project second	$\pi_2(x)$	`\pi \sub 2 (x)`
$f(x)$	functional application	$f(x)_a$	`f(x) \sub a`
$f \oplus \{k_1 \mapsto v_1, \cdots\}$	functional update	$f(k_1 \mapsto v_1, \cdots)_u$	`f(k1 \mapsto v1, ...)\sub u`
$\{k_1 \mapsto v_1, \cdots\}$	enumerated map	$[k_1 \mapsto v_1, \cdots]_u$	`[k1 \mapsto v1, ...]\sub u`
$\emptyset$	empty collection	$[]_u$	`[] \sub u`
$\#x$	size of collection	$\#_u(x)$	`# \sub u (x)`
$\mathrm{dom}(x)$	domain	$\mathrm{dom}_u(x)$	`dom \sub u (x)`
$\mathrm{ran}(x)$	range	$\mathrm{dom}_u(x)$	`ran \sub u (x)`
$f \lhd A$	domain restriction	$f \lhd_u A$	`f \lhd \sub u A`
$A \rhd f$	range restriction	$A \rhd_u f$	`A \rhd \sub u f`

2.3 Sequence Operators

2.4 Set Operators

3 Predicate Operators

Math Operator	Description	Isabelle code(s)
true	logical true / universal relation	`true`
false	logical false / empty relation	`false`
$\neg P$	negation / complement	`~` or `\not`
$P \wedge Q$	conjunction	`/\` or `\and`
$P \vee Q$	disjunction	`\/` or `\or`
$P \Rightarrow Q$	implication	`=>` or `\Rightarrow`
$P \sqsubseteq Q$	refinement	`[=` or `\sqsubseteq`
x	predicate variable	`&x`
x	relational before variable	`$x`
x'	relational after variable	`$x\acute`
$\ll v \gg$	HOL term / variable quotation	`<<x>>`
$\forall x \bullet P$	universal quantifier (UTP variable)	`! x \bullet P`
$\exists x \bullet P$	existential quantifier (UTP variable)	`? x \bullet P`
$\boldsymbol{\forall} x \bullet P$	universal quantifier (HOL variable)	`\bold ! x \bullet P`
$\boldsymbol{\exists} x \bullet P$	existential quantifier (HOL variable)	`\bold ? x \bullet P`
$P \sqcap Q$	binary infimum / internal choice	`P \sqcap Q`
$P \sqcup Q$	binary supremum	`P \sqcup Q`
$\bigsqcap i \in I \bullet P(i)$	indexed infimum / internal choice	`\Sqcap i \in I \bullet P(i)`
$\bigsqcup i \in I \bullet P(i)$	indexed supremum	`\Sqcup i \in I \bullet P(i)`
$\mu X \bullet P(X)$	weakest fixed-point	`\mu X \bullet P(X)`
$\nu X \bullet P(X)$	strongest fixed-point	`\nu X \bullet P(X)`
P	UTP predicate tautology	`` `P` ``

4 Meta-logic Operators