Summary of TLA$^+$

Module-Level Constructs

MODULE M

Begins the module or submodule named M.

EXTENDS $M_1, \ldots, M_n$

Incorporates the declarations, definitions, assumptions, and theorems from the modules named $M_1, \ldots, M_n$ into the current module.

CONSTANTS $C_1, \ldots, C_n$ (1)

Declares the C_j to be constant parameters (rigid variables). Each C_j is either an identifier or has the form $C(\_, \ldots, \_)$, the latter form indicating that C is an operator with the indicated number of arguments.

VARIABLES $x_1, \ldots, x_n$ (1)

Declares the x_j to be variables (parameters that are flexible variables).

ASSUME P

Asserts P as an assumption.

$F(x_1, \ldots, x_n) \triangleq exp$

Defines F to be the operator such that $F(e_1, \ldots, e_n)$ equals exp with each identifier x_k replaced by e_k. (For $n = 0$, it is written $F \triangleq exp$.)

$f[x \in S] \triangleq exp$ (2)

Defines f to be the function with domain S such that $f[x] = exp$ for all x in S. (The symbol f may occur in exp, allowing a recursive definition.)

---

(1) The terminal S in the keyword is optional.

(2) $x \in S$ may be replaced by a comma-separated list of items $v \in S$, where v is either a comma-separated list or a tuple of identifiers.

INSTANCE M WITH $p_1 \leftarrow e_1, \ldots, p_m \leftarrow e_m$

For each defined operator F of module M, this defines F to be the operator whose definition is obtained from the definition of F in M by replacing each declared constant or variable p_j of M with e_j. (If $m = 0$, the WITH is omitted.)

$N(x_1, \ldots, x_n) \triangleq$ INSTANCE M WITH $p_1 \leftarrow e_1, \ldots, p_m \leftarrow e_m$

For each defined operator F of module M, this defines $N(d_1, \ldots, d_n)!F$ to be the operator whose definition is obtained from the definition of F by replacing each declared constant or variable p_j of M with e_j, and then replacing each identifier x_k with d_k. (If $m = 0$, the WITH is omitted.)

THEOREM P

Asserts that P can be proved from the definitions and assumptions of the current module.

LOCAL *def*

Makes the definition(s) of *def* (which may be a definition or an INSTANCE statement) local to the current module, thereby not obtained when extending or instantiating the module.

Ends the current module or submodule.

The Constant Operators

Logic

$\land \quad \lor \quad \neg \quad \Rightarrow \quad \equiv$

TRUE FALSE BOOLEAN [the set {TRUE, FALSE}]

$\forall\, x \in S : p$ (1) $\quad \exists\, x \in S : p$ (1)

CHOOSE $x \in S : p$ [An x in S satisfying p]

Sets

$= \quad \neq \quad \in \quad \notin \quad \cup \quad \cap \quad \subseteq \quad \setminus$ [set difference]

$\{e_1, \ldots, e_n\}$	[Set consisting of elements e_i]
$\{x \in S : p\}$ (2)	[Set of elements x in S satisfying p]
$\{e : x \in S\}$ (1)	[Set of elements e such that x in S]
SUBSET S	[Set of subsets of S]
UNION S	[Union of all elements of S]

Functions

$f[e]$	[Function application]
DOMAIN f	[Domain of function f]
$[x \in S \mapsto e]$ (1)	[Function f such that $f[x] = e$ for $x \in S$]
$[S \to T]$	[Set of functions f with $f[x] \in T$ for $x \in S$]
$[f$ EXCEPT $![e_1] = e_2]$ (3)	[Function $\widehat{f}$ equal to f except $\widehat{f}[e_1] = e_2$]

Records

$e.h$	[The h-field of record e]
$[h_1 \mapsto e_1, \ldots, h_n \mapsto e_n]$	[The record whose h_i field is e_i]
$[h_1 : S_1, \ldots, h_n : S_n]$	[Set of all records with h_i field in S_i]
$[r$ EXCEPT $!.h = e]$ (3)	[Record $\widehat{r}$ equal to r except $\widehat{r}.h = e$]

Tuples

$e[i]$	[The i^{th} component of tuple e]
$\langle e_1, \ldots, e_n \rangle$	[The n-tuple whose i^{th} component is e_i]
$S_1 \times \ldots \times S_n$	[The set of all n-tuples with i^{th} component in S_i]

---

(1) $x \in S$ may be replaced by a comma-separated list of items $v \in S$, where v is either a comma-separated list or a tuple of identifiers.

(2) x may be an identifier or tuple of identifiers.

(3) $![e_1]$ or $!.h$ may be replaced by a comma separated list of items $!\,a_1 \cdots a_n$, where each a_i is $[e_i]$ or $.h_i$.

Miscellaneous Constructs

IF p THEN e_1 ELSE e_2	[e_1 if p true, else e_2]
CASE $p_1 \to e_1 \,\Box\, \ldots \,\Box\, p_n \to e_n$	[Some e_i such that p_i true]
CASE $p_1 \to e_1 \,\Box\, \ldots \,\Box\, p_n \to e_n \,\Box$ OTHER $\to e$	[Some e_i such that p_i true, or e if all p_i are false]

LET $d_1 \triangleq e_1 \;\ldots\; d_n \triangleq e_n$ IN e [e in the context of the definitions]

$\bigwedge\; p_1$ [the conjunction $p_1 \wedge \ldots \wedge p_n$]
$\vdots$
$\bigwedge\; p_n$

$\bigvee\; p_1$ [the disjunction $p_1 \vee \ldots \vee p_n$]
$\vdots$
$\bigvee\; p_n$

Action Operators

e'	[The value of e in the final state of a step]
$[A]_e$	$[A \vee (e' = e)]$
$\langle A \rangle_e$	$[A \wedge (e' \neq e)]$
ENABLED A	[An A step is possible]
UNCHANGED e	$[e' = e]$
$A \cdot B$	[Composition of actions]

Temporal Operators

$\Box F$	[F is always true]
$\Diamond F$	[F is eventually true]
$\mathrm{WF}_e(A)$	[Weak fairness for action A]
$\mathrm{SF}_e(A)$	[Strong fairness for action A]
$F \rightsquigarrow G$	[F leads to G]

User-Definable Operator Symbols

Infix Operators

$+$ (1)	$-$ (1)	$*$ (1)	$/$ (2)	$\circ$ (3)	$++$
$\div$ (1)	$\%$ (1)	$\hat{}$ (1,4)	$..$ (1)	$\ldots$	$--$
$\oplus$ (5)	$\ominus$ (5)	$\otimes$	$\oslash$	$\odot$	$**$
$<$ (1)	$>$ (1)	$\leq$ (1)	$\geq$ (1)	$\sqcap$	$//$
$\prec$	$\succ$	$\preceq$	$\succeq$	$\sqcup$	$\hat{}\hat{}$
$\ll$	$\gg$	$<:$	$:>$ (6)	$\&$	$\&\&$
$\sqsubset$	$\sqsupset$	$\sqsubseteq$ (5)	$\sqsupseteq$	$\mid$	$\%\%$
$\subset$	$\supset$		$\supseteq$	$\star$	$@@$ (6)
$\vdash$	$\dashv$	$\models$	$=\!\mid$	$\bullet$	$\#\#$
$\sim$	$\simeq$	$\approx$	$\cong$	$\$$	$\$\$$
$\bigcirc$	$::=$	$\asymp$	$\doteq$	$??$	$!!$
$\propto$	$\wr$	$\uplus$			

Postfix Operators (7)

$\hat{}+$ $\hat{}*$ $\hat{}\#$

---

(1) Defined by the *Naturals*, *Integers*, and *Reals* modules.
(2) Defined by the *Reals* module.
(3) Defined by the *Sequences* module.
(4) $x\hat{}y$ is printed as x^y.
(5) Defined by the *Bags* module.
(6) Defined by the *TLC* module.
(7) $e\hat{}+$ is printed as e^+, and similarly for $\hat{}*$ and $\hat{}\#$.

Precedence Ranges of Operators

The relative precedence of two operators is unspecified if their ranges overlap.
Left-associative operators are indicated by (a).

Prefix Operators

$\neg$	4–4	$\Box$	4–15	UNION	8–8
ENABLED	4–15	$\Diamond$	4–15	DOMAIN	9–9
UNCHANGED	4–15	SUBSET	8–8	$-$	12–12

Infix Operators

$\Rightarrow$	1–1	$\leq$	5–5	$<:$	7–7	$\ominus$	11–11 (a)
$\stackrel{+}{\triangleright}$	2–2	$\ll$	5–5	$\setminus$	8–8	$-$	11–11 (a)
$\equiv$	2–2	$\prec$	5–5	$\cap$	8–8 (a)	$--$	11–11 (a)
$\rightsquigarrow$	2–2	$\preceq$	5–5	$\cup$	8–8 (a)	$\&$	13–13 (a)
$\wedge$	3–3 (a)	$\propto$	5–5	$..$	9–9	$\&\&$	13–13 (a)
$\vee$	3–3 (a)	$\sim$	5–5	$\ldots$	9–9	$\odot$	13–13 (a)
$\neq$	5–5	$\simeq$	5–5	$!!$	9–13	$\oslash$	13–13
$\dashv$	5–5	$\sqsubset$	5–5	$\#\#$	9–13 (a)	$\otimes$	13–13 (a)
$::=$	5–5	$\sqsubseteq$	5–5	$\$$	9–13 (a)	$*$	13–13 (a)
$:=$	5–5	$\sqcup$	5–5	$\$\$$	9–13 (a)	$**$	13–13 (a)
$<$	5–5	$\sqcup$	5–5	$??$	9–13 (a)	$/$	13–13
$=$	5–5	$\subset$	5–5	$\sqcap$	9–13 (a)	$//$	13–13
$\models$	5–5	$\subseteq$	5–5	$\sqcup$	9–13 (a)	$\bigcirc$	13–13 (a)
$>$	5–5	$\succ$	5–5	$\uplus$	9–13 (a)	$\bullet$	13–13 (a)
$\approx$	5–5	$\succeq$	5–5	$\wr$	9–14	$\div$	13–13
$\asymp$	5–5	$\cup$	5–5	$\oplus$	10–10 (a)	$\circ$	13–13 (a)
$\cong$	5–5	$\cup$	5–5	$+$	10–10 (a)	$\star$	13–13 (a)
$\doteq$	5–5	$\top$	5–5	$++$	10–10 (a)	$\hat{}$	14–14
$\geq$	5–5	$\models$	5–5	$\%$	10–11	$\hat{}\hat{}$	14–14
$\gg$	5–5	$\cdot^{(1)}$	5–14 (a)	$\%\%$	10–11 (a)	$.^{(2)}$	17–17 (a)
$\in$	5–5	@@	6–6 (a)	$\mid$	10–11 (a)		
$\notin$	5–5	$:>$	7–7	$\parallel$	10–11 (a)		

Postfix Operators

`^+`	15–15	`^*`	15–15	`^#`	15–15	$'$	15–15

(1) Action composition (`\cdot`).
(2) Record field (period).

Operators Defined in Standard Modules.

Modules *Naturals*, *Integers*, *Reals*

$+$	$-$ (1)	$*$	$/$ (2)	$\hat{}$ (3)	$..$	*Nat*	*Real* (2)
$\div$	$\%$	$\leq$	$\geq$	$<$	$>$	*Int* (4)	*Infinity* (2)

(1) Only infix $-$ is defined in *Naturals*.
(2) Defined only in *Reals* module.
(3) Exponentiation.
(4) Not defined in *Naturals* module.

Module *Sequences*

$\circ$	*Head*	*SelectSeq*	*SubSeq*
Append	*Len*	*Seq*	*Tail*

Module *FiniteSets*

IsFiniteSet *Cardinality*

Module *Bags*

$\oplus$	*BagIn*	*CopiesIn*	*SubBag*
$\ominus$	*BagOfAll*	*EmptyBag*	
$\sqsubseteq$	*BagToSet*	*IsABag*	
BagCardinality	*BagUnion*	*SetToBag*	

Module *RealTime*

RTBound *RTnow* *now* (declared to be a variable)

Module *TLC*

$:>$	@@	*Print*	*Assert*	*JavaTime*	*Permutations*
SortSeq					

ASCII Representation of Typeset Symbols

$\wedge$	/\ or \land	$\vee$	\/ or \lor	$\Rightarrow$	=>
$\neg$	~ or \lnot or \neg	$\equiv$	<=> or \equiv	$\triangleq$	==
$\in$	\in	$\notin$	\notin	$\neq$	# or /=
$\langle$	<<	$\rangle$	>>	$\Box$	[]
$<$	<	$>$	>	$\Diamond$	<>
$\leq$	\leq or =< or <=	$\geq$	\geq or >=	$\rightsquigarrow$	~>
$\ll$	\ll	$\gg$	\gg	$\stackrel{+}{\rightarrow}$	-+->
$\prec$	\prec	$\succ$	\succ	$\mapsto$	|->
$\preceq$	\preceq	$\succeq$	\succeq	$\div$	\div
$\subseteq$	\subseteq	$\supseteq$	\supseteq	$\cdot$	\cdot
$\subset$	\subset	$\supset$	\supset	$\circ$	\o or \circ
$\sqsubset$	\sqsubset	$\sqsupset$	\sqsupset	$\bullet$	\bullet
$\sqsubseteq$	\sqsubseteq	$\sqsupseteq$	\sqsupseteq	$\star$	\star
$\vdash$	|-	$\dashv$	-|	$\bigcirc$	\bigcirc
$\models$	|=	$=\!\!\mid$	=|	$\sim$	\sim
$\rightarrow$	->	$\leftarrow$	<-	$\simeq$	\simeq
$\cap$	\cap or \intersect	$\cup$	\cup or \union	$\asymp$	\asymp
$\sqcap$	\sqcap	$\sqcup$	\sqcup	$\approx$	\approx
$\oplus$	(+) or \oplus	$\uplus$	\uplus	$\cong$	\cong
$\ominus$	(-) or \ominus	$\times$	\X or \times	$\doteq$	\doteq
$\odot$	(.) or \odot	$\wr$	\wr	x^y	x^y (2)
$\otimes$	(\X) or \otimes	$\propto$	\propto	x^+	x^+ (2)
$\oslash$	(/) or \oslash	“s”	"s" (1)	x^*	x^* (2)
$\exists$	\E	$\forall$	\A	$x^\#$	x^# (2)
$\mathbf{\exists}$	\EE	$\mathbf{\forall}$	\AA	$'$	'
$]_v$	]_v	$\rangle_v$	>>_v		
WF_v	WF_v	SF_v	SF_v		

┌────────	-------- (3)	────────	-------- (3)
├────────	-------- (3)	└────────	======== (3)

(1) s is a sequence of characters.
(2) x and y are any expressions.
(3) a sequence of four or more - or = characters.