

Index

- ' (prime), 1
- " (double quote), 1
- "..." (string), 1
- ^ (exponentiation), 1
- || (PlusCal multi-assignment), 1
- ⊢, 1
- ||−, 1
- ||− rule, 1
- ↦ (record constructor), 1
- / (division), 1
- \ (set difference), 1, 2, 3
- * (end-of-line comment), 2
- − (minus), 1
- ⊢—⊣ (section separator), 1
- − after PlusCal label, 1, 2
- ¬ (negation), 1
- ↪ (leads to), 1, 2
- ÷ (integer division), 1
- ⊕ (plus), 1
- × (Cartesian product), 1
- ⇒ (implies), 1
- ⇔ (equivalence), 1
- ≡ (equivalence), 1
 - versus =, 1
- .. (integer interval), 1, 2
- (action composition), 1, 2, 3
- * (multiplication), 1
- (sequence concatenation), 1
- (always)
 - semantics, 1
- formula, 1
- ↪ rule, 1
- ◇ (eventually)
 - semantics, 2
- < (less than), 1
- ≤ (less than or equal), 1
- (*...*) (comment), 1
- [[...]] (semantics), 1
- [A]_v (action operator), 1
- [h₁ ↦ e₁, ..., h_n ↦ e_n] (record constructor), 1
- [S → T] (set of functions), 1, 2
- [x ∈ S ↦ e] (function constructor), 1
- ⟨A⟩_v (action operator), 1, 2
- ⟨e₁, ..., e_n⟩ (tuple), 1, 2
- ⟨⊢⟩ (in proofs), 1
- ⟨*⟩ (in proofs), 1
- > (greater than), 1
- ≥ (greater than or equal), 1, 1
- ∧ (conjunction), 1
- ∨ (disjunction), 1
- ⋃, *see* UNION
- :: (in subexpression name), 1
- ⋮ (PlusCal separator), 1
- :> (function constructor), 1
- ! (in EXCEPT), 1
- ! (with instantiation), 1, 2
- ! (in subexpression name), 1
- % (modulus), 1, 2, 3
- @ (in proof step), 1
- @@ (function constructor), 1
- ∀, 1
- ∃, 1
- ∃ (temporal existential quantification), 1
- ε, Hilbert's, 1
- action, 1, 2, G
 - enabled, 1
 - next-state, 1
- action composition operator, 1, 2, 3
- action formula, G
- algorithm
 - deriving, 1
- algorithm** token, 1
- Alloy, 1
- alternation, 1
- and (conjunction), 1
- Append*, 2
- arithmetic, 1
- arithmetic, modular, 1

- array, [1](#) , [2](#)
- Arrow's theorem, [1](#)
- ASCII representation of symbols, [1](#)
- Assert (defined in *TLC* module), [1](#)
- assert** (PlusCal statement), [1](#)
- assertion, [1](#)
- ASSUME (proof construct), [1](#)
- ASSUME (TLA⁺ statement), [1](#)
- assume/prove, [1](#) , [2](#)
- assumption, domain, [1](#)
- atomic bakery algorithm, [1](#)
- atomicity
 - grain of, [2](#)
- auto (Isabelle method), [1](#)
- auxiliary variable, [1](#)
- await** (PlusCal statement), [1](#)
- back-end prover, [2](#)
- backend prover, [1](#)
- bag, [1](#)
- bakery algorithm, [1](#)
 - atomic, [1](#)
- BEGIN TRANSLATION, [1](#)
- behavior, [1](#) , [2](#) , [3](#) , [4](#) , [G](#)
 - computing, [1](#)
 - finite, [1](#)
 - infinite, [1](#)
- behavior specification, [1](#) , [2](#) , [G](#)
- behavioral proof, [1](#)
- big-step bakery algorithm, [1](#)
- blast (Isabelle method), [1](#)
- blueprint, [1](#)
- Boolean, [1](#)
- BOOLEAN (TLA⁺ statement), [1](#)
- Borda ranking, [1](#)
- bounded channel, [1](#)
- bounded quantification, [1](#)
- busy waiting, [1](#)
- BY, [1](#)
- BY proof, [1](#)
- Cardinality*, [1](#) , [2](#)
- Cartesian product (\times), [1](#)
- CASE (expression), [1](#) , [2](#) , [3](#)
- CASE (proof step), [1](#) , [2](#)
- CHOOSE, [1](#) , [2](#) , [3](#)
 - how TLC evaluates it, [1](#)
 - is deterministic, [1](#)
- Chord, [1](#)
- clarsimp (Isabelle method), [1](#)
- Clock
 - One-Bit, [1](#)
- cloning a model, [1](#)
- closing
 - of module, [1](#)
- coarser-grained algorithm, [1](#)
- comment
 - multi-line, [1](#)
- comments, [1](#)
 - boxed, [1](#)
 - pretty printed, [1](#)
- composition of relations, [1](#)
- computation, [1](#)
 - parallel, [1](#)
- concurrency, [1](#)
- concurrent, [1](#)
 - system, [1](#)
- Condorcet ranking, [1](#)
- conjunction, [1](#)
- CONSTANT, [1](#)
- constant declaration, [1](#)
- constant expression, evaluating, [1](#)
- CONSTANTS, [1](#)
- constraint, state, [1](#) , [2](#)
- consumer, [1](#)
- contradiction, proof by, [1](#) , [2](#)
- COROLLARY, [1](#)
- critical section, [1](#)
- cursor, [1](#)
- data item, elementary, [1](#)
- deadlock
 - versus termination, [1](#)
- deadlock freedom, [1](#)
- debugging, [1](#)
- decimal fraction, [1](#)
- declaration

TLA ⁺ variable, 1	for records, 1
Decompose Proof (Toolbox command), 1	exclusive or, 1
DEF, 1 , 2	exponentiation, 1
default backend provers, 1	expression, G
<i>defaultInitValue</i> , 1	state, 1 , G
DEFINE, 1	transition, 1
definition, 1	EXTENDS, 1
function, 1	where it goes, 1
inductive, 1	facts
local, 1	known, 1
overriding in TLC, 1	usable, 1
deriving algorithms, 1	fair
determinism	strongly, 1 , 9
of CHOOSE, 1	fair --algorithm , 1
Die Hard, 1	fair process , 1
Dijkstra, Edsger, 1	fairness, 1 , 1 , 2
dining philosophers problem, 1	strong, 1 , 9
directed graph, 1	weak, 1 , 2
disjunction, 1	weak, of next-state action, 1
divisor, 1	FALSE, 1
Doligez, Damien, 1	fast (Isabelle method), 1
DOMAIN, 1 , 2	fastsimp (Isabelle method), 1
domain, 1 , 2	FCFS, 1
domain assumption, 1	field (of a record), 1
dominating set, 1	FIFO (first in, first out), 1
dummy variable, 1	file
either (PlusCal keyword), 1	fingerprint, 1
elementary data item, 1	finer-grained algorithm, 1
ENABLED, 1	fingerprint file, 1
enabled, 1 , 2	first-come-first-served, 1
enabling condition, 1	flexible variable, 1
END TRANSLATION, 1	Floyd-Hoare method, 1
equivalence, 1	Floyd-Warshall algorithm, 1
Euclid's Algorithm, 1	Focus On Step (Toolbox command), 1
for sets, 1	force (Isabelle method), 1
TLAPLUS proof, 1	formula, G
evaluating a constant expression, 1	□, 1
evaluation, lazy, 1	action, G
event, 1	state, 1
eventually always, 1	temporal, G
EXCEPT, 1 , 2	Fraenkel, 1
	free symbol, 1

- function, [1](#) , [2](#)
 - definition, [1](#)
 - definition of, [1](#)
 - domain of, [1](#)
 - of multiple arguments, [1](#)
 - recursive definition of, [1](#) , [2](#)
 - use in proof, [1](#)
 - state, [1](#) , [G](#)
 - variant, [1](#)
 - versus array, [1](#) , [2](#)
 - versus operator, [1](#)
- functions, set of, [1](#)
- gcd, [1](#) , [2](#) , [3](#)
- goal, of proof, [1](#)
- Gonthier, Georges, [1](#)
- goto** (PlusCal statement), [1](#)
- Goto Declaration (Toolbox command), [1](#) , [2](#) , [3](#)
- goto line, [1](#)
- Goto Next Use (Toolbox command), [1](#)
- Goto PCal Source (Toolbox command), [1](#)
- Goto Prev Use (Toolbox command), [1](#)
- grain of atomicity, [2](#)
- graph
 - directed, [1](#)
 - of a relation, [1](#)
 - state, [1](#)
- greatest common divisor, *see* [gcd](#)
- halting, [1](#)
- handshake, [1](#)
- HAVE, [1](#)
- Head*, [1](#)
- HIDE, [1](#)
- Hide All Proofs (Toolbox command), [1](#)
- Hide Current Subtree (Toolbox command), [JavaTime](#), [1](#)
- hiding variables, [1](#) , [2](#)
- higher-order operator, [1](#) , [2](#) , [3](#)
- Hilbert's ε , [1](#)
- history variable, [1](#)
- Hyman, Harris, [1](#)
- identifier, process, [1](#)
- identity refinement mapping, [1](#)
- IF... THEN... ELSE, [1](#) , [2](#)
- IF versus **if**, [1](#)
- iff, [1](#)
- implementation
 - checking with TLC, [1](#)
- implements under refinement mapping, [1](#)
- implication, [1](#)
- implies, [1](#)
- index set, [1](#) , [1](#)
- induction, leads-to, [1](#)
- inductive definition, [1](#) , [2](#)
- inductive invariant, [1](#) , [2](#) , [G](#)
 - checking with TLC, [1](#)
- infinite sequence, [1](#)
- infinitely often, [1](#)
- initial predicate, [1](#)
- input/output (I/O) specification, [1](#)
- insensitive to stuttering, [1](#)
- INSTANCE, [1](#) , [2](#)
 - theorems imported by, [1](#)
- INSTANCE (proof step), [1](#)
- instantiates, [1](#)
- instantiation
 - overriding definitions with, [1](#)
- Int*, [1](#) , [2](#)
- Integers* module, [1](#) , [2](#)
- INV1 (proof rule), [1](#)
- INV2 (proof rule), [1](#)
- invariant, [1](#) , [G](#)
 - inductive, [1](#) , [2](#)
 - type correctness, [1](#)
- Isa* (BY fact), [1](#)
- Isabelle* (backend prover), [1](#) , [2](#)
- Knuth, Donald Ervin, [1](#)
- label, in subexpression name, [1](#)
- labels, PlusCal, [1](#)
 - rules for, [1](#) , [2](#)

- LAMBDA, [1](#) , [2](#)
- largest enumerable set, [1](#)
- lattice rule, [1](#)
- Launch Prover (Toolbox command), [1](#)
- lazy evaluation, [1](#)
- leads to, [1](#)
- leads-to induction rule, [1](#)
- leaf proof, [1](#)
- LEMMA, [1](#)
- Len*, [2](#)
- LET, [1](#)
 - optimizing TLC execution with, [1](#)
- level specifier, [1](#)
- lexicographical ordering, [1](#) , [2](#)
- line numbers, [1](#)
- line, going to a, [1](#)
- list, [1](#)
- livelock, [1](#)
- liveness property, [1](#) , [2](#)
- lock, [1](#)
- logic
 - mathematical, [1](#)
 - predicate, [1](#)
 - propositional, [1](#)
- LS4, [1](#)
- mapping
 - refinement, [1](#)
- math
 - not, [1](#)
 - ordinary, [1](#)
- maximum
 - of set of numbers, [1](#)
- Merz, Stephan, [1](#)
- model
 - cloning, [1](#)
 - creating, [1](#)
 - editor, [1](#)
 - standard, [1](#)
 - system, [1](#)
- model value, [1](#) , [2](#)
- modular arithmetic, [1](#)
- module closing, [1](#)
- module opening, [1](#)
- module, special, [1](#)
- modulus operator, [1](#) , [2](#) , [4](#)
- mouse pointer, [1](#)
- multi-assignment, PlusCal, [1](#)
- multi-line comment, [1](#)
- multiplication, [1](#)
- multiset, [2](#)
- mutual exclusion, [1](#)
 - in modern computers, [1](#)
- name, step, [1](#)
- Nat*, [1](#) , [2](#)
- Naturals* module, [1](#)
- negation, [1](#)
- NEW, [1](#)
- next states, too many, [1](#)
- next-state action, [1](#)
- next-state relation, *see* [next-state action](#)
- non-enumerable quantifier bound, [1](#)
- non-leaf proof, [1](#)
- noncritical section, [1](#)
- nondeterminism, [2](#)
- not (negation), [1](#)
- number, [1](#)
 - unequal to string, [1](#)
- numbering lines, [1](#)
- obligation
 - proof, [1](#) , [2](#)
- observable variable, [1](#)
- OMITTED, [1](#)
- One-Bit Algorithm
 - N*-Process, [1](#)
 - Two-Process, [1](#)
- One-Bit Clock, [1](#)
- One-Bit Protocol, [1](#)
- opening
 - of module, [1](#)
- opening a new spec in Toolbox, [1](#)
- operator
 - higher-order, [1](#) , [2](#) , [3](#)
 - recursive definition of, [1](#) , [2](#)

- use in proof, [1](#)
- versus function, [1](#)
- or (disjunction), [1](#)
- or** (PlusCal keyword), [1](#)
- order
 - lexicographic, [1](#) , [2](#)
 - partial, [1](#)
 - total, [1](#)
- ordinary math, [1](#)
- TLA⁺ versus, [1](#)
- overriding a definition in TLC, [1](#)
- \mathcal{P} , [1](#)
- parallel computation, [1](#)
- parser preferences, [1](#)
- partial correctness, [1](#)
- partial order, [1](#)
- path, [1](#)
- Paulson, Lawrence, [1](#)
- pc* variable, [1](#)
- pdflatex**, [1](#)
- PICK proof step, [1](#)
- PlusCal, [1](#)
 - labels in, [1](#)
 - translator, [1](#)
- power set, [1](#)
- predicate
 - initial, [1](#)
 - state, [1](#) , [G](#)
- predicate logic, [1](#)
- preference
 - numbering lines, [1](#)
- pretty printing, [1](#)
 - comments, [1](#)
- prime (')
 - of a variable, [1](#)
- Print*, [1](#)
- print statement, [1](#)
- PrintT*, [1](#)
- process, [1](#)
 - PlusCal, [1](#)
 - variable local to, [1](#)
 - versus thread, [1](#)

- process** (PlusCal declaration), [1](#)
- process identifier, [1](#)
- producer, [1](#)
- program control variable, [1](#)
- proof
 - behavioral, [1](#)
 - BY, [1](#)
 - correct, [1](#)
 - goal, [1](#)
 - manager, [1](#)
 - mathematicians', [1](#)
 - non-leaf, [1](#)
 - nonleaf, [1](#)
 - reading, [1](#)
 - state, [1](#)
 - structured, [1](#) , [2](#)
 - TLA⁺, [1](#)
- proof manager, [1](#)
- proof obligation, [1](#) , [2](#)
- proof rule, [1](#)
- proof step
 - green, [2](#)
- property
 - checking with TLC, [1](#)
 - liveness, [1](#) , [2](#)
 - safety, [1](#) , [2](#)
- PROPOSITION, [1](#)
- propositional logic, [1](#)
- provability, [1](#)
- PROVE, [1](#)
- Prove (Toolbox command), [1](#)
- Prove Step or Module (Toolbox command), [1](#)
- prover
 - back-end, [2](#)
 - backend, [1](#)
- PTL* (backend prover), [1](#) , [2](#)
- Q.E.D., [1](#)
- QED, [1](#)
- quantification
 - bounded, [1](#)
 - unbounded, [1](#)

- quantifier, [1](#)
- RandomElement*, [1](#)
- reachable state, [1](#) , [G](#)
- reactive system, [1](#)
- Reals* module, [1](#) , [1](#)
- record, [1](#)
- RECURSIVE, [1](#)
- recursive
 - function definition, [1](#) , [2](#)
 - use in proof, [1](#)
 - operator definition, [1](#) , [2](#)
 - in proof, [1](#)
 - versus inductive, [1](#)
- refinement mapping, [1](#)
 - identity, [1](#)
- relation, [1](#)
 - composition, [1](#)
 - well-founded, [1](#) , [2](#)
- Renumber Proof (Toolbox command), [2](#) , [2](#)
- resizing TLC Errors view, [1](#)
- Return from Goto Declaration (Toolbox command), [1](#) , [2](#)
- rigid variable, [1](#)
- round robin synchronization, [1](#)
- rule
 - $\Box \leadsto$, [1](#)
 - \Vdash , [1](#)
 - INV1, [1](#)
 - INV2, [1](#)
 - lattice, [1](#)
 - leads-to induction, [1](#)
 - SF1, [1](#)
 - SF2, [1](#)
 - WF1, [1](#) , [2](#)
 - WF2, [1](#)
- rules
 - labeling, [1](#) , [2](#)
- running TLC model checker, [1](#)
- safety property, [1](#) , [2](#)
- satisfies, [1](#)
- saving a module in Toolbox, [1](#)
- section
 - critical, [1](#)
 - noncritical, [1](#)
- section separator, [1](#)
- SelectSeq*, [1](#)
- semantics
 - of temporal formula, [1](#)
- semicolon (;) (PlusCal separator), [1](#)
- Seq*, [2](#)
- sequence
 - finite, [1](#) , [2](#)
 - infinite, [1](#)
- Sequences* module, [1](#) , [2](#)
- set
 - of functions, [1](#)
- set difference, [2](#) , [3](#)
- set theory, [1](#)
- SF, [1](#) , [2](#)
- SF1 (proof rule), [1](#)
- SF2 (proof rule), [1](#)
- Show All Proofs (Toolbox command), [1](#)
- Show Children Only (Toolbox command), [1](#)
- Show Current Subtree (Toolbox command), [1](#)
- Show Declarations (Toolbox command), [1](#)
- Show Uses (Toolbox command), [1](#)
- simp (Isabelle method), [1](#)
- simulation, step, [1](#)
- single access rule, [1](#)
- skip** (PlusCal statement), [1](#)
- SMT* (backend prover), [1](#)
- SMT solver, [1](#)
- solver, SMT, [1](#)
- sorting, [1](#)
- SortSeq*, [1](#)
- specification, [1](#) , [G](#)
 - behavior, [1](#) , [2](#)
 - input/output (I/O), [1](#)
 - two meanings of, [1](#)
- standard arithmetic modules, [1](#)
- standard model, [1](#)

- starvation freedom, 1
- state, 1, 2, 3, G
 - assigns values to all variables, 1
 - of a proof, 1
 - reachable, 1
 - system, 1
- state constraint, 2
- state constraint, in TLC, 1
- state expression, 1, G
- state formula, 1
- state function, 1, G
- state graph, 1
- state predicate, 1, 2, G
- state space progress table, 1
- step, 1, 2, 3, G
 - stuttering, 1
- step name, 1
- step simulation, 1
- store, 1
- string, 1
 - unequal to number, 1
- strong fairness, 1, 2, 9
- strongly fair, 1, 9
- structured proof, 1
- stuttering
 - insensitive to, 1
- stuttering step, 1
- subaction, 1, G
- subexpression names, 1, 2
- SubSeq*, 1
- SUBSET, 1, 2
- SUFFICES, 1
- symbol
 - ASCII representation, 1
 - free, 1
 - user-definable, 1
- synchronization primitive, 1
- system, 1
 - abstract, 1
 - concurrent, 1
 - reactive, 1
 - state, 1

Tail, 2

- TAKE, 1
- tautology, 1
- temporal formula, G
 - semantics, 1
- termination
 - versus deadlock, 1
- THEOREM, 1
- theorem, temporal, 1, 9
- thread, 1
- time, measuring with TLC, 1
- timeout, prover, 1
- TLA, 1
- TLAPS, 1, 2
 - what it checks, 1
- TLC, 1
 - checking implementation with, 1
 - debugging with, 1
 - how it evaluates CHOOSE, 1
 - overriding definition in, 1
 - running, 1
 - setting and reading values, 1
 - state constraint, 1
 - trace explorer, 1, 9
- TLC Errors view
 - resizing, 1
- TLC* module, 1
- TLCEval*, 1
- TLCGet*, 1
- TLCSet*, 1
- Toolbox, 1
 - opening a new spec in, 1
 - saving a module in, 1
 - TLA⁺ parser preferences, 1
- total order, 1
- trace explorer, 1, 9
- transition expression, 1
- transitive closure, 1
- TRUE, 1
- truth, 1
- truth value, 1
- tuple, 1, 2, 3
 - domain of, 1
 - same as sequence, 1, 2

- two-phase handshake, [1](#)
- type correctness, [1](#)
 - invariant, [1](#)
- unbounded quantification, [1](#)
- UNCHANGED, [1](#)
- undefined or not an operator, [1](#)
- understanding, [1](#)
- UNION, [1](#)
- USE, [1](#)
- validity, [1](#)
- value
 - model, [1](#) , [9](#)
 - truth, [1](#)
- variable
 - auxiliary, [1](#)
 - flexible, [1](#)
 - history, [1](#)
 - observable, [1](#)
 - process-local, [1](#)
 - rigid, [1](#)
- VARIABLE, [1](#)
- variable** (PlusCal keyword), [1](#)
- variable declaration (TLA⁺), [1](#)
- variable hiding, [1](#) , [2](#)
- variant function, [1](#)
- vars*, [1](#)
- waiting, busy, [1](#)
- Warshall's algorithm, [1](#)
- weak fairness, [1](#) , [2](#)
 - of next-state action, [1](#)
- well-founded, [1](#) , [2](#)
- WF, [1](#) , [2](#)
- WF1 (proof rule), [1](#) , [2](#)
- WF2 (proof rule), [1](#)
- while** (PlusCal statement), [1](#)
- with** (PlusCal keyword), [1](#) , [2](#)
- WITNESS proof step, [1](#)
- Zave, Pamela, [1](#)
- Zenon* (backend prover), [1](#) , [2](#)
- Zermelo-Fraenkel set theory, [1](#)