a Practical Theory of Programming

2025-4-10 edition

Eric C.R. Hehner

Department of Computer Science
University of Toronto
Toronto ON M5S 2E4 Canada

The first edition of this book was published by
Springer-Verlag Publishers, New York, 1993
ISBN 0-387-94106-1 QA76.6.H428

The current edition is available free at
hehner.ca/aPToP

An on-line course based on this book is at
hehner.ca/FMSD

The author's website is
hehner.ca

The cover picture is an inukshuk, which is a human-like figure made of piled stones. Inukshuks are found throughout arctic Canada. They are built by the Inuit people, who use them to mean “You are on the right path.”.

11.3 Laws

11.3.0 Generic

The operators $=$ $\neq$ **if then else fi** apply to every type of expression (but the first operand of **if then else fi** must be binary), with the laws

$x=x$	Reflexivity	**if** $\top$ **then** x **else** y **fi** $= x$	Case Base
$x=y \;=\; y=x$	Symmetry	**if** $\bot$ **then** x **else** y **fi** $= y$	Case Base
$x=y \wedge y=z \;\Rightarrow\; x=z$	Transitivity	**if** a **then** x **else** x **fi** $= x$	Case Idempotent
$x=y \;\Rightarrow\; f\,x = f\,y$	Transparency	**if** a **then** x **else** y **fi** $=$ **if** $\neg a$ **then** y **else** x **fi**	Case Reversal
$x \neq y \;=\; \neg(x=y)$	Unequality		

The operators $\uparrow$ $\downarrow$ $<$ $\le$ $>$ $\ge$ apply to numbers, characters, strings, and lists, with the laws

$x \le x$	Reflexivity	$\neg\, x<x$	Irreflexivity
$\neg(x<y \wedge x=y)$	Exclusivity	$\neg(x>y \wedge x=y)$	Exclusivity
$\neg(x<y \wedge x>y)$	Exclusivity	$x \le y \;=\; x<y \vee x=y$	Inclusivity
$x \le y \wedge y \le z \Rightarrow x \le z$	Transitivity	$x<y \wedge y \le z \Rightarrow x<z$	Transitivity
$x<y \wedge y<z \Rightarrow x<z$	Transitivity	$x \le y \wedge y<z \Rightarrow x<z$	Transitivity
$x>y \;=\; y<x$	Mirror	$x \ge y \;=\; y \le x$	Mirror
$\neg\, x<y \;=\; x \ge y$	Totality	$\neg\, x \le y \;=\; x>y$	Totality
$x \le y \wedge y \le x \;=\; x=y$	Antisymmetry	$x<y \vee x=y \vee x>y$	Totality, Trichotomy
$x \uparrow x = x$	Idempotence	$x \downarrow x = x$	Idempotence
$x \uparrow y = y \uparrow x$	Symmetry	$x \downarrow y = y \downarrow x$	Symmetry
$x\uparrow(y\uparrow z) = (x\uparrow y)\uparrow z$	Associativity	$x\downarrow(y\downarrow z) = (x\downarrow y)\downarrow z$	Associativity
$x\uparrow(y\downarrow z) = (x\uparrow y)\downarrow(x\uparrow z)$	Distributivity	$x\downarrow(y\uparrow z) = (x\downarrow y)\uparrow(x\downarrow z)$	Distributivity
$x\uparrow y \le z \;=\; x \le z \wedge y \le z$	Connection	$x\downarrow y \le z \;=\; x \le z \vee y \le z$	Connection
$x \le y\uparrow z \;=\; x \le y \vee x \le z$	Connection	$x \le y\downarrow z \;=\; x \le y \wedge x \le z$	Connection
$x\uparrow y \;=$ **if** $x \ge y$ **then** x **else** y **fi**		$x\downarrow y \;=$ **if** $x \le y$ **then** x **else** y **fi**	
$x\downarrow y \le x \le x\uparrow y$			

End of Generic

11.3.1 Binary

Let a , b , c , d , and e be binary.

Binary

$$\top$$
$$\neg\bot$$
$$\top \neq \bot$$

Excluded Middle

$$a \vee \neg a$$

Noncontradiction

$$\neg(a \wedge \neg a)$$

Base

$$\neg(a \wedge \bot)$$
$$a \vee \top$$
$$a \Rightarrow \top$$
$$\bot \Rightarrow a$$

Mirror

$$a \Leftarrow b \;=\; b \Rightarrow a$$

Double Negation

$$\neg\neg a \;=\; a$$

Duality

$$\neg(a \wedge b) \;=\; \neg a \vee \neg b$$
$$\neg(a \vee b) \;=\; \neg a \wedge \neg b$$

Exclusion

$$a \Rightarrow \neg b \;=\; b \Rightarrow \neg a \qquad \text{(Contrapositive)}$$
$$a = \neg b \;=\; a \neq b \;=\; \neg a = b$$

Inclusion

$$a \Rightarrow b \;=\; \neg a \vee b \text{ (Material Implication)}$$
$$a \Rightarrow b \;=\; (a \wedge b \;=\; a)$$
$$a \Rightarrow b \;=\; (a \vee b \;=\; b)$$

Identity

$\top \land a \;=\; a$
$\bot \lor a \;=\; a$
$\top \Rightarrow a \;=\; a$
$\top = a \;=\; a$

Idempotent

$a \land a \;=\; a$
$a \lor a \;=\; a$

Reflexive

$a \Rightarrow a$
$a = a$

Indirect Proof

$\neg a \Rightarrow \bot \;=\; a$
$\neg a \Rightarrow a \;=\; a$

Specialization

$a \land b \Rightarrow a$

Associative

$a \land (b \land c) \;=\; (a \land b) \land c$
$a \lor (b \lor c) \;=\; (a \lor b) \lor c$
$a = (b = c) \;=\; (a = b) = c$
$a \ne (b \ne c) \;=\; (a \ne b) \ne c$
$a = (b \ne c) \;=\; (a = b) \ne c$

Symmetry (Commutative)

$a \land b \;=\; b \land a$
$a \lor b \;=\; b \lor a$
$a = b \;=\; b = a$
$a \ne b \;=\; b \ne a$

Antisymmetry (Double Implication)

$(a \Rightarrow b) \land (b \Rightarrow a) \;=\; a = b$

Discharge

$a \land (a \Rightarrow b) \;=\; a \land b$
$a \Rightarrow (a \land b) \;=\; a \Rightarrow b$

Antimonotonic

$a \Rightarrow b \;=\; \neg a \Leftarrow \neg b$ (Contrapositive)
$a \Rightarrow b \;\Longrightarrow\; (a \Rightarrow c) \Leftarrow (b \Rightarrow c)$

Monotonic

$a \Rightarrow b \;\Longrightarrow\; a \land c \Rightarrow b \land c$
$a \Rightarrow b \;\Longrightarrow\; a \lor c \Rightarrow b \lor c$
$a \Rightarrow b \;\Longrightarrow\; (c \Rightarrow a) \Rightarrow (c \Rightarrow b)$

Absorption

$a \land (a \lor b) \;=\; a$
$a \lor (a \land b) \;=\; a$

Direct Proof

$(a \Rightarrow b) \land a \Rightarrow b$
$(a \Rightarrow b) \land \neg b \Rightarrow \neg a$
$(a \lor b) \land \neg a \Rightarrow b$

Transitive

$(a \land b) \land (b \land c) \Rightarrow (a \land c)$
$(a \Rightarrow b) \land (b \Rightarrow c) \Rightarrow (a \Rightarrow c)$
$(a = b) \land (b = c) \Rightarrow (a = c)$
$(a \Rightarrow b) \land (b = c) \Rightarrow (a \Rightarrow c)$
$(a = b) \land (b \Rightarrow c) \Rightarrow (a \Rightarrow c)$

Distributive (Factoring)

$a \land (b \land c) \;=\; (a \land b) \land (a \land c)$
$a \land (b \lor c) \;=\; (a \land b) \lor (a \land c)$
$a \lor (b \land c) \;=\; (a \lor b) \land (a \lor c)$
$a \lor (b \lor c) \;=\; (a \lor b) \lor (a \lor c)$
$a \lor (b \Rightarrow c) \;=\; (a \lor b) \Rightarrow (a \lor c)$
$a \lor (b = c) \;=\; (a \lor b) = (a \lor c)$
$a \Rightarrow (b \land c) \;=\; (a \Rightarrow b) \land (a \Rightarrow c)$
$a \Rightarrow (b \lor c) \;=\; (a \Rightarrow b) \lor (a \Rightarrow c)$
$a \Rightarrow (b \Rightarrow c) \;=\; (a \Rightarrow b) \Rightarrow (a \Rightarrow c)$
$a \Rightarrow (b = c) \;=\; (a \Rightarrow b) = (a \Rightarrow c)$

Generalization

$a \Rightarrow a \lor b$

Antidistributive

$a \land b \Rightarrow c \;=\; (a \Rightarrow c) \lor (b \Rightarrow c)$
$a \lor b \Rightarrow c \;=\; (a \Rightarrow c) \land (b \Rightarrow c)$

Portation

$a \land b \Rightarrow c \;=\; a \Rightarrow (b \Rightarrow c)$
$a \land b \Rightarrow c \;=\; a \Rightarrow \neg b \lor c$

Conflation

$(a \Rightarrow b) \land (c \Rightarrow d) \;\Longrightarrow\; a \land c \Rightarrow b \land d$
$(a \Rightarrow b) \land (c \Rightarrow d) \;\Longrightarrow\; a \lor c \Rightarrow b \lor d$

Equality and Difference

$a{=}b \;=\; (a \land b) \lor (\neg a \land \neg b)$
$a{\ne}b \;=\; (a \land \neg b) \lor (\neg a \land b)$

Resolution

$a \wedge c \;\Rightarrow\; (a \vee b) \wedge (\neg b \vee c) \;=\; (a \wedge \neg b) \vee (b \wedge c) \;\Rightarrow\; a \vee c$

Case Creation

$a \;=\;$ **if** b **then** $b{\Rightarrow}a$ **else** $\neg b \Rightarrow a$ **fi**
$a \;=\;$ **if** b **then** $b{\wedge}a$ **else** $\neg b \wedge a$ **fi**
$a \;=\;$ **if** b **then** $b{=}a$ **else** $b{\neq}a$ **fi**

Case Analysis

if a **then** b **else** c **fi** $=$ $(a \wedge b) \vee (\neg a \wedge c)$
if a **then** b **else** c **fi** $=$ $(a{\Rightarrow}b) \wedge (\neg a \Rightarrow c)$

One Case

if a **then** $\top$ **else** b **fi** $=$ $a{\vee}b$
if a **then** $\bot$ **else** b **fi** $=$ $\neg a \wedge b$
if a **then** b **else** $\top$ **fi** $=$ $a{\Rightarrow}b$
if a **then** b **else** $\bot$ **fi** $=$ $a{\wedge}b$
if a **then** b **else** $\neg b$ **fi** $=$ $a{=}b$
if a **then** $\neg b$ **else** b **fi** $=$ $a{\neq}b$

Case Absorption

if a **then** b **else** c **fi** $=$ **if** a **then** $a{\wedge}b$ **else** c **fi**
if a **then** b **else** c **fi** $=$ **if** a **then** $a{\Rightarrow}b$ **else** c **fi**
if a **then** b **else** c **fi** $=$ **if** a **then** $a{=}b$ **else** c **fi**
if a **then** b **else** c **fi** $=$ **if** a **then** b **else** $\neg a \wedge c$ **fi**
if a **then** b **else** c **fi** $=$ **if** a **then** b **else** $a{\vee}c$ **fi**
if a **then** b **else** c **fi** $=$ **if** a **then** b **else** $a{\neq}c$ **fi**

Case Distributive (Case Factoring)

$\neg$ **if** a **then** b **else** c **fi** $=$ **if** a **then** $\neg b$ **else** $\neg c$ **fi**
if a **then** b **else** c **fi** $\wedge\, d$ $=$ **if** a **then** $b{\wedge}d$ **else** $c{\wedge}d$ **fi**
and similarly replacing $\wedge$ by any of $\vee$ $=$ $\neq$ $\Rightarrow$ $\Leftarrow$
if a **then** $b{\wedge}c$ **else** $d{\wedge}e$ **fi** $=$ **if** a **then** b **else** d **fi** $\wedge$ **if** a **then** c **else** e **fi**
and similarly replacing $\wedge$ by any of $\vee$ $=$ $\neq$ $\Rightarrow$ $\Leftarrow$

End of Binary

11.3.2 Numbers

Let d be a sequence of (zero or more) digits, and let x , y , and z be numbers.

$d0{+}1 = d1$	$d5{+}1 = d6$	Counting (see Exercise 32)
$d1{+}1 = d2$	$d6{+}1 = d7$	Counting
$d2{+}1 = d3$	$d7{+}1 = d8$	Counting
$d3{+}1 = d4$	$d8{+}1 = d9$	Counting
$d4{+}1 = d5$	$9{+}1 = 10$	Counting
for nonempty d	$d9{+}1 = (d{+}1)0$	Counting
$x{+}0 = x$		Identity
$x{+}y = y{+}x$		Symmetry
$x{+}(y{+}z) \;=\; (x{+}y){+}z$		Associativity
$-\infty{<}x{<}\infty \Rightarrow (x{+}y = x{+}z \;=\; y{=}z)$		Cancellation
$-\infty{<}x \Rightarrow\; \infty{+}x = \infty$		Absorption
$x{<}\infty \;\Rightarrow\; -\infty + x = -\infty$		Absorption
$-x \;=\; 0{-}x$		Negation
$- -x = x$		Self-inverse
$-(x{+}y) = -x + -y$		Distributivity
$-(x{-}y) = y{-}x$		Antisymmetry
$-x \times y \;=\; -(x{\times}y) \;=\; x \times -y$		Semi-distributivity
$-x \,/\, y \;=\; -(x/y) \;=\; x \,/\, -y$		Semi-distributivity
$x{-}0 = x$		Identity
$x{-}y \;=\; x + -y$		Subtraction
$x{+}(y{-}z) = (x{+}y){-}z$		Associativity
$-\infty{<}x{<}\infty \;\Rightarrow\; (x{-}y = x{-}z \;=\; y{=}z)$		Cancellation
$-\infty{<}x{<}\infty \;\Rightarrow\; x{-}x = 0$		Inverse
$x{<}\infty \;\Rightarrow\; \infty{-}x = \infty$		Absorption
$-\infty{<}x \;\Rightarrow\; -\infty - x = -\infty$		Absorption
$-\infty{<}x{<}\infty \;\Rightarrow\; x{\times}0 = 0$		Base
$x{\times}1 = x$		Identity

$x \times y = y \times x$	Symmetry
$x \times (y+z) = x \times y + x \times z$	Distributivity
$x \times (y \times z) = (x \times y) \times z$	Associativity
$-\infty<x<\infty \wedge x \neq 0 \Rightarrow (x \times y = x \times z \;=\; y=z)$	Cancellation
$0<x \Rightarrow x \times \infty = \infty$	Absorption
$0<x \Rightarrow x \times -\infty = -\infty$	Absorption
$x/1 = x$	Identity
$x \neq 0 \Rightarrow 0/x = 0$	Base
$-\infty<x<\infty \wedge x \neq 0 \Rightarrow x/x = 1$	Base
$x \times (y/z) = (x \times y)/z = (x/z) \times y = x/(z/y)$	Multiplication-Division
$(x/y)/z = x/(y \times z)$	Multiplication-Division
$-\infty<y<\infty \wedge y \neq 0 \Rightarrow (x/y) \times y = x$	Multiplication-Division
$-\infty<x<\infty \Rightarrow x/\infty = 0 = x/-\infty$	Annihilation
$-\infty<x<\infty \Rightarrow x^0 = 1$	Base
$x^1 = x$	Identity
$x^{y+z} = x^y \times x^z$	Adding Exponents
$-\infty<0<1<\infty$	Direction
$x<y \;=\; -y<-x$	Reflection
$-\infty<x<\infty \Rightarrow (x+y < x+z \;=\; y<z)$	Cancellation, Translation
$0<x<\infty \Rightarrow (x \times y < x \times z \;=\; y<z)$	Cancellation, Scale
$x<y \vee x=y \vee x>y$	Trichotomy
$-\infty \le x \le \infty$	Extremes

$x \uparrow \infty = \infty$	$x \downarrow -\infty = -\infty$	Base
$x \uparrow -\infty = x$	$x \downarrow \infty = x$	Identity
$-(x \uparrow y) = -x \downarrow -y$	$-(x \downarrow y) = -x \uparrow -y$	Duality
$x + y \uparrow z = (x+y) \uparrow (x+z)$	$x - y \uparrow z = (x-y) \downarrow (x-z)$	Distributivity
$x \ge 0 \Rightarrow x \times (y \uparrow z) = (x \times y) \uparrow (x \times z)$	$x \ge 0 \Rightarrow x \times (y \downarrow z) = (x \times y) \downarrow (x \times z)$	Distributivity
$x \le 0 \Rightarrow x \times (y \uparrow z) = (x \times y) \downarrow (x \times z)$	$x \le 0 \Rightarrow x \times (y \downarrow z) = (x \times y) \uparrow (x \times z)$	Distributivity

End of Numbers

11.3.3 Bunches

Let x and y be elements (binaries, numbers, characters, sets, strings and lists of elements).

$x: y \;=\; x=y$	Elementary
$x: A, B \;=\; x: A \vee x: B$	Compound
$A, A = A$	Idempotence
$A, B = B, A$	Symmetry
$A, (B, C) = (A, B), C$	Associativity
$A‘A = A$	Idempotence
$A‘B = B‘A$	Symmetry
$A‘(B‘C) = (A‘B)‘C$	Associativity
$A, B: C \;=\; A: C \wedge B: C$	Antidistributivity
$A: B‘C \;=\; A: B \wedge A: C$	Distributivity
$A: A, B$	Generalization
$A‘B: A$	Specialization
$A: A$	Reflexivity
$A: B \wedge B: A \;=\; A=B$	Antisymmetry
$A: B \wedge B: C \Rightarrow A: C$	Transitivity
$A:: B \;=\; B: A$	Mirror
$¢\, null = 0$	Size

$\text{¢}\, x \;=\; 1$	Size
$\text{¢}\, nat \;=\; \infty$	Size
$\text{¢}(A, B) + \text{¢}(A\text{‘}B) \;=\; \text{¢}A + \text{¢}B$	Size
$\neg\, x: A \;\;=\;\; \text{¢}(A\text{‘}x) = 0$	Size
$A: B \;\Rightarrow\; \text{¢}A \le \text{¢}B$	Size
$A, (A\text{‘}B) \;=\; A \;=\; A\text{‘}(A, B)$	Absorption
$A: B \;\;=\;\; A, B = B \;\;=\;\; A = A\text{‘}B$	Inclusion
$A, (B, C) \;=\; (A, B), (A, C)$	Distributivity
$A, (B\text{‘}C) \;=\; (A, B)\text{‘}(A, C)$	Distributivity
$A\text{‘}(B, C) \;=\; (A\text{‘}B), (A\text{‘}C)$	Distributivity
$A\text{‘}(B\text{‘}C) \;=\; (A\text{‘}B)\text{‘}(A\text{‘}C)$	Distributivity
$A: B \;\wedge\; C: D \;\Rightarrow\; A, C: B, D$	Conflation, Monotonicity
$A: B \;\wedge\; C: D \;\Rightarrow\; A\text{‘}C: B\text{‘}D$	Conflation, Monotonicity
$null: A$	Induction
$A, null \;=\; A \;=\; null, A$	Identity
$A\text{‘}null \;=\; null \;=\; null\text{‘}A$	Base
$\text{¢}A = 0 \;\;=\;\; A = null$	Size
$x, y: xint \;\wedge\; x{\le}y \;\Rightarrow\; (i: x,..y \;\;=\;\; i: xint \;\wedge\; x{\le}i{<}y)$	Interval
$x, y: xint \;\wedge\; x{\le}y \;\Rightarrow\; \text{¢}(x,..y) \;=\; y{-}x$	Interval
$nat = 0,..\infty$	Interval
$\infty, -\infty:\; x/0$	Division by 0
$xreal: 0/0$	Division by 0
$x^{y\times z}:\; (x^y)^z$	Multiplying Exponents
$-null \;=\; null$	Distribution
$-(A, B) \;=\; -A, -B$	Distribution
$A{+}null \;=\; null \;=\; null{+}A$	Distribution
$(A, B){+}(C, D) \;=\; A{+}C, A{+}D, B{+}C, B{+}D$	Distribution

and similarly for many other operators (see the final page of the book)

End of Bunches

11.3.4 Sets

Let S be a set.

$\{{\sim}S\} \;=\; S$

${\sim}\{A\} \;=\; A$

$\{A\} \;\neq\; A$

$A \in \{B\} \;\;=\;\; A: B$

$\{A\} \subseteq \{B\} \;\;=\;\; A: B$

$\{A\}: \text{ϟ}B \;\;=\;\; A: B$

$\$\{A\} = \text{¢}A$

$\{A\} \cup \{B\} \;=\; \{A, B\}$

$\{A\} \cap \{B\} \;=\; \{A\text{ ‘ }B\}$

$\{A\} = \{B\} \;\;=\;\; A = B$

$\{A\} \neq \{B\} \;\;=\;\; A \neq B$

End of Sets

11.3.5 Strings

Let S, T, and U be strings; let i and j be items (binary values, numbers, characters, sets, lists, functions); let n and m be extended natural; let x, y, and z be extended integers such that $x{\le}y{\le}z$.

$S;\, nil \;=\; S \;=\; nil;\, S$

$S;\, (T;\, U) \;=\; (S;\, T);\, U$

$\leftrightarrow nil \;=\; 0$

$\leftrightarrow i \;=\; 1$

$\leftrightarrow(S;\, T) \;=\; \leftrightarrow S + \leftrightarrow T$

$S_{(T_U)} \;=\; (S_T)_U$

$S_{nil} \;=\; nil$

$S_{T;\, U} \;=\; S_T;\, S_U$

$S_{\{A\}} \;=\; \{S_A\}$

$\leftrightarrow S{<}\infty \;\Rightarrow\; nil \;\le\; S \;<\; S;\, i;\, T$

$¢\,nil = 1$
$¢(A;\,B) \le ¢A \times ¢B$
$\leftrightarrow S<\infty \;\Rightarrow\; (S;\,i;\,T)_{\leftrightarrow S} = i$
$\leftrightarrow S<\infty \;\Rightarrow\; S;i;T \triangleleft \leftrightarrow S \triangleright j = S;j;T$
$0*S = nil$
$(n+1)*S = n*S;\,S$
$*S = **S = nat*S$

$\leftrightarrow S<\infty \wedge i<j \;\Rightarrow\; S;\,i;\,T < S;\,j;\,U$
$\leftrightarrow S<\infty \;\Rightarrow\; (S;A;T : S;B;T \;=\; A:\,B)$
$\leftrightarrow S<\infty \;\Rightarrow\; (i=j \;=\; S;\,i;\,T = S;\,j;\,T)$
$(S \triangleleft n \triangleright i)_m = \textbf{if } n=m \textbf{ then } i \textbf{ else } S_m \textbf{ fi}$
$-\infty<x<\infty \;\Rightarrow\; x;..x = nil$
$-\infty<x<\infty \;\Rightarrow\; x;..x+1 = x$
$(x;..y)\,;\,(y;..z) = x;..z$
$\leftrightarrow(x;..y) = y-x$

End of Strings

11.3.6 Lists

Let S and T be strings; let i be an item (binary value, number, character, set, list, function); let L, M, and N be lists; let n and m be extended natural.

$[S] + S = \sim[S]$
$[\sim L] = L$
$[S];;[T] = [S;\,T]$
$[S] = [T] \;=\; S = T$
$[S] < [T] \;=\; S < T$
$[A]:\,[B] \;=\; A:\,B$
$\#[S] = \leftrightarrow S$
$nil \rightarrow i \mid L = i$
$n \rightarrow i \mid [S] = [S \triangleleft n \triangleright i]$
$(n \rightarrow i \mid L)\; m = \textbf{if } n=m \textbf{ then } i \textbf{ else } L\, m \textbf{ fi}$
$(S;\,T) \rightarrow i \mid L = S \rightarrow (T \rightarrow i \mid L@S) \mid L$

$\Box L = 0,..\#L$
$[S]\; T = S_T$
$S_{[T]} = [S_T]$
$[S]\; [T] = [S_T]$
$L\,\{A\} = \{L\,A\}$
$L\,[S] = [L\,S]$
$(L\,M)\;N = L\,(M\,N)$
$\#L = ¢\Box L$
$L@nil = L$
$L@i = L\,i$
$L@(S;\,T) = L@S@T$

End of Lists

11.3.7 Functions

Renaming — if v and w do not appear in D and w does not appear in b
$\langle v: D\cdot b\rangle = \langle w: D\cdot \langle v: D\cdot b\rangle\, w\rangle$

Function Composition — if $\neg\, f\!: \Box g$
$\Box(g\,f) = \S x: \Box f\cdot f\,x: \Box g$
$(g\,f)\;x = g\;(f\,x)$

Domain
$\Box\langle v: D\cdot b\rangle = D$

Application — if element $x: D$
$\langle v: D\cdot b\rangle\; x =$ (substitute x for v in b)

Distributive
$f\,null = null$
$f\,(A,B) = f\,A, f\,B$
$f\,(\S g) = \S y: f\,(\Box g)\cdot \exists x: \Box g\cdot f\,x = y \;\wedge\; g\,x$
$f\; \textbf{if } b \textbf{ then } x \textbf{ else } y \textbf{ fi} = \textbf{if } b \textbf{ then } f\,x \textbf{ else } f\,y \textbf{ fi}$
$\textbf{if } b \textbf{ then } f \textbf{ else } g \textbf{ fi}\; x = \textbf{if } b \textbf{ then } f\,x \textbf{ else } g\,x \textbf{ fi}$

Function Union
$\Box(f,\,g) = \Box f \text{ ‘ } \Box g$
$(f,\,g)\;x = f\,x,\, g\,x$

Function Intersection
$\Box(f \text{ ‘} g) = \Box f,\, \Box g$
$(f \text{ ‘} g)x = (f \mid g)x \text{ ‘ } (g \mid f)x$

Selective Union
$\Box(f \mid g) = \Box f,\, \Box g$
$(f \mid g)x = \textbf{if } x: \Box f \textbf{ then } f\,x \textbf{ else } g\,x \textbf{ fi}$
$f \mid f = f$
$f \mid (g \mid h) = (f \mid g) \mid h$
$(g \mid h)\,f = g\,f \mid h\,f$

Function Inclusion and Equality
$f\!: g \;=\; \Box f\!:: \Box g \;\wedge\; \forall x: \Box g\cdot f\,x: g\,x$
$f = g \;=\; \Box f = \Box g \;\wedge\; \forall x: \Box f\cdot f\,x = g\,x$

Arrow

$f\colon null \to A$
$A \to B\colon\ (A‘C) \to (B, D)$
$(A, B) \to C\ =\ A \to C \mid B \to C$
$f\colon A \to B\ =\ \Box f\colon\colon A\ \wedge\ \forall a\colon A \cdot f\, a\colon B$

Size

$\#f\ =\ ¢\Box f$

Extension

$f\ =\ \langle v\colon \Box f \cdot f\, v \rangle$

End of Functions

11.3.8 Quantifiers

Let x be an element, let a, b and c be binary, let n and m be numeric, let f and g be functions, and let p be a predicate.

$\forall v\colon null \cdot b\ =\ \top$
$\forall v\colon x \cdot b\ =\ \langle v\colon x \cdot b \rangle\, x$
$\forall v\colon A, B \cdot b\ =\ (\forall v\colon A \cdot b) \wedge (\forall v\colon B \cdot b)$
$\forall v\colon (\S v\colon D \cdot b) \cdot c\ =\ \forall v\colon D \cdot b \Rightarrow c$

$\exists v\colon null \cdot b\ =\ \bot$
$\exists v\colon x \cdot b\ =\ \langle v\colon x \cdot b \rangle\, x$
$\exists v\colon A, B \cdot b\ =\ (\exists v\colon A \cdot b) \vee (\exists v\colon B \cdot b)$
$\exists v\colon (\S v\colon D \cdot b) \cdot c\ =\ \exists v\colon D \cdot b \wedge c$

$\Sigma v\colon null \cdot n\ =\ 0$
$\Sigma v\colon x \cdot n\ =\ \langle v\colon x \cdot n \rangle\, x$
$(\Sigma v\colon A, B \cdot n) + (\Sigma v\colon A‘B \cdot n)\ =\ (\Sigma v\colon A \cdot n) + (\Sigma v\colon B \cdot n)$
$\Sigma v\colon (\S v\colon D \cdot b) \cdot n\ =\ \Sigma v\colon D \cdot \textbf{if}\ b\ \textbf{then}\ n\ \textbf{else}\ 0\ \textbf{fi}$

$\Pi v\colon null \cdot n\ =\ 1$
$\Pi v\colon x \cdot n\ =\ \langle v\colon x \cdot n \rangle\, x$
$(\Pi v\colon A, B \cdot n) \times (\Pi v\colon A‘B \cdot n)\ =\ (\Pi v\colon A \cdot n) \times (\Pi v\colon B \cdot n)$
$\Pi v\colon (\S v\colon D \cdot b) \cdot n\ =\ \Pi v\colon D \cdot \textbf{if}\ b\ \textbf{then}\ n\ \textbf{else}\ 1\ \textbf{fi}$

$\Downarrow v\colon null \cdot n\ =\ \infty$
$\Downarrow v\colon x \cdot n\ =\ \langle v\colon x \cdot n \rangle\, x$
$\Downarrow v\colon A, B \cdot n\ =\ (\Downarrow v\colon A \cdot n) \downarrow (\Downarrow v\colon B \cdot n)$
$\Downarrow v\colon (\S v\colon D \cdot b) \cdot n\ =\ \Downarrow v\colon D \cdot \textbf{if}\ b\ \textbf{then}\ n\ \textbf{else}\ \infty\ \textbf{fi}$

$\Uparrow v\colon null \cdot n\ =\ -\infty$
$\Uparrow v\colon x \cdot n\ =\ \langle v\colon x \cdot n \rangle\, x$
$\Uparrow v\colon A, B \cdot n\ =\ (\Uparrow v\colon A \cdot n) \uparrow (\Uparrow v\colon B \cdot n)$
$\Uparrow v\colon (\S v\colon D \cdot b) \cdot n\ =\ \Uparrow v\colon D \cdot \textbf{if}\ b\ \textbf{then}\ n\ \textbf{else}\ -\infty\ \textbf{fi}$

$\S v\colon null \cdot b\ =\ null$
$\S v\colon x \cdot b\ =\ \textbf{if}\ \langle v\colon x \cdot b \rangle\, x\ \textbf{then}\ x\ \textbf{else}\ null\ \textbf{fi}$
$\S v\colon A, B \cdot b\ =\ (\S v\colon A \cdot b), (\S v\colon B \cdot b)$
$\S v\colon A‘B \cdot b\ =\ (\S v\colon A \cdot b)\ ‘\ (\S v\colon B \cdot b)$
$\S v\colon (\S v\colon D \cdot b) \cdot c\ =\ \S v\colon D \cdot b \wedge c$

Inclusion

$A\colon B\ =\ \forall x\colon A \cdot x\colon B$

Cardinality

$¢A\ =\ \Sigma\,(A \to 1)$

Change of Variable — if d does not appear in b

$\forall r\colon f D \cdot b\ =\ \forall d\colon D \cdot \langle r\colon f D \cdot b \rangle\, (f\, d)$
$\exists r\colon f D \cdot b\ =\ \exists d\colon D \cdot \langle r\colon f D \cdot b \rangle\, (f\, d)$
$\Downarrow r\colon f D \cdot n\ =\ \Downarrow d\colon D \cdot \langle r\colon f D \cdot n \rangle\, (f\, d)$
$\Uparrow r\colon f D \cdot n\ =\ \Uparrow d\colon D \cdot \langle r\colon f D \cdot n \rangle\, (f\, d)$

Identity

$\forall v \cdot \top$
$\neg \exists v \cdot \bot$

Specialize and Generalize — if element $x\colon \Box f$

$\Downarrow f\ \le\ f\, x\ \le\ \Uparrow f$

Bunch-Element Conversion

$A\colon B\ =\ \forall a\colon A \cdot \exists b\colon B \cdot a{=}b$
$f A\colon g B\ =\ \forall a\colon A \cdot \exists b\colon B \cdot f\, a = g\, b$

Idempotent — if $D \ne null$ and v does not appear in b

$\forall v\colon D \cdot b\ =\ b$
$\exists v\colon D \cdot b\ =\ b$

Distributive — if $D \ne null$ and v does not appear in a

$a \wedge \forall v\colon D \cdot b\ =\ \forall v\colon D \cdot a \wedge b$
$a \wedge \exists v\colon D \cdot b\ =\ \exists v\colon D \cdot a \wedge b$
$a \vee \forall v\colon D \cdot b\ =\ \forall v\colon D \cdot a \vee b$
$a \vee \exists v\colon D \cdot b\ =\ \exists v\colon D \cdot a \vee b$
$a \Rightarrow \forall v\colon D \cdot b\ =\ \forall v\colon D \cdot a \Rightarrow b$
$a \Rightarrow \exists v\colon D \cdot b\ =\ \exists v\colon D \cdot a \Rightarrow b$

Absorption — if $x: D$

$\langle v: D\cdot b\rangle\, x \land \exists v: D\cdot b \;=\; \langle v: D\cdot b\rangle\, x$
$\langle v: D\cdot b\rangle\, x \lor \forall v: D\cdot b \;=\; \langle v: D\cdot b\rangle\, x$
$\langle v: D\cdot b\rangle\, x \land \forall v: D\cdot b \;=\; \forall v: D\cdot b$
$\langle v: D\cdot b\rangle\, x \lor \exists v: D\cdot b \;=\; \exists v: D\cdot b$

Specialization — if element $x: \Box p$

$\forall p \;\Rightarrow\; p\, x$

One-Point — if $x: D$
and v does not appear in x

$\forall v: D\cdot v{=}x \Rightarrow b \;=\; \langle v: D\cdot b\rangle\, x$
$\exists v: D\cdot v{=}x \land b \;=\; \langle v: D\cdot b\rangle\, x$

Duality

$\neg \forall v\cdot b \;=\; \exists v\cdot \neg b$
$\neg \exists v\cdot b \;=\; \forall v\cdot \neg b$
$-\Uparrow v\cdot n \;=\; \Downarrow v\cdot -n$
$-\Downarrow v\cdot n \;=\; \Uparrow v\cdot -n$

Solution

$\S v: D\cdot \top \;=\; D$
$(\S v: D\cdot b):\; D$
$\S v: D\cdot \bot \;=\; null$
$(\S v\cdot b): (\S v\cdot c) \;=\; \forall v\cdot b{\Rightarrow}c$
$(\S v\cdot b), (\S v\cdot c) \;=\; \S v\cdot b \lor c$
$(\S v\cdot b) \,‘\, (\S v\cdot c) \;=\; \S v\cdot b \land c$
$x: \S p \;=\; x: \Box p \;\land\; p\, x$
$\forall f \;=\; (\S f){=}(\Box f)$
$\exists f \;=\; (\S f){\ne}null$

Bounding — if $D{\ne}null$
and v does not appear in n

$n > (\Uparrow v: D\cdot m) \;\Rightarrow\; (\forall v: D\cdot n{>}m)$
$n < (\Downarrow v: D\cdot m) \;\Rightarrow\; (\forall v: D\cdot n{<}m)$
$n \ge (\Uparrow v: D\cdot m) \;=\; (\forall v: D\cdot n{\ge}m)$
$n \le (\Downarrow v: D\cdot m) \;=\; (\forall v: D\cdot n{\le}m)$
$n \ge (\Downarrow v: D\cdot m) \;\Leftarrow\; (\exists v: D\cdot n{\ge}m)$
$n \le (\Uparrow v: D\cdot m) \;\Leftarrow\; (\exists v: D\cdot n{\le}m)$
$n > (\Downarrow v: D\cdot m) \;=\; (\exists v: D\cdot n{>}m)$
$n < (\Uparrow v: D\cdot m) \;=\; (\exists v: D\cdot n{<}m)$

Antidistributive — if $D{\ne}null$
and v does not appear in a

$a \Leftarrow \exists v: D\cdot b \;=\; \forall v: D\cdot a \Leftarrow b$
$a \Leftarrow \forall v: D\cdot b \;=\; \exists v: D\cdot a \Leftarrow b$

Generalization — if element $x: \Box p$

$p\, x \;\Rightarrow\; \exists p$

Splitting — for any fixed domain

$\forall v\cdot a \land b \;=\; (\forall v\cdot a) \;\land\; (\forall v\cdot b)$
$\exists v\cdot a \land b \;\Longrightarrow\; (\exists v\cdot a) \;\land\; (\exists v\cdot b)$
$\forall v\cdot a \lor b \;\Longleftarrow\; (\forall v\cdot a) \;\lor\; (\forall v\cdot b)$
$\exists v\cdot a \lor b \;=\; (\exists v\cdot a) \;\lor\; (\exists v\cdot b)$
$\forall v\cdot a \Rightarrow b \;\Longrightarrow\; (\forall v\cdot a) \;\Rightarrow\; (\forall v\cdot b)$
$\forall v\cdot a \Rightarrow b \;\Longrightarrow\; (\exists v\cdot a) \;\Rightarrow\; (\exists v\cdot b)$
$\forall v\cdot a = b \;\Longrightarrow\; (\forall v\cdot a) \;=\; (\forall v\cdot b)$
$\forall v\cdot a = b \;\Longrightarrow\; (\exists v\cdot a) \;=\; (\exists v\cdot b)$

Commutative

$\forall v\cdot \forall w\cdot b \;=\; \forall w\cdot \forall v\cdot b$
$\exists v\cdot \exists w\cdot b \;=\; \exists w\cdot \exists v\cdot b$

Semicommutative

$\exists v\cdot \forall w\cdot b \;\Longrightarrow\; \forall w\cdot \exists v\cdot b$
$\forall x\cdot \exists y\cdot p\, x\, y \;=\; \exists f\cdot \forall x\cdot p\, x\, (f\, x)$

Domain Change

$A: B \;\Longrightarrow\; (\forall v: A\cdot b) \;\Leftarrow\; (\forall v: B\cdot b)$
$A: B \;\Longrightarrow\; (\exists v: A\cdot b) \;\Rightarrow\; (\exists v: B\cdot b)$
$\forall v: A\cdot v: B \Rightarrow p \;=\; \forall v: A‘B\cdot p$
$\exists v: A\cdot v: B \land p \;=\; \exists v: A‘B\cdot p$

Extreme

$(\Downarrow n: int\cdot n) = (\Downarrow n: real\cdot n) = -\infty$
$(\Uparrow n: int\cdot n) = (\Uparrow n: real\cdot n) = \infty$

Connection

$n{\le}m \;=\; \forall k\cdot\; k{\le}n \Rightarrow k{\le}m$
$n{\le}m \;=\; \forall k\cdot\; k{<}n \Rightarrow k{<}m$
$n{\le}m \;=\; \forall k\cdot\; m{\le}k \Rightarrow n{\le}k$
$n{\le}m \;=\; \forall k\cdot\; m{<}k \Rightarrow n{<}k$

Distributive — if $D{\ne}null$ and v does not appear in n

$n{\uparrow}(\Uparrow v: D\cdot m) \;=\; (\Uparrow v: D\cdot n{\uparrow}m)$
$n{\uparrow}(\Downarrow v: D\cdot m) \;=\; (\Downarrow v: D\cdot n{\uparrow}m)$
$n + (\Uparrow v: D\cdot m) \;=\; (\Uparrow v: D\cdot n{+}m)$
$n - (\Uparrow v: D\cdot m) \;=\; (\Downarrow v: D\cdot n{-}m)$
$(\Uparrow v: D\cdot m) - n \;=\; (\Uparrow v: D\cdot m{-}n)$
$n{\ge}0 \;\Rightarrow\; n{\times}(\Uparrow v: D\cdot m) = (\Uparrow v: D\cdot n{\times}m)$
$n{\le}0 \;\Rightarrow\; n{\times}(\Uparrow v: D\cdot m) = (\Downarrow v: D\cdot n{\times}m)$
$n{\times}(\Sigma v: D\cdot m) = (\Sigma v: D\cdot n{\times}m)$

$n{\downarrow}(\Downarrow v: D\cdot m) \;=\; (\Downarrow v: D\cdot n{\downarrow}m)$
$n{\downarrow}(\Uparrow v: D\cdot m) \;=\; (\Uparrow v: D\cdot n{\downarrow}m)$
$n + (\Downarrow v: D\cdot m) \;=\; (\Downarrow v: D\cdot n{+}m)$
$n - (\Downarrow v: D\cdot m) \;=\; (\Uparrow v: D\cdot n{-}m)$
$(\Downarrow v: D\cdot m) - n \;=\; (\Downarrow v: D\cdot m{-}n)$
$n{\ge}0 \;\Rightarrow\; n{\times}(\Downarrow v: D\cdot m) = (\Downarrow v: D\cdot n{\times}m)$
$n{\le}0 \;\Rightarrow\; n{\times}(\Downarrow v: D\cdot m) = (\Uparrow v: D\cdot n{\times}m)$
$(\Pi v: D\cdot m)^n = (\Pi v: D\cdot m^n)$

End of Quantifiers

11.3.9 Limits

$$(\Uparrow m\cdot \Downarrow n\cdot f\,(m+n)) \;\le\; \Updownarrow f \;\le\; (\Downarrow m\cdot \Uparrow n\cdot f\,(m+n))$$
$$\exists m\cdot \forall n\cdot p\,(m+n) \;\;\Rightarrow\;\; \Updownarrow p \;\;\Rightarrow\;\; \forall m\cdot \exists n\cdot p\,(m+n)$$
$$\Updownarrow n\cdot n \;=\; \infty$$

End of Limits

11.3.10 Specifications and Programs

For specifications P , Q , R , and S , and binary b ,

$ok \;=\; x'=x \;\land\; y'=y \;\land\; ...$

$x{:=}\, e \;=\; x'=e \;\land\; y'=y \;\land\; ...$

$P.\,Q \;=\; \exists x'', y'', ...\cdot\ \langle x', y', ...\cdot P\rangle\, x''\, y''\, ... \;\land\; \langle x, y, ...\cdot Q\rangle\, x''\, y''\, ...$

$P\|Q \;=\; \exists t_P, t_Q\cdot\ \langle t'\cdot P\rangle\, t_P \;\land\; \langle t'\cdot Q\rangle\, t_Q \;\land\; t' = t_P \uparrow t_Q$

$\textbf{if } b \textbf{ then } P \textbf{ else } Q \textbf{ fi} \;=\; b \land P \lor \lnot b \land Q \;=\; (b \Rightarrow P) \land (\lnot b \Rightarrow Q)$

$\textbf{new } x\colon T\cdot P \;=\; \exists x, x'\colon T\cdot P$

$\textbf{frame } x\cdot P \;=\; P \;\land\; y'=y \;\land\; ...$

$\textbf{while } b \textbf{ do } P \textbf{ od} \;=\; t'\ge t \land \textbf{if } b \textbf{ then } P.\;\; t{:=}\, t{+}1.\;\; \textbf{while } b \textbf{ do } P \textbf{ od else } ok \textbf{ fi}$

$\forall\sigma, \sigma'\cdot \textbf{if } b \textbf{ then } P.\;\; W \textbf{ else } ok \textbf{ fi} \Leftarrow W \;\;\Rightarrow\;\; \forall\sigma, \sigma'\cdot \textbf{while } b \textbf{ do } P \textbf{ od} \Leftarrow W$

To prove $F\, m \;\Leftarrow\; \textbf{for } i{:=}\, m;..n \textbf{ do } P \textbf{ od}$

prove $F\, i \;\Leftarrow\; i\colon m,..n \;\land\; (P.\;\; F(i{+}1))$

and $F\, n \;\Leftarrow\; ok$

$A\, m \Rightarrow A'n \;\Leftarrow\; \textbf{for } i{:=}\, m;..n \textbf{ do } i\colon m,..n \land A\, i \;\Rightarrow\; A'(i{+}1) \textbf{ od}$

$\textbf{wait until } w \;=\; t{:=}\, t\uparrow w$

$\textbf{assert } b \;=\; \textbf{if } b \textbf{ then } ok \textbf{ else } screen!\ \text{“error”}.\;\; \textbf{wait until } \infty \textbf{ fi}$

$\textbf{ensure } b \;=\; b \land ok$

$P.\;\; (P \textbf{ value } e){=}e$ but do not double-prime or substitute in $(P \textbf{ value } e)$

Data transformer D satisfies $\forall new\cdot \exists old\cdot D$ and transforms specification S to

$\forall old\cdot D \Rightarrow \exists old'\cdot D' \land S$

$c? \;=\; t{:=}\, t\uparrow(\mathcal{T}c_{\boldsymbol{rc}} + (\text{transit time})).\;\; \boldsymbol{rc}{:=}\, \boldsymbol{rc}+1$

$c \;=\; \mathcal{M}c_{\boldsymbol{rc}-1}$

$c!\ e \;=\; \mathcal{M}c_{\boldsymbol{wc}} = e \;\land\; \mathcal{T}c_{\boldsymbol{wc}} = t \;\land\; (\boldsymbol{wc}{:=}\, \boldsymbol{wc}+1)$

$\surd c \;=\; \mathcal{T}c_{\boldsymbol{rc}} + (\text{transit time}) \le t$

$\textbf{new } x\colon time{\rightarrow}T\cdot S \;=\; \exists x\colon time{\rightarrow}T\cdot S$

$\textbf{new } c?\ T\cdot S \;=\; \exists \mathcal{M}c\colon \infty{*}T\cdot \exists \mathcal{T}c\colon \infty{*}xnat\cdot \exists \boldsymbol{rc}, \boldsymbol{rc}', \boldsymbol{wc}, \boldsymbol{wc}'\colon xnat\cdot \boldsymbol{rc}{=}\boldsymbol{wc}{=}0 \land S$

$P.\,ok \;=\; P \;=\; ok.\,P$	identity
$P.\,(Q.\,R) \;=\; (P.\,Q).\,R$	associativity
$P{\lor}Q.\,R{\lor}S \;=\; (P.\,R) \lor (P.\,S) \lor (Q.\,R) \lor (Q.\,S)$	distributivity
$\textbf{if } b \textbf{ then } P \textbf{ else } Q \textbf{ fi}.\,R \;=\; \textbf{if } b \textbf{ then } P.\,R \textbf{ else } Q.\,R \textbf{ fi}$	distributivity (unprimed b)
$P.\;\; \textbf{if } b \textbf{ then } Q \textbf{ else } R \textbf{ fi} \;=\; \textbf{if } P.\,b \textbf{ then } P.\,Q \textbf{ else } P.\,R \textbf{ fi}$	distributivity (unprimed b)
$P \| Q \;=\; Q \| P$	symmetry
$P \| (Q \| R) \;=\; (P \| Q) \| R$	associativity
$P \| Q{\lor}R \;=\; (P \| Q) \lor (P \| R)$	distributivity
$P \| \textbf{if } b \textbf{ then } Q \textbf{ else } R \textbf{ fi} \;=\; \textbf{if } b \textbf{ then } P \| Q \textbf{ else } P \| R \textbf{ fi}$	distributivity
$\textbf{if } b \textbf{ then } P\|Q \textbf{ else } R\|S \textbf{ fi} \;=\; \textbf{if } b \textbf{ then } P \textbf{ else } R \textbf{ fi} \| \textbf{if } b \textbf{ then } Q \textbf{ else } S \textbf{ fi}$	distributivity
$x{:=}\, \textbf{if } b \textbf{ then } e \textbf{ else } f \textbf{ fi} \;=\; \textbf{if } b \textbf{ then } x{:=}\, e \textbf{ else } x{:=}\, f \textbf{ fi}$	functional-imperative

End of Specifications and Programs

11.3.11 Substitution

Let x and y be different boundary state variables, let e and f be expressions of the prestate, and let S be a specification.

$x{:=}\,e.\,S \;=\;$ (for x substitute e in S)

$(x{:=}\,e \,\|\, y{:=}\,f).\; S \;=\;$ (for x substitute e and concurrently for y substitute f in S)

End of Substitution

11.3.12 Assertions

Let P and Q be specifications. Let A be an assertion and let A' be the same as A but with primes on all the variables.

$A \wedge (P.\,Q) \;\Leftarrow\; A{\wedge}P.\,Q$

$A \Rightarrow (P.Q) \;\Leftarrow\; A{\Rightarrow}P.\,Q$

$(P.Q) \wedge A' \;\Leftarrow\; P.\,Q{\wedge}A'$

$(P.Q) \Leftarrow A' \;\Leftarrow\; P.\,Q{\Leftarrow}A'$

$P.\,A{\wedge}Q \;\Leftarrow\; P{\wedge}A'.\,Q$

$P.\,Q \;\Leftarrow\; P{\wedge}A'.\; A{\Rightarrow}Q$

A is a sufficient precondition for P to be refined by S
if and only if $A{\Rightarrow}P$ is refined by S .

A' is a sufficient postcondition for P to be refined by S
if and only if $A'{\Rightarrow}P$ is refined by S .

End of Assertions

11.3.13 Refinement

Refinement by Steps (Stepwise Refinement) (monotonicity, transitivity)

If $A \Leftarrow$ **if** b **then** C **else** D **fi** and $C \Leftarrow E$ and $D \Leftarrow F$ are theorems,
then $A \Leftarrow$ **if** b **then** E **else** F **fi** is a theorem.

If $A \Leftarrow B.C$ and $B \Leftarrow D$ and $C \Leftarrow E$ are theorems, then $A \Leftarrow D.E$ is a theorem.

If $A \Leftarrow B\|C$ and $B \Leftarrow D$ and $C \Leftarrow E$ are theorems, then $A \Leftarrow D\|E$ is a theorem.

If $A \Leftarrow B$ and $B \Leftarrow C$ are theorems, then $A \Leftarrow C$ is a theorem.

Refinement by Parts (monotonicity, conflation)

If $A \Leftarrow$ **if** b **then** C **else** D **fi** and $E \Leftarrow$ **if** b **then** F **else** G **fi** are theorems,
then $A{\wedge}E \Leftarrow$ **if** b **then** $C{\wedge}F$ **else** $D{\wedge}G$ **fi** is a theorem.

If $A \Leftarrow B.C$ and $D \Leftarrow E.F$ are theorems, then $A{\wedge}D \Leftarrow B{\wedge}E.\,C{\wedge}F$ is a theorem.

If $A \Leftarrow B\|C$ and $D \Leftarrow E\|F$ are theorems, then $A{\wedge}D \Leftarrow B{\wedge}E \,\|\, C{\wedge}F$ is a theorem.

If $A \Leftarrow B$ and $C \Leftarrow D$ are theorems, then $A{\wedge}C \Leftarrow B{\wedge}D$ is a theorem.

Refinement by Cases

$P \Leftarrow$ **if** b **then** Q **else** R **fi** is a theorem if and only if
$P \Leftarrow b \wedge Q$ and $P \Leftarrow \neg b \wedge R$ are theorems.

End of Refinement

End of Laws

11.4 Names

abs: *xreal*→(§*r*: *xreal*· *r*≥0)	*abs r* = **if** *r*≥0 **then** *r* **else** –*r* **fi**
bin (the binary values)	*bin* = ⊤, ⊥
ceil: *real*→*int*	*r* ≤ *ceil r* < *r*+1
char (the characters)	*char* = ..., “a”, “A”, ...
div: *real*→(§*r*: *real*· *r*>0)→*int*	*div x y* = *floor* (*x*/*y*)
divides: (*nat*+1)→*int*→*bin*	*divides n i* = *i*/*n*: *int*
entro: *prob*→(§*r*: *xreal*· *r*≥0)	*entro p* = *p* × *info p* + (1–*p*) × *info* (1–*p*)
even: *int*→*bin*	*even i* = *i*/2: *int*
	even = *divides* 2
floor: *real*→*int*	*floor r* ≤ *r* < *floor r* + 1
info: *prob*→(§*r*: *xreal*· *r*≥0)	*info p* = – *log p*
int (the integers)	*int* = *nat*, –*nat*
log: (§*r*: *xreal*· *r*≥0)→*xreal*	*log* (2^x) = *x*
	log (*x*×*y*) = *log x* + *log y*
mod: *real*→(§*r*: *real*· *r*>0)→*real*	0 ≤ *mod a d* < *d*
	a = *div a d* × *d* + *mod a d*
nat (the naturals)	0, *nat*+1: *nat*
	0, *B*+1: *B* ⇒ *nat*: *B*
nil (the empty string)	↔*nil* = 0
	nil; *S* = *S* = *S*; *nil*
	nil ≤ *S*
null (the empty bunch)	¢*null* = 0
	null, *A* = *A* = *A*, *null*
	null: *A*
odd: *int*→*bin*	*odd i* = ¬ *i*/2: *int*
	odd = ¬*even*
ok (the empty program)	*ok* = σ′=σ
	ok.*P* = *P* = *P*.*ok*
prob (probability)	*prob* = §*r*: *real*· 0≤*r*≤1
rand (random number)	*rand n*: 0,..*n*
rat (the rationals)	*rat* = *int*/(*nat*+1)
real (the reals)	*real* = §*r*: *xreal*· –∞ < *r* <∞
suc: *nat*→(*nat*+1)	*suc n* = *n*+1
time (time)	either *time*=*xnat* or *time*=§*r*: *xreal*· *r*≥0
xint (the extended integers)	*xint* = –∞, *int*, ∞
xnat (the extended naturals)	*xnat* = *nat*, ∞
xrat (the extended rationals)	*xrat* = –∞, *rat*, ∞
xreal (the extended reals)	*x*: *xreal* = ∃*f*: *nat*→*rat*· *x* = ⇳*f*

End of Names

11.5 Symbols

symbol	page	pronunciation	symbol	page	pronunciation
⊤	3	top, true	√	136	input check
⊥	3	bottom, false	( )	4	precedence brackets
¬	3	not	{ }	17	set brackets
∧	3	and	[ ]	20	list brackets
∨	3	or	⟨ ⟩	23	function (scope) brackets
⇒	3	implies	ϟ	17	power
⟹	4	implies	¢	14	bunch size, cardinality
⇐	3	follows from, is implied by	$	17	set size, cardinality
⟸	4	follows from, is implied by	↔	17	string size (length)
=	3	equals, if and only if	#	20,23	list size (length), function size
=	4	equals, if and only if	$\vert$	20,24	otherwise, selective union
≠	3	differs from, is unequal to	$\Vert$	121	concurrent (parallel) composition
<	13	less than	~	17,20	contents of a set or list
>	13	greater than	*	18	repetition of a string
≤	13	less than or equal to	□	20,23	domain of a list or function
≥	13	greater than or equal to	→	25	function arrow
+	12	plus	∈	17	element of a set
–	12	minus	⊆	17	subset
×	12	times, multiplication	∪	17	set union
/	12	divided by	∩	17	set intersection
↑	12	maximum	@	22	index with a pointer
↓	12	minimum	∀	26	for all, universal quantifier
,	14	bunch union	∃	26	there exists, existential quantifier
,..	16	union from (including) to (excluding)	Σ	26	sum of, summation quantifier
‘	14	bunch intersection	Π	26	product of, product quantifier
;	17	string join	⇑	26	maximum (lub) quantifier
;;	20	list join	⇓	26	minimum (glb) quantifier
;..	19	join from (including) to (excluding)	⇕	33	limit quantifier
:	14	is in, are in, bunch inclusion	§	28	those, solution quantifier
::	14	includes	′	34	x' is final value of state variable x
:=	36	assignment	“ ”	13,19	“hi” is a text or string of characters
⦂	78	label, target of **go to**	a^b	12	exponentiation
.	36	sequential composition	a_b	18	string indexing
·	23	function and quantifier	$a\ b$	20,24,31	indexing, application, composition
!	136	output	⊲ ⊳	18	string modification
?	136	input	∞	12	infinity

assert	79	**if then else fi**	4
do od	73	**new**	68
ensure	80	**or**	80
exit when	73	**value**	81
for do od	76	**wait until**	79
frame	69	**while do od**	71
go to	78		

11.6 Precedence

0 ⊤ ⊥ ( ) { } [ ] ⟨ ⟩ **if fi** **do od** number text name superscript subscript
1 @ adjacency
2 prefix– ¢ $ ↔ # * ~ ↯ □ → √ ∀ ∃ Σ Π ⇑ ⇓ ⇕ §
3 × / ∩ ↑ ↓
4 + infix– ∪
5 ; ;.. ;; ‘
6 , ,.. | ⊲⊳
7 = ⧧ < > ≤ ≥ : :: ∈ ⊆
8 ¬
9 ∧
10 ∨
11 ⇒ ⇐
12 := ! ?
13 **exit when** **go to** **wait until** **assert** **ensure** **or**
14 . || **value**
15 ∀· ∃· Σ· Π· ⇑· ⇓· ⇕· §· **new**· **frame**·
16 **=** **⇒** **⇐**

Superscripting and subscripting associate from right to left, and bracket what is in them.

Adjacency associates from left to right, so $a\ b\ c$ means the same as $(a\ b)\ c$. The infix operators @ / – associate from left to right. The infix operators * → associate from right to left. The infix operators × ∩ ↑ ↓ + ∪ ; ;; ‘ , | ∧ ∨ . || are associative (they associate in both directions).

Quantifiers as prefix operators are on level 2, but in the abbreviated quantifier notation they are on level 15.

On levels 7, 11, and 16 the operators are continuing. For example, $a=b=c$ neither associates to the left nor associates to the right, but means the same as $a=b \land b=c$. On any one of these levels, a mixture of continuing operators can be used. For example, $a\le b<c$ means the same as $a\le b \land b<c$.

The operators **=** **⇒** **⇐** are identical to = ⇒ ⇐ except for precedence.

End of Precedence

11.7 Distribution

The operators in the following expressions distribute over bunch union in any operand:

$\lnot A$ $A\land B$ $A\lor B$ $A\Rightarrow B$ $A\Leftarrow B$ $-A$ $A+B$ $A-B$ $A\times B$ A/B A^B $A\uparrow B$ $A\downarrow B$
A,B $A`B$ $\$A$ $A\cup B$ $A\cap B$ $\sim A$ $A;B$ $\leftrightarrow A$ A_B $[A]$ $A;;B$ $A\ B$ $\#A$ $A@B$

The operator in $A*B$ distributes over bunch union in its left operand only.

End of Distribution

End of Reference

End of a Practical Theory of Programming