Reference Card

Jean-Raymond Abrial and Thai Son Hoang

April 2008

Contents

Basic Inference Rules of Mathematical Reasoning

$$\frac{}{H,\ P \vdash P}\ \textbf{HYP}$$

$$\frac{H \vdash Q}{H,\ P \vdash Q}\ \textbf{MON}$$

$$\frac{H \vdash P \qquad H,\ P \vdash Q}{H \vdash Q}\ \textbf{CUT}$$

Propositional Calculus Rules of Inference (1)

- Rules about conjunction

$$\frac{H, P, Q \vdash R}{H,\ P \wedge Q \vdash R}\ \textbf{AND\_L} \qquad \frac{H \vdash P \qquad H \vdash Q}{H \vdash P \wedge Q}\ \textbf{AND\_R}$$

- Rules about implication

$$\frac{H, P, Q \vdash R}{H,\ P,\ P \Rightarrow Q \vdash R}\ \textbf{IMP\_L} \qquad \frac{H, P \vdash Q}{H \vdash P \Rightarrow Q}\ \textbf{IMP\_R}$$

Note: Rules with a double horizontal line can be applied in both directions

Propositional Calculus Rules of Inference (2)

- Rules about negation

$$\frac{}{P,\ \neg P \ \vdash\ Q}\ \textbf{NOT\_L} \qquad \frac{}{\bot \ \vdash\ P}\ \textbf{CNTR}$$

$$\frac{H, P \ \vdash\ Q \qquad H, P \ \vdash\ \neg Q}{H \ \vdash\ \neg P}\ \textbf{NOT\_R}$$

$$\frac{H, \neg P \ \vdash\ Q \qquad H, \neg P \ \vdash\ \neg Q}{H \ \vdash\ P}\ \textbf{NOT\_R}$$

Propositional Calculus Rules of Inference (3)

- Rules about disjunction

$$\frac{H, P \ \vdash\ R \qquad H, Q \ \vdash\ R}{H,\ P \vee Q \ \vdash\ R}\ \textbf{OR\_L}$$

$$\frac{H \ \vdash\ P}{H \ \vdash\ P \vee Q}\ \textbf{OR\_R1} \qquad \frac{H \ \vdash\ Q}{H \ \vdash\ P \vee Q}\ \textbf{OR\_R2}$$

- Transforming a disjunctive goal

$$\frac{H, \neg P \ \vdash\ Q}{H \ \vdash\ P \vee Q}\ \textbf{NEG}$$

Predicate Calculus Rules of Inference

$$\frac{H,\ \forall x \cdot P(x),\ P(E) \ \vdash\ Q}{H,\ \forall x \cdot P(x) \ \vdash\ Q}\ \textbf{ALL\_L} \qquad \frac{H \ \vdash\ P(x)}{H \ \vdash\ \forall x \cdot P(x)}\ \textbf{ALL\_R}$$

$$\frac{H,\ P(x) \ \vdash\ Q}{H,\ \exists x \cdot P(x) \ \vdash\ Q}\ \textbf{XST\_L} \qquad \frac{H \ \vdash\ P(E)}{H \ \vdash\ \exists x \cdot P(x)}\ \textbf{XST\_R}$$

- In rule **ALL_L** and **XST_R**, **E** is an expression

- In rule **ALL_R** and **XST_L**, variable **x** is not free in **H**.

Equality Rules of Inference

$$\frac{H(F),\ E = F \ \vdash\ P(F)}{H(E),\ E = F \ \vdash\ P(E)}\ \textbf{EQ\_LR} \qquad \frac{H(E),\ E = F \ \vdash\ P(E)}{H(F),\ E = F \ \vdash\ P(F)}\ \textbf{EQ\_RL}$$

$$\frac{}{\vdash\ E = E}\ \textbf{EQL}$$

$$\frac{H \ \vdash\ E = G \ \wedge\ F = I}{H \ \vdash\ E \mapsto F \ =\ G \mapsto I}\ \textbf{PAIR}$$

Basic Set Operator Memberships (Axioms)

These axioms are defined by equivalences.

Left Part	Right Part
$E \mapsto F \in S \times T$	$E \in S \ \wedge \ F \in T$
$S \in \mathbb{P}(T)$	$\forall x \cdot x \in S \Rightarrow x \in T$
$E \in \{x \cdot x \in S \ \wedge \ P(x) \mid F(x)\}$	$\exists x \cdot x \in S \ \wedge \ P(x) \ \wedge \ E = F(x)$
$E \in \{x \mid x \in S \ \wedge \ P(x)\}$	$E \in S \ \wedge \ P(E)$

Set Inclusion and Extensionality Axiom

Left Part	Right Part
$S \subseteq T$	$S \in \mathbb{P}(T)$
$S = T$	$S \subseteq T \ \wedge \ T \subseteq S$

The first rule is just a syntactic extension

The second rule is the Extensionality Axiom

Elementary Set Operator Memberships

$E \in S \cup T$	$E \in S \ \vee \ E \in T$
$E \in S \cap T$	$E \in S \ \wedge \ E \in T$
$E \in S \setminus T$	$E \in S \ \wedge \ E \notin T$
$E \in \{a, \ldots, b\}$	$E = a \ \vee \ \ldots \ \vee \ E = b$
$E \in \varnothing$	$\bot$

Generalizations of Elementary Operator Memberships

$E \in \mathrm{union}\,(S)$	$\exists s \cdot s \in S \ \wedge \ E \in s$
$E \in \bigcup x \cdot x \in S \ \wedge \ P(x) \mid T(x)$	$\exists x \cdot x \in S \ \wedge \ P(x) \ \wedge \ E \in T(x)$
$E \in \mathrm{inter}\,(S)$	$\forall s \cdot s \in S \Rightarrow E \in s$
$E \in \bigcap x \cdot x \in S \ \wedge \ P(x) \mid T(x)$	$\forall x \cdot x \in S \ \wedge \ P(x) \Rightarrow E \in T(x)$

Well-definedness condition for case 3: $S \neq \varnothing$

Well-definedness condition for case 4: $\exists x \cdot x \in S \ \wedge P(x)$

Binary Relation Operator Memberships (1)

Left Part	Right Part
$r \in S \leftrightarrow T$	$r \subseteq S \times T$
$E \in \mathbf{dom}\,(r)$	$\exists y \cdot E \mapsto y \in r$
$F \in \mathbf{ran}\,(r)$	$\exists x \cdot x \mapsto F \in r$
$E \mapsto F \in r^{-1}$	$F \mapsto E \in r$

Binary Relation Operator Memberships (2)

Left Part	Right Part
$r \in S \leftrightarrow\!\!\!\!\rightarrow T$	$r \in S \leftrightarrow T \ \wedge\ \mathrm{ran}(r) = T$
$r \in S \leftarrow\!\!\!\!\leftrightarrow T$	$r \in S \leftrightarrow T \ \wedge\ \mathrm{dom}(r) = T$
$r \in S \leftarrow\!\!\!\!\leftrightarrow\!\!\!\!\rightarrow T$	$r \in S \leftrightarrow\!\!\!\!\rightarrow T \ \wedge\ r \in S \leftarrow\!\!\!\!\leftrightarrow T$

Binary Relation Operator Memberships (3)

Left Part	Right Part
$E \mapsto F \in S \lhd r$	$E \in S \ \wedge\ E \mapsto F \in r$
$E \mapsto F \in r \rhd T$	$E \mapsto F \in r \ \wedge\ F \in T$
$E \mapsto F \in S \lhd\!\!\!- r$	$E \notin S \ \wedge\ E \mapsto F \in r$
$E \mapsto F \in r \rhd\!\!\!- T$	$E \mapsto F \in r \ \wedge\ F \notin T$

Binary Relation Operator Memberships (4)

$F \in r[w]$	$\exists x \cdot x \in w \ \wedge\ x \mapsto F \in r$
$E \mapsto F \in (p\,;q)$	$\exists x \cdot E \mapsto x \in p \ \wedge\ x \mapsto F \in q$
$p \lhd\!\!\!- q$	$(\mathrm{dom}\,(q) \lhd p) \ \cup\ q$
$E \mapsto F \in \mathbf{id}\,(S)$	$E \in S \ \wedge\ F = E$

Binary Relation Operator Memberships (5)

$E \mapsto (F \mapsto G) \in p \otimes q$	$E \mapsto F \in p \ \wedge \ E \mapsto G \in q$
$(E \mapsto F) \mapsto G \in \mathbf{prj_1}(S, T)$	$E \in S \ \wedge \ F \in T \ \wedge \ G = E$
$(E \mapsto F) \mapsto G \in \mathbf{prj_2}(S, T)$	$E \in S \ \wedge \ F \in T \ \wedge \ G = F$
$(E \mapsto G) \mapsto (F \mapsto H) \in p \parallel q$	$E \mapsto F \in p \ \wedge \ G \mapsto H \in q$

Some Useful Modelling Concepts

Given a relation r such that $r \in S \leftrightarrow S$

$r = r^{-1}$	r is symmetric
$r \cap r^{-1} = \varnothing$	r is asymmetric
$r \cap r^{-1} \subseteq \mathrm{id}(S)$	r is antisymmetric
$\mathrm{id}(S) \subseteq r$	r is reflexive
$r \cap \mathrm{id}(S) = \varnothing$	r is irreflexive
$r; r \subseteq r$	r is transitive

Function Operator Memberships (1)

Left Part	Right Part
$f \in S \nrightarrow T$	$f \in S \leftrightarrow T \ \wedge \ (f^{-1}; f) = \mathrm{id}(\mathrm{ran}(f))$
$f \in S \rightarrow T$	$f \in S \nrightarrow T \ \wedge \ S = \mathrm{dom}(f)$
$f \in S \rightarrowtail T$	$f \in S \nrightarrow T \ \wedge \ f^{-1} \in T \nrightarrow S$
$f \in S \rightarrowtail T$	$f \in S \rightarrow T \ \wedge \ f^{-1} \in T \nrightarrow S$

Function Operator Memberships (2)

Left Part	Right Part
$f \in S \twoheadrightarrow T$	$f \in S \nrightarrow T \ \wedge \ T = \mathrm{ran}(f)$
$f \in S \twoheadrightarrow T$	$f \in S \rightarrow T \ \wedge \ T = \mathrm{ran}(f)$
$f \in S \rightarrowtail T$	$f \in S \rightarrowtail T \ \wedge \ f \in S \twoheadrightarrow T$

Applying a Function

Given a partial function f, we have

Left Part	Right Part
$F = f(E)$	$E \mapsto F \in f$

Well-definedness conditions: f is a partial function

Context Structure

context
$< context\_identifier >$
extends $\star$
$< context\_identifier >$
$\ldots$
sets $\star$
$< set\_identifier >$
$\ldots$
constants $\star$
$< constant\_identifier >$
$\ldots$
axioms $\star$
$< label >: < predicate >$
$\ldots$
theorems $\star$
$< label >: < predicate >$
$\ldots$
end

- Sections with "$\star$" might be empty
- All keyword sections are predefined in the Rodin Platform
- All labels are generated automatically by the Rodin Platform (but can be modified)

Machine Structure

machine
$< machine\_identifier >$
refines $\star$
$< machine\_identifier >$
sees $\star$
$< context\_identifier >$
$\ldots$
variables
$< variable\_identifier >$
$\ldots$
invariants
$< label >: < predicate >$
$\ldots$
theorems $\star$
$< label >: < predicate >$
$\ldots$
events
initialisation $\ldots$
$\ldots$
variant $\star$
$< variant >$
end

- Each machine has exactly one **initialisation** event
- All keyword sections are predefined in the Rodin Platform
- All labels are generated automatically by the Rodin Platform (but can be modified)

Event Structure

$< event\_identifier > \widehat{=}$
status
$\{ordinary, convergent, anticipated\}$
refines $\star$
$< event\_identifier >$
$\ldots$
any $\star$
$< parameter\_identifier >$
$\ldots$
where $\star$
$< label >: < predicate >$
$\ldots$
with $\star$
$< label >: < witness >$
$\ldots$
then $\star$
$< label >: < action >$
$\ldots$
end

- Notice that keyword "**where**" becomes "**when**" in the Rodin Platform Pretty Print when there is no "**any**".
- Again, all keyword sections are predefined in the Rodin Platform.
- All labels are generated automatically by the Rodin Platform (but can be modified)

Formal Definition of Invariant Preservation (INV)

evt
any x **where**
$G(x, s, c, v)$
then
$v :| BAP(x, s, c, v, v')$
end

s	:	seen sets
c	:	seen constants
v	:	variables
$A(s, c)$	:	seen axioms and thms
$I(s, c, v)$	:	invariants and thms.
evt	:	specific event
x	:	event parameters
$G(x, s, c, v)$	:	event guards
$BAP(x, s, c, v, v')$	:	event before-after predicate
$inv(s, c, v')$	:	modified specific invariant

Axioms Invariants Guards of the event Before-after predicate of the event ⊢ Modified Specific Invariant	$evt/inv/$INV

$A(s, c)$
$I(s, c, v)$
$G(x, s, c, v)$
$BAP(x, s, c, v, v')$
⊢
$inv(s, c, v')$

- In case of the initialization event, $I(s, c, v)$ is removed from the hypotheses

Formal Definition of the Feasibility PO (FIS)

evt
any x **where**
$G(x, s, c, v)$
then
$v :| BAP(x, s, c, v, v')$
end

s	:	seen sets
c	:	seen constants
v	:	variables
$A(s, c)$	:	seen axioms and thms
$I(s, c, v)$	:	invariants and thms.
evt	:	specific event
x	:	event parameters
$G(x, s, c, v)$	:	event guards
$BAP(x, s, c, v, v')$	:	event action

Axioms Invariants Guards of the event ⊢ $\exists v' \cdot$ Before-after predicate	$evt/act/$FIS

$A(s, c)$
$I(s, c, v)$
$G(x, s, c, v)$
⊢
$\exists v' \cdot BAP(x, s, c, v, v')$

Formal Definition of the Guard Strengthening PO (GRD)

evt0
any
x
where
$g(x, s, c, v)$
...
then
...
end

evt
refines
evt0
any
y
where
$H(y, s, c, w)$
with
$x : W(x, y, s, c, w)$
then
...
end

s	:	seen sets
c	:	seen constants
v	:	abstract variables
w	:	concrete variables
$A(s, c)$	:	seen axioms and thms
$I(s, c, v)$	:	abs. invts. and thms.
$J(s, c, v, w)$	:	conc. invts. and thms.
evt	:	specific concrete event
x	:	abstract event parameter
y	:	concrete event parameter
$g(x, s, c, v)$	:	abstract event specific guard
$H(y, s, c, w)$	:	concrete event guards

Axioms Abstract invariants and thms. Concrete invariants and thms. Concrete event guards witness predicate ⊢ Abstract event specific guard	$evt/grd/$GRD

$A(s, c)$
$I(s, c, v)$
$J(s, c, v, w)$
$H(y, s, c, w)$
$W(x, y, s, c, w)$
⊢
$g(x, s, c, v)$

- It is simplified when there are no parameters

Formal Definition of the Simulation PO (SIM)

evt0
any
x
where
...
then
$v :| BA1(v, v', \ldots)$
end

evt
refines
evt0
any
y
where
$H(y, s, c, w)$
with
$x : W1(x, y, s, c, w)$
$v' : W2(y, v', s, c, w)$
then
$w :| BA2(w, w', \ldots)$
end

s	:	seen sets
c	:	seen constants
v	:	abstract vrbls
w	:	concrete vrbls
$A(s, c)$	:	seen axioms and thms
$I(s, c, v)$	:	abs. invts. and thms.
$J(s, c, v, w)$	:	conc. invts. and thms.
evt	:	concrete event
x	:	abstract prm
y	:	concrete prm
$H(y, s, c, w)$	:	concrete guards
$BA1(v, v')$	:	abstract action
$BA2(w, w')$	:	concrete action

Axioms Abstract invariants and thms. Concrete invariants and thms. Concrete event guards witness predicate witness predicate Concrete before-after predicate ⊢ Abstract before-after predicate	$evt/act/$SIM

$A(s, c)$
$I(s, c, v)$
$J(s, c, v, w)$
$H(y, s, c, w)$
$W1(x, y, s, c, w)$
$W2(y, v', s, c, w)$
$BA2(w, w', \ldots)$
⊢
$BA1(v, v', \ldots)$

Formal Definition of the Numeric Variant PO (NAT)

machine m
refines ...
sees ...
variables v
invariants and thms. $I(s,c,v)$
theorems ...
events ...
variant $n(s,c,v)$
end

evt
status convergent
any x **where** $G(x,s,c,v)$
then A
end

s	:	seen sets
c	:	seen constants
v	:	variables
$A(s,c)$	:	seen axioms and thms
$I(s,c,v)$	:	abs. invts. and thms.
$J(s,c,v,w)$	:	conc. invts. and thms.
evt	:	specific event
x	:	event parameters
$G(x,s,c,v)$	:	event guards
$n(s,c,v)$	:	numeric variant

Axioms Abstract invariants and thms. Concrete invariants and thms. Event guards ⊢ a numeric variant is a natural number	evt/NAT

$A(s,c)$
$I(s,c,v)$
$J(s,c,v,w)$
$G(x,s,c,v)$
⊢
$n(s,c,v) \in \mathbb{N}$

Formal Definition of the Set Variant PO (FIN)

machine m
refines ...
sees ...
variables v
invariants and thms. $J(s,c,v,w)$
theorems ...
events ...
variant $t(s,c,v)$
end

s	:	seen sets
c	:	seen constants
v	:	variables
$A(s,c)$	:	seen axioms and thms
$I(s,c,v)$	:	abs. invts. and thms.
$J(s,c,v,w)$	:	conc. invts. and thms.
$t(s,c,v)$	:	set variant

Axioms Abstract invariants and thms. Concrete invariants and thms. ⊢ Finiteness of set variant	FIN

$A(s,c)$
$I(s,c,v)$
$J(s,c,v,w)$
⊢
$\mathrm{finite}(t(s,c,v))$

Formal Definition of Numeric Variant Decreasing PO (VAR)

evt
status convergent
any x **where** $G(x,s,c,w)$
then $v :\mid BAP(x,s,c,w,w')$
end

s	:	seen sets
c	:	seen constants
v	:	variables
$A(s,c)$	:	seen axioms and thms
$I(s,c,v)$	:	abs. invts. and thms.
$J(s,c,v,w)$	:	conc. invts. and thms.
evt	:	specific event
x	:	event parameters
$G(x,s,c,v)$	:	event guards
$BAP(x,s,c,w,w')$	:	event before-after predicate
$n(s,c,w)$	:	numeric variant

Axioms Abstract invariants and thms. Concrete invariants and thms. Guards of the event Before-after predicate of the event ⊢ Modified variant smaller than variant	evt/VAR

$A(s,c)$
$I(s,c,v)$
$J(s,c,v,w)$
$G(x,s,c,w)$
$BAP(x,s,c,w,w')$
⊢
$n(s,c,w') < n(s,c,w)$

Formal Definition of the Set Variant Decreasing PO (VAR)

evt
status convergent
any x **where** $G(x,s,c,w)$
then $v :\mid BAP(x,s,c,w,w')$
end

s	:	seen sets
c	:	seen constants
v	:	variables
$A(s,c)$	:	seen axioms and thms
$I(s,c,v)$	:	abs. invts. and thms.
$J(s,c,v,w)$	:	conc. invts. and thms.
evt	:	specific event
x	:	event parameters
$G(x,s,c,v)$	:	event guards
$BAP(x,s,c,w,w')$	:	event before-after predicate
$t(s,c,w)$	:	set variant

Axioms Abstract Invariants Concrete Invariants Guards of the event Before-after predicate of the event ⊢ Modified variant strictly included in variant	evt/VAR

$A(s,c)$
$I(s,c,v)$
$J(s,c,v,w)$
$G(x,s,c,v)$
$BAP(x,s,c,w,w')$
⊢
$t(s,c,w') \subset t(s,c,w)$

Formal Definition of the Witness Feasibility PO (WFIS)

evt
refines
evt0
any
y
where
$H(y, s, c, w)$
with
$x : W(x, y, s, c, w)$
then
$\ldots$
end

s	:	seen sets
c	:	seen constants
v	:	abstract variables
w	:	concrete variables
$A(s, c)$	:	seen axioms and thms
$I(s, c, v)$	:	abs. invts. and thms.
$J(s, c, v, w)$	:	conc. invts. and thms.
evt	:	specific concrete event
x	:	abstract event parameter
y	:	concrete event parameter
$H(y, s, c, w)$	:	concrete event guards
$W(x, y, s, c, w)$	:	witness predicate

Axioms Abstract invariants and thms. Concrete invariants and thms. Concrete event guards $\vdash$ $\exists x \cdot$ Witness	$evt/x/$WFIS

$A(s, c)$
$I(s, c, v)$
$J(s, c, v, w)$
$H(y, s, c, w)$
$\vdash$
$\exists x \cdot W(x, y, s, c, w)$

Formal Definition of the Context Theorem PO (THM)

context
ctx
extends
$\ldots$
sets
s
constants
c
axioms
$A(s, c)$
theorems
$\ldots$
$thm :\ P(s, c)$
$\ldots$
end

s	:	seen sets
c	:	seen constants
$A(s, c)$	:	seen axioms and previous thms
$P(s, c)$	:	specific theorem

Axioms $\vdash$ Theorem	$thm/$THM

$A(s, c)$
$\vdash$
$P(s, c)$

Formal Definition of the Machine Theorem PO (THM)

machine
$m0$
refines
$\ldots$
sees
$\ldots$
variables
v
invariants and thms.
$I(s, c, v)$
theorems
$\ldots$
$thm :\ P(s, c, v)$
$\ldots$
events
$\ldots$
end

s	:	seen sets
c	:	seen constants
v	:	variables
$A(s, c)$	:	seen axioms and thms
$I(s, c, v)$	:	invariants and previous thms.
$P(s, c, v)$	:	specific theorem

Axioms Invariants $\vdash$ Theorem	$thm/$THM

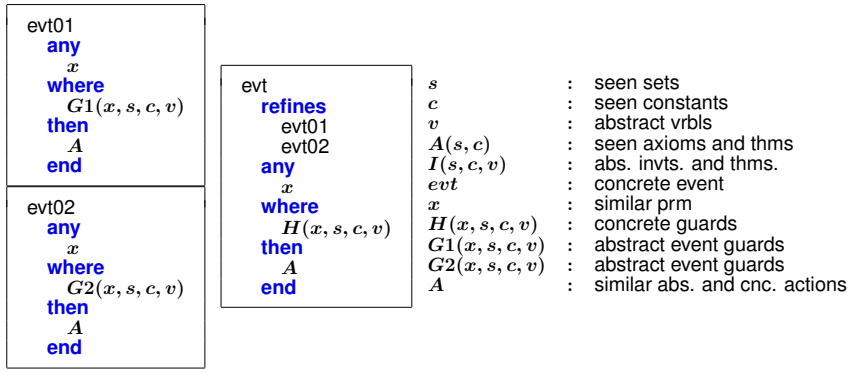
$A(s, c)$
$I(s, c, v)$
$\vdash$
$P(s, c, v)$

Formal Definition of the Well-definedness PO (WD)

- It depends on the potentially ill-defined expression

$\text{inter}\,(S)$	$S \neq \varnothing$
$\bigcap\ x \cdot x \in S\ \wedge\ P(x) \mid T(x)$	$\exists x \cdot x \in S\ \wedge P(x)$
$f(E)$	f is a partial function $E \in \text{dom}(f)$
E/F	$F \neq 0$
$E \bmod F$	$F \neq 0$
$\text{card}(S)$	$\text{finite}(S)$
$\min(S)$	$S \subseteq \mathbb{Z}$ $\exists x \cdot x \in \mathbb{Z}\ \wedge\ (\forall n \cdot n \in S\ \Rightarrow\ x \leq n)$
$\max(S)$	$S \subseteq \mathbb{Z}$ $\exists x \cdot x \in \mathbb{Z}\ \wedge\ (\forall n \cdot n \in S\ \Rightarrow\ x \geq n)$

Grd. Strengthening PO when Merging Abs. Events (MRG)



Axioms Abstract invariants and thms. Concrete event guards ⊢ Disjunction of abstract guards	evt/MRG

$A(s, c)$
$I(s, c, v)$
$H(x, s, c, v)$
⊢
$G1(x, s, c, v) \lor G2(x, s, c, v)$

ASCII Representations of the Mathematical Symbols (1)

- Atomic Symbols

ASCII	Symbol
true	$\top$
false	$\bot$
INT	$\mathbb{Z}$

ASCII	Symbol
NAT	$\mathbb{N}$
NAT1	$\mathbb{N}_1$
BOOL	**BOOL**

ASCII	Symbol
TRUE	**TRUE**
FALSE	**FALSE**
{}	$\varnothing$

- Assignment Operators

ASCII	Symbol
:=	$:=$

ASCII	Symbol
:|	$:\mid$

ASCII	Symbol
::	$:\in$

ASCII Representations of the Mathematical Symbols (2)

- Unary Operators

ASCII	Symbol
not	$\neg$
finite	**finite**
card	**card**
POW	$\mathbb{P}$
POW1	$\mathbb{P}_1$

ASCII	Symbol
union	**union**
inter	**inter**
dom	**dom**
ran	**ran**
prj1	$\mathbf{prj_1}$

ASCII	Symbol
prj2	$\mathbf{prj_2}$
id	**id**
min	**min**
max	**max**
-	$-$

ASCII Representations of the Mathematical Symbols (3)

- Binary Operators

ASCII	Symbol
&	$\land$
or	$\lor$
=>	$\Rightarrow$
<=>	$\Leftrightarrow$
=	$=$
/=	$\neq$
:	$\in$
<<:	$\subset$

ASCII	Symbol
/<<:	$\not\subset$
<:	$\subseteq$
/<:	$\not\subseteq$
<	$<$
<=	$\leq$
>	$>$
>=	$\geq$
/:	$\notin$

ASCII	Symbol
|-> or ,,	$\mapsto$
<->	$\leftrightarrow$
<<->	$\twoheadleftarrow$
<->>	$\twoheadrightarrow$
<<->>	$\twoheadleftarrow\!\!\!\!\twoheadrightarrow$
+->	$\nrightarrow$
-->	$\rightarrow$
+->>	$\twoheadrightarrow\!\!\!\!\!\!\rightarrow$

ASCII Representations of the Mathematical Symbols (4)

- Binary Operators (Cont.)

ASCII	Symbol
`-->>`	$\twoheadrightarrow$
`>+>`	$\rightarrowtail$
`>->`	$\rightarrowtail$
`>->>`	$\twoheadrightarrow$
`/\`	$\cap$
`\/`	$\cup$
`\`	$\setminus$
`**`	$\times$

ASCII	Symbol
`<+`	$\leftarrow$
`||`	$\parallel$
`><`	$\otimes$
`;`	;
`<|`	$\triangleleft$
`<<|`	$\trianglelefteq$
`|>`	$\triangleright$
`|>>`	$\trianglerighteq$

ASCII	Symbol
`*`	$*$
`/`	$\div$
`mod`	**mod**
`..`	$\ddot{.}$
`^`	$\hat{}$
`~`	$^{-1}$
`+`	$+$
`-`	$-$

ASCII Representations of the Mathematical Symbols (5)

- Quantifiers

ASCII	Symbol
`!`	$\forall$
`#`	$\exists$
`%`	$\boldsymbol{\lambda}$

ASCII	Symbol
`UNION`	$\bigcup$
`INTER`	$\bigcap$

ASCII	Symbol
`|`	$\mid$
`.`	$\cdot$

- Bracketing

ASCII	Symbol
`(`	(
`)`	)

ASCII	Symbol
`[`	[
`]`	]

ASCII	Symbol
`{`	{
`}`	}