Basic Inference Rules of Mathematical Reasoning

Reference Card

Jean-Raymond Abrial and Thai Son Hoang

April 2008

Contents

- Set-theoretic Axioms and Definitions: slides 8 to 20.
- Syntax of Event-B: slides 21 to 23.
- Proof Obligation Rules: slides 24 to 36.

Basic Inference Rules of Mathematical Reasoning

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HYP</td>
<td>( \frac{H, P \vdash P}{\text{HYP}} )</td>
</tr>
<tr>
<td>MON</td>
<td>( \frac{H \vdash Q}{\text{MON}} )</td>
</tr>
<tr>
<td>CUT</td>
<td>( \frac{H \vdash P \quad H, P \vdash Q}{\text{CUT}} )</td>
</tr>
</tbody>
</table>

Propositional Calculus Rules of Inference (1)

- Rules about conjunction

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AND_L</td>
<td>( \frac{H, P, Q \vdash R}{H, P \wedge Q \vdash R} )</td>
</tr>
<tr>
<td>AND_R</td>
<td>( \frac{H \vdash P \wedge Q}{H \vdash P \quad H \vdash Q} )</td>
</tr>
</tbody>
</table>

- Rules about implication

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMP_L</td>
<td>( \frac{H, P, Q \vdash R}{H, P, P \Rightarrow Q \vdash R} )</td>
</tr>
<tr>
<td>IMP_R</td>
<td>( \frac{H, P \vdash Q}{H \vdash P \Rightarrow Q} )</td>
</tr>
</tbody>
</table>

Note: Rules with a double horizontal line can be applied in both directions.
Propositional Calculus Rules of Inference (2)

- Rules about negation

\[ P, \neg P \vdash Q \]  \hspace{1cm} \text{NOT}_L

\[ \bot \vdash P \]  \hspace{1cm} \text{CNTR}

\[ H, P \vdash Q \hspace{1cm} H, P \vdash \neg Q \]  \hspace{1cm} \text{NOT}_R

\[ H, \neg P \vdash Q \hspace{1cm} H, \neg P \vdash \neg Q \]  \hspace{1cm} \text{NOT}_R

- Transforming a disjunctive goal

\[ H, \neg P \vdash Q \]  \hspace{1cm} \text{NEG}

- Rules about disjunction

\[ H, P \vdash R \hspace{1cm} H, Q \vdash R \]  \hspace{1cm} \text{OR}_L

\[ H \vdash P \]  \hspace{1cm} \text{OR}_1

\[ H \vdash P \lor Q \]  \hspace{1cm} \text{OR}_2

Predicate Calculus Rules of Inference

\[ H, \forall x \cdot P(x), P(E) \vdash Q \]  \hspace{1cm} \text{ALL}_L

\[ H \vdash P(x) \]  \hspace{1cm} \text{ALL}_R

\[ H, P(x) \vdash Q \]  \hspace{1cm} \text{XST}_L

\[ H \vdash P(E) \]  \hspace{1cm} \text{XST}_R

\[ H, \exists x \cdot P(x) \vdash Q \]  \hspace{1cm} \text{XST}_R

- In rule \text{ALL}_L and \text{XST}_R, \( E \) is an expression

- In rule \text{ALL}_R and \text{XST}_L, variable \( x \) is not free in \( H \).

Equality Rules of Inference

\[ H(F), E = F \vdash P(F) \]  \hspace{1cm} \text{EQ}_LR

\[ H(E), E = F \vdash P(E) \]  \hspace{1cm} \text{EQ}_RL

\[ \vdash E = E \]  \hspace{1cm} \text{EQL}

\[ H \vdash E = G \land F = I \]  \hspace{1cm} \text{PAIR}
Basic Set Operator Memberships (Axioms)

These axioms are defined by equivalences.

<table>
<thead>
<tr>
<th>Left Part</th>
<th>Right Part</th>
</tr>
</thead>
<tbody>
<tr>
<td>( E \leftrightarrow F \in S \times T )</td>
<td>( E \in S \land F \in T )</td>
</tr>
<tr>
<td>( S \in \mathcal{P}(T) )</td>
<td>( \forall x \cdot x \in S \Rightarrow x \in T )</td>
</tr>
<tr>
<td>( E \in { x \cdot x \in S \land P(x) \mid F(x) } )</td>
<td>( \exists x \cdot x \in S \land P(x) \land E = F(x) )</td>
</tr>
<tr>
<td>( E \in { x \mid x \in S \land P(x) } )</td>
<td>( E \in S \land P(E) )</td>
</tr>
</tbody>
</table>

Elementary Set Operator Memberships

| \( E \in S \cup T \) | \( E \in S \lor E \in T \) |
| \( E \in S \cap T \) | \( E \in S \land E \in T \) |
| \( E \in S \setminus T \) | \( E \in S \land E \notin T \) |
| \( E \in \{ a, \ldots, b \} \) | \( E = a \lor \ldots \lor E = b \) |
| \( E \in \emptyset \) | \( \bot \) |

Set Inclusion and Extensionality Axiom

<table>
<thead>
<tr>
<th>Left Part</th>
<th>Right Part</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S \subseteq T )</td>
<td>( S \in \mathcal{P}(T) )</td>
</tr>
<tr>
<td>( S = T )</td>
<td>( S \subseteq T \land T \subseteq S )</td>
</tr>
</tbody>
</table>

The first rule is just a syntactic extension.
The second rule is the Extensionality Axiom.

Generalizations of Elementary Operator Memberships

| \( E \in \text{union}(S) \) | \( \exists s \cdot s \in S \land E \in s \) |
| \( E \in \bigcup x \cdot x \in S \land P(x) \mid T(x) \) | \( \exists x \cdot x \in S \land P(x) \land E \in T(x) \) |
| \( E \in \text{inter}(S) \) | \( \forall s \cdot s \in S \Rightarrow E \in s \) |
| \( E \in \bigcap x \cdot x \in S \land P(x) \mid T(x) \) | \( \forall x \cdot x \in S \land P(x) \Rightarrow E \in T(x) \) |

Well-definedness condition for case 3: \( S \neq \emptyset \)
Well-definedness condition for case 4: \( \exists x \cdot x \in S \land P(x) \)
### Binary Relation Operator Memberships (1)

<table>
<thead>
<tr>
<th>Left Part</th>
<th>Right Part</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r \in S \leftrightarrow T$</td>
<td>$r \subseteq S \times T$</td>
</tr>
<tr>
<td>$E \in \text{dom}(r)$</td>
<td>$\exists y \cdot E \leftrightarrow y \in r$</td>
</tr>
<tr>
<td>$F \in \text{ran}(r)$</td>
<td>$\exists x \cdot x \leftrightarrow F \in r$</td>
</tr>
<tr>
<td>$E \leftrightarrow F \in r^{-1}$</td>
<td>$F \leftrightarrow E \in r$</td>
</tr>
</tbody>
</table>

### Binary Relation Operator Memberships (2)

<table>
<thead>
<tr>
<th>Left Part</th>
<th>Right Part</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r \in S \leftrightarrow T$</td>
<td>$r \in S \leftrightarrow T \land \text{ran}(r) = T$</td>
</tr>
<tr>
<td>$r \in S \leftrightarrow T$</td>
<td>$r \in S \leftrightarrow T \land \text{dom}(r) = T$</td>
</tr>
<tr>
<td>$r \in S \leftrightarrow T$</td>
<td>$r \in S \leftrightarrow T \land r \in S \leftrightarrow T$</td>
</tr>
</tbody>
</table>

### Binary Relation Operator Memberships (3)

<table>
<thead>
<tr>
<th>Left Part</th>
<th>Right Part</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E \leftrightarrow F \in S \triangleleft r$</td>
<td>$E \in S \land E \leftrightarrow F \in r$</td>
</tr>
<tr>
<td>$E \leftrightarrow F \in r \triangleright T$</td>
<td>$E \leftrightarrow F \in r \land F \in T$</td>
</tr>
<tr>
<td>$E \leftrightarrow F \in S \triangleleft r$</td>
<td>$E \notin S \land E \leftrightarrow F \in r$</td>
</tr>
<tr>
<td>$E \leftrightarrow F \in r \triangleright T$</td>
<td>$E \leftrightarrow F \in r \land F \notin T$</td>
</tr>
</tbody>
</table>

### Binary Relation Operator Memberships (4)

<table>
<thead>
<tr>
<th>Left Part</th>
<th>Right Part</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F \in r[w]$</td>
<td>$\exists x \cdot x \in w \land x \leftrightarrow F \in r$</td>
</tr>
<tr>
<td>$E \leftrightarrow F \in (p ; q)$</td>
<td>$\exists x \cdot E \leftrightarrow x \in p \land x \leftrightarrow F \in q$</td>
</tr>
<tr>
<td>$p \leftrightarrow q$</td>
<td>$(\text{dom}(q) \triangleleft p) \cup q$</td>
</tr>
<tr>
<td>$E \leftrightarrow F \in \text{id}(S)$</td>
<td>$E \in S \land F = E$</td>
</tr>
</tbody>
</table>
Given a relation \( r \) such that \( r \in S \leftrightarrow S \)

\[
\begin{align*}
  r &= r^{-1} & \text{\( r \) is symmetric} \\
  r \cap r^{-1} &= \emptyset & \text{\( r \) is asymmetric} \\
  r \cap r^{-1} &\subseteq \text{id}(S) & \text{\( r \) is antisymmetric} \\
  \text{id}(S) &\subseteq r & \text{\( r \) is reflexive} \\
  r \cap \text{id}(S) &= \emptyset & \text{\( r \) is irreflexive} \\
  r; r &\subseteq r & \text{\( r \) is transitive}
\end{align*}
\]
Applying a Function

Given a partial function \( f \), we have

<table>
<thead>
<tr>
<th>Left Part</th>
<th>Right Part</th>
</tr>
</thead>
<tbody>
<tr>
<td>( F = f(E) )</td>
<td>( E \mapsto F \in f )</td>
</tr>
</tbody>
</table>

Well-definedness conditions: \( f \) is a partial function

Context Structure

- Sections with "*" 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

- 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

- 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) 24

\[ \text{evt} \]
\[
\text{any } x \text{ where } \\
G(x, s, c, v) \\
\text{then} \\
\text{v : BAP}(x, s, c, v, v') \\
\text{end} \\
\text{\textit{Modified Specific Invariant}}
\]

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

Formal Definition of the Guard Strengthening PO (GRD) 26

\[ \text{evt} \]
\[
\text{any } x \text{ where } \\
\text{G}(x, s, c, v) \\
\text{then} \\
\text{v : BAP}(x, s, c, v, v') \\
\text{end} \\
\text{\textit{Abstract before-after predicate}}
\]

- It is simplified when there are no parameters
Formal Definition of the Numeric Variant PO (NAT)  

\[
\text{Formal Definition of the Numeric Variant PO (NAT)}  
\]

Formal Definition of the Numeric Variant PO (NAT)

\[
\begin{align*}
\text{machine} & \quad m \\
\text{refines} & \quad \ldots \\
\text{sees} & \quad \ldots \\
\text{variables} & \quad v \\
\text{invariants and thms.} & \quad I(s, c, v) \\
\text{events} & \quad \ldots \\
\text{variant} & \quad n(s, c, v) \\
\end{align*}
\]

\[
\begin{align*}
\text{evt} \quad \text{status} & \quad \text{convergent} \\
\text{any} \quad x & \quad \text{where} \\
G(x, s, c, v) & \quad \text{then} \\
A(s, c) & \quad \text{variables} \\
I(s, c, v) & \quad \text{seen constants} \\
J(s, c, v, w) & \quad \text{seen sets} \\
\text{evt} & \quad \text{numeric variant} \\
x & \quad \text{guards} \\
G(x, s, c, v) & \quad \text{event parameters} \\
n(s, c, v) & \quad \text{before-after predicate} \\
\end{align*}
\]

Axioms
Abstract invariants and thms.
Concrete invariants and thms.
Event guards
\[\vdash n(s, c, v) \in \mathbb{N}\]

Formal Definition of the Set Variant PO (FIN)

\[
\text{Formal Definition of the Set Variant PO (FIN)}  
\]

Formal Definition of the Set Variant PO (FIN)

\[
\begin{align*}
\text{machine} & \quad m \\
\text{refines} & \quad \ldots \\
\text{sees} & \quad \ldots \\
\text{variables} & \quad v \\
\text{invariants and thms.} & \quad I(s, c, v) \\
\text{events} & \quad \ldots \\
\text{variant} & \quad t(s, c, v) \\
\end{align*}
\]

\[
\begin{align*}
\text{evt} \quad \text{status} & \quad \text{convergent} \\
\text{any} \quad x & \quad \text{where} \\
G(x, s, c, v) & \quad \text{then} \\
A(s, c) & \quad \text{variables} \\
I(s, c, v) & \quad \text{seen constants} \\
J(s, c, v, w) & \quad \text{seen sets} \\
\text{evt} & \quad \text{numeric variant} \\
x & \quad \text{guards} \\
G(x, s, c, v) & \quad \text{event parameters} \\
t(s, c, v) & \quad \text{before-after predicate} \\
\end{align*}
\]

Axioms
Abstract Invariants
Concrete Invariants
Event guards
\[\vdash \text{finite}(t(s, c, v))\]

Formal Definition of the Numeric Variant Decreasing PO (VAR)

\[
\text{Formal Definition of the Numeric Variant Decreasing PO (VAR)}  
\]

Formal Definition of the Numeric Variant Decreasing PO (VAR)

\[
\begin{align*}
\text{machine} & \quad m \\
\text{refines} & \quad \ldots \\
\text{sees} & \quad \ldots \\
\text{variables} & \quad v \\
\text{invariants and thms.} & \quad I(s, c, v) \\
\text{events} & \quad \ldots \\
\text{variant} & \quad n(s, c, v) \\
\end{align*}
\]

\[
\begin{align*}
\text{evt} \quad \text{status} & \quad \text{convergent} \\
\text{any} \quad x & \quad \text{where} \\
G(x, s, c, w) & \quad \text{then} \\
A(s, c) & \quad \text{variables} \\
I(s, c, v) & \quad \text{seen constants} \\
J(s, c, v, w) & \quad \text{seen sets} \\
\text{evt} & \quad \text{numeric variant} \\
x & \quad \text{guards} \\
G(x, s, c, v) & \quad \text{event parameters} \\
n(s, c, v) & \quad \text{before-after predicate} \\
\end{align*}
\]

Axioms
Abstract invariants and thms.
Concrete invariants and thms.
Event guards
\[\vdash n(s, c, v) \in \mathbb{N}\]

Formal Definition of the Set Variant Decreasing PO (VAR)

\[
\text{Formal Definition of the Set Variant Decreasing PO (VAR)}  
\]

Formal Definition of the Set Variant Decreasing PO (VAR)

\[
\begin{align*}
\text{machine} & \quad m \\
\text{refines} & \quad \ldots \\
\text{sees} & \quad \ldots \\
\text{variables} & \quad v \\
\text{invariants and thms.} & \quad I(s, c, v) \\
\text{events} & \quad \ldots \\
\text{variant} & \quad t(s, c, v) \\
\end{align*}
\]

\[
\begin{align*}
\text{evt} \quad \text{status} & \quad \text{convergent} \\
\text{any} \quad x & \quad \text{where} \\
G(x, s, c, w) & \quad \text{then} \\
A(s, c) & \quad \text{variables} \\
I(s, c, v) & \quad \text{seen constants} \\
J(s, c, v, w) & \quad \text{seen sets} \\
\text{evt} & \quad \text{numeric variant} \\
x & \quad \text{guards} \\
G(x, s, c, v) & \quad \text{event parameters} \\
t(s, c, v) & \quad \text{before-after predicate} \\
\end{align*}
\]

Axioms
Abstract Invariants
Concrete Invariants
Event guards
\[\vdash \text{finite}(t(s, c, v))\]
Formal Definition of the Witness Feasibility PO (WFIS) 32

\[ \text{evt refines evt0 any y} \]

where \( H(y, s, c, w) \)

with \( x : W(x, y, s, c, w) \)

\[ \exists x \cdot \text{Witness} \]

Formal Definition of the Context Theorem PO (THM) 33

\[ \text{context ctx extends . . . sets s constants c axioms A(s, c) theorems P(s, c) . . . end} \]

\[ \text{Axioms \cdot \cdot \cdot thm : P(s, c)} \]

Formal Definition of the Machine Theorem PO (THM) 34

\[ \text{machine m0 refines . . . sees . . . variables v invariants and thms. I(s, c, v) theorems . . . thm : P(s, c, v) . . . events . . . end} \]

\[ \text{Axioms \cdot \cdot \cdot thm/THM \vdash \text{P(s, c)}} \]

Formal Definition of the Well-definedness PO (WD) 35

- It depends on the potentially ill-defined expression

\[ \text{inter (S) } \quad S \neq \varnothing \]

\[ \bigcap x \cdot x \in S \land P(x) \mid T(x) \quad \exists x \cdot x \in S \land P(x) \]

\[ f(E) \quad f \text{ is a partial function} \]

\[ E \mod F \quad F \neq 0 \]

\[ \text{card}(S) \quad \text{finite}(S) \]

\[ \text{min}(S) \quad S \subseteq \mathbb{Z} \land (\forall n \cdot n \in S \Rightarrow x \leq n) \]

\[ \text{max}(S) \quad S \subseteq \mathbb{Z} \land (\forall n \cdot n \in S \Rightarrow x \geq n) \]
 evt01
   any
   x
 where
 G1(x, s, c, v)
 then
   A
 end

 evt02
   any
   x
 where
 H(x, s, c, v)
 then
 A
 end

 evt refines evt01 evt02
 any
 x
 where
 G1(x, s, c, v) \lor G2(x, s, c, v)
 then
 A
 end

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

 ASCII Representations of the Mathematical Symbols (1)

 - Atomic Symbols

 ASCII | Symbol
--------|--------
 true | ⊤
 false | ⊥
 INT | Z
 NAT | N
 NAT1 | N₁
 BOOL | TRUE FALSE
 {} | ∅

 - Assignment Operators

 ASCII | Symbol
--------|--------
 := :=
 :| :|
 :: :∈

 ASCII Representations of the Mathematical Symbols (2)

 - Unary Operators

 ASCII | Symbol
--------|--------
 not | ¬
 finite | finite
card | card
POW | P
POW1 | P₁

 - Binary Operators

 ASCII | Symbol
--------|--------
 & | ∧
 or | ∨
 => | ⇒
 <=> | ⇔
 = | =
 /= | ≠
 |-> | ↦→
 <-> | ↔
 /<<: | ∈
 /:<: | ⊂
 /:<-> | ⊆
 /><: | ⊇
 /::<> | ⊞
 /<->: | ⊠
 /<>: | ⊡
### Binary Operators (Cont.)

<table>
<thead>
<tr>
<th>ASCII</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--&gt;&gt;</code></td>
<td>→</td>
</tr>
<tr>
<td><code>&gt;&gt;&gt;</code></td>
<td>➔</td>
</tr>
<tr>
<td><code>--&gt;</code></td>
<td>↦</td>
</tr>
<tr>
<td><code>&gt;--&gt;</code></td>
<td>➓</td>
</tr>
<tr>
<td><code>/\</code></td>
<td>∩</td>
</tr>
<tr>
<td><code>\</code></td>
<td>∪</td>
</tr>
<tr>
<td><code>\</code></td>
<td>∈</td>
</tr>
<tr>
<td><code>*</code></td>
<td>×</td>
</tr>
</tbody>
</table>

### Quantifiers

<table>
<thead>
<tr>
<th>ASCII</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>!</code></td>
<td>∀</td>
</tr>
<tr>
<td><code>#</code></td>
<td>∃</td>
</tr>
<tr>
<td><code>%</code></td>
<td>λ</td>
</tr>
</tbody>
</table>

### Bracketing

<table>
<thead>
<tr>
<th>ASCII</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>(</code></td>
<td>(</td>
</tr>
<tr>
<td><code>)</code></td>
<td>)</td>
</tr>
<tr>
<td><code>[</code></td>
<td>[</td>
</tr>
<tr>
<td><code>]</code></td>
<td>]</td>
</tr>
<tr>
<td><code>{</code></td>
<td>{</td>
</tr>
<tr>
<td><code>}</code></td>
<td>}</td>
</tr>
</tbody>
</table>