The Definition of `Red JonPRL`, *the people's refinement logic*

The JonPRL Group

January 24, 2016

Contents

Chapter 1

Signatures

Decisively Smash The Formalist Clique!

Chairman Jon

A *signature* is a collection of definitions, including terms, tactics and theorems.

1.1 Grammar

The grammar of `Red JonPRL` signatures is presented in Figure 1.1. Note that an optional production of sort s is formatted $\langle s \rangle$ in the rules.

sigexp	$::=$	$\langle \cdot \rangle$	empty signature
		$\mathit{sigexp}\ \mathit{sigdec}.$	signature extension
sigdec	$::=$	$\texttt{Def}\ \mathit{opid}\langle[\mathit{params}]\rangle\langle(\mathit{args})\rangle : \mathit{sortid} = [\mathit{term}]$	operator definition
		$\texttt{Tac}\ \mathit{opid}\langle[\mathit{params}]\rangle\langle(\mathit{args})\rangle = [\mathit{term}]$	tactic definition
		$\texttt{Thm}\ \mathit{opid}\langle[\mathit{params}]\rangle\langle(\mathit{args})\rangle : [\mathit{term}]\ \texttt{by}\ [\mathit{term}]$	theorem declaration
params	$::=$	$\langle \cdot \rangle$	empty parameter list
		$\mathit{params}, \mathit{symbind}$	parameter list extension
args	$::=$	$\langle \cdot \rangle$	empty argument list
		$\mathit{args}, \mathit{metabind}$	argument list extension
$\mathit{symbind}$	$::=$	$\mathit{symid} : \mathit{sortid}$	symbol binding
$\mathit{metabind}$	$::=$	$\mathit{metaid} : \mathit{valence}$	metavariable binding
$\mathit{valence}$	$::=$	$\langle\langle\{\mathit{sortlist}\}\rangle\langle[\mathit{sortlist}]\rangle.\rangle\mathit{sortid}$	valence
$\mathit{sortlist}$	$::=$	$\langle \cdot \rangle$	empty sort list
		$\mathit{sortlist}, \mathit{sortid}$	sort list extension

Figure 1.1: Grammar of signature expressions. The identifier sorts *opid*, *sortid*, *symid* and *metaid* can be assumed to be arbitrary strings; the sort *term* is left uninterpreted.

1.2 Static Semantics

The static semantics for `Red JonPRL` signatures begins with a specification of the class of *semantic* objects that will serve as the meanings for the *syntactic* objects defined in Section 1.1. We assume an ambient abstract binding tree signature such that at least the following facts hold:

$$\frac{\overline{\texttt{tac}\ \mathit{sort}} \quad \overline{\texttt{thm}\ \mathit{sort}} \quad \overline{\texttt{exp}\ \mathit{sort}} \quad \overline{\texttt{opid}\ \mathit{sort}}}{\Upsilon \Vdash \texttt{prove} : (.\,\texttt{exp}, .\,\texttt{tac})\,\texttt{thm}}$$

Then, our semantic objects are defined as in Figure 1.2.

$$\begin{array}{rcll}
a, b & \in & \text{Sym} & \\
\mathfrak{m}, \mathfrak{n} & \in & \text{Metavar} & \\
\sigma, \tau & \in & \text{Sort} & \triangleq \{\, \tau \mid \tau\ \mathit{sort} \,\} \\
v & \in & \text{ProdValence} & \triangleq \{\, v \mid v\ \mathit{valence} \,\} \\
\vartheta & \in & \text{Opid} & \triangleq \text{Sym} \\
\Upsilon & \in & \text{Params} & \triangleq \text{Sym} \rightharpoonup \text{Sort} \\
\Theta & \in & \text{Args} & \triangleq \text{Metavar} \rightharpoonup \text{ProdValence} \\
M, N & \in & \text{Tm}(\Theta, \Upsilon, \tau) & \triangleq \{\, M \mid \Theta \triangleright \Upsilon \parallel \cdot \vdash M : \tau \,\} \\
D & \in & \text{Decl} & \triangleq \coprod_{\Upsilon, \Theta, \tau} \text{Tm}(\Theta, \Upsilon, \tau) \\
\Sigma & \in & \text{Sig} & \triangleq \text{Opid} \rightharpoonup \text{Decl}
\end{array}$$

Figure 1.2: Specification of the semantic objects.

A *natural semantics* hinges on the elaboration judgment $E \vdash A \Longrightarrow A'$, which means that the syntactic object A elaborates to the semantic object A' in the environment E. Let the $\Upsilon_\Sigma \in \text{Params}$ be defined as follows:

$$\Upsilon_\Sigma(\vartheta) \triangleq \begin{cases} \texttt{opid} & \mathit{if} \quad \vartheta \in \mathbf{dom}(\Sigma) \\ \bot & \mathit{otherwise} \end{cases}$$

Symbol Bindings

$$\boxed{\Sigma \vdash \mathit{symbind} \Longrightarrow (a, \tau)}$$

$$\frac{\Sigma \vdash \mathit{symid} \Longrightarrow a \quad \Sigma \vdash \mathit{sortid} \Longrightarrow \tau}{\Sigma \vdash \mathit{symid} : \mathit{sortid} \Longrightarrow (a, \tau)} \tag{1.1}$$

Metavariable Bindings

$$\boxed{\Sigma \vdash \mathit{metabind} \Longrightarrow (\mathfrak{m}, v)}$$

$$\frac{\Sigma \vdash \mathit{metaid} \Longrightarrow \mathfrak{m} \quad \Sigma \vdash \mathit{valence} \Longrightarrow v}{\Sigma \vdash \mathit{metaid} : \mathit{valence} \Longrightarrow (\mathfrak{m}, v)} \tag{1.2}$$

Parameters

$$\boxed{\Sigma \vdash \mathit{params} \Longrightarrow \Upsilon}$$

$$\overline{\Sigma \vdash \langle \cdot \rangle \Longrightarrow \{\}} \tag{1.3}$$

$$\frac{\Sigma \vdash \mathit{params} \Longrightarrow \Upsilon \quad \Sigma \vdash \mathit{symbind} \Longrightarrow (a, \tau)}{\Sigma \vdash \mathit{params}, \mathit{symbind} \Longrightarrow \Upsilon \cup a \mapsto \tau} \tag{1.4}$$

Arguments

$$\boxed{\Sigma \vdash \mathit{args} \Longrightarrow \Theta}$$

$$\overline{\Sigma \vdash \langle \cdot \rangle \Longrightarrow \{\}} \tag{1.5}$$

$$\frac{\Sigma \vdash \mathit{args} \Longrightarrow \Theta \quad \Sigma \vdash \mathit{metabind} \Longrightarrow (\mathfrak{m}, v)}{\Sigma \vdash \mathit{args}, \mathit{metabind} \Longrightarrow \Theta \cup \mathfrak{m} \mapsto v} \tag{1.6}$$

Operator Identifiers

$$\boxed{\Sigma \vdash \mathit{opid} \Longrightarrow \vartheta}$$

$$\frac{\vartheta \notin \mathbf{dom}(\Sigma)}{\Sigma \vdash \mathit{opid} \Longrightarrow \vartheta} \tag{1.7}$$

Declarations

$$\boxed{\Sigma \vdash \mathit{sigdec} \Longrightarrow (\vartheta, D)}$$

$$\frac{\begin{array}{lll} \Sigma \vdash \mathit{params} \Longrightarrow \Upsilon & \Sigma \vdash \mathit{sortid} \Longrightarrow \tau & \Sigma \vdash \mathit{opid} \Longrightarrow \vartheta \\ \Sigma \vdash \mathit{args} \Longrightarrow \Theta & \Sigma \vdash \mathit{term} \Longrightarrow M & \Theta \triangleright \Upsilon_\Sigma \oplus \Upsilon \parallel \cdot \vdash M : \tau \end{array}}{\Sigma \vdash \texttt{Def}\ \mathit{opid}\langle[\mathit{params}]\rangle\langle(\mathit{args})\rangle : \mathit{sortid} = [\mathit{term}] \Longrightarrow (\vartheta, \langle \Upsilon, \Theta, \tau, M\rangle)} \tag{1.8}$$

$$\frac{\begin{array}{ll} \Sigma \vdash \mathit{params} \Longrightarrow \Upsilon & \\ \Sigma \vdash \mathit{args} \Longrightarrow \Theta & \Sigma \vdash \mathit{opid} \Longrightarrow \vartheta \\ \Sigma \vdash \mathit{term} \Longrightarrow M & \Theta \triangleright \Upsilon_\Sigma \oplus \Upsilon \parallel \cdot \vdash M : \texttt{tac} \end{array}}{\Sigma \vdash \texttt{Tac}\ \mathit{opid}\langle[\mathit{params}]\rangle\langle(\mathit{args})\rangle = [\mathit{term}] \Longrightarrow (\vartheta, \langle \Upsilon, \Theta, \texttt{tac}, M\rangle)} \tag{1.9}$$

$$\frac{\begin{array}{llll} \Sigma \vdash \mathit{params} \Longrightarrow \Upsilon & \Sigma \vdash \mathit{term}_1 \Longrightarrow P & \Theta \triangleright \Upsilon_\Sigma \oplus \Upsilon \parallel \cdot \vdash P : \texttt{exp} & \\ \Sigma \vdash \mathit{args} \Longrightarrow \Theta & \Sigma \vdash \mathit{term}_2 \Longrightarrow M & \Theta \triangleright \Upsilon_\Sigma \oplus \Upsilon \parallel \cdot \vdash M : \texttt{tac} & \Sigma \vdash \mathit{opid} \Longrightarrow \vartheta \end{array}}{\Sigma \vdash \texttt{Thm}\ \mathit{opid}\langle[\mathit{params}]\rangle\langle(\mathit{args})\rangle : [\mathit{term}_1]\ \texttt{by}\ [\mathit{term}_2] \Longrightarrow (\vartheta, \langle \Upsilon, \Theta, \texttt{thm}, \texttt{prove}(P; M)\rangle)} \tag{1.10}$$

Signatures

$$\boxed{\vdash \mathit{sigexp} \Longrightarrow \Sigma}$$

$$\frac{}{\vdash \langle \cdot \rangle \Longrightarrow \{\}} \tag{1.11}$$

$$\frac{\vdash \mathit{sigexp} \Longrightarrow \Sigma \quad \Sigma \vdash \mathit{sigdec} \Longrightarrow (\vartheta, D)}{\vdash \mathit{sigexp}\ \mathit{sigdec}. \Longrightarrow \Sigma \cup \vartheta \mapsto D} \tag{1.12}$$

Chapter 2

Nominal LCF: a language for tactics

In a sequent calculus, left rules add hypotheses to the context; for instance, consider the left rule for positive conjunctions:

$$\frac{H, x : A \otimes B, y : A, z : B \gg [\langle y, z \rangle / x]\, C}{H, x : A \otimes B \gg C}\ \otimes_L^{x,y,z}$$

From a proof refinement perspective (see [2]), such a rule is typically manifested as an ML tactic $\otimes_{\mathtt{L}}[x, y, z]$ which takes three names as parameters: the target hypothesis x, and the names to use for the new hypotheses y, z. However, whilst the identity of the name x is essential to the meaning of the tactic, the names supplied for the generated hypotheses can be freshly renamed with impunity.

Indeed, in a proof term assignment for this sequent calculus, the corresponding elimination form would *bind* variables x, y rather than take them as parameters. However, in the standard LCF tactic paradigm, it is not possible to reproduce this structure, because the sequencing of rules is mediated by the general purpose `THEN` tactical, which has no knowledge of names or binding.

We will design a language for tactics called **Nominal LCF** which supports a distinction between names bound and names taken as parameters, and then show how it can be elaborated into standard LCF.

2.1 The LCF Tactic Language

The essence of the LCF tactic system is captured in the following (idealized) ML signature:

$$\begin{array}{lcl}
\texttt{type}\ \mathit{judgment} & & \\
\texttt{type}\ \mathit{evidence} & & \\
\texttt{type}\ \mathit{state} & = & \mathit{judgment\ list} \otimes (\mathit{evidence\ list} \rightharpoonup \mathit{evidence}) \\
\texttt{type}\ \mathit{tactic} & = & \mathit{judgment} \rightharpoonup \mathit{state}
\end{array}$$

In other words, a tactic is a partial function that takes a goal to its subgoals, and specifies how to transform the evidence of its subgoals into the evidence for the main goal. In the case of the sequence calculus we were considering, `judgment` would be a type of sequents. We will write $|P|$ for the list of subgoals $\pi_1(P)$ to a proof state $P \in \mathit{state}$, $\|P\|$ for the length of $|P|$, and $P \star \overrightarrow{x}$ for the application of the hypothetical evidence $\pi_2(P)$ to the concrete evidence $\overrightarrow{x} \in \mathit{evidence\ list}$.

Now, in general a tactic may need to consume names from a *name store*, which is an infinite stream of *atoms* or *symbols*:

$$\begin{array}{lcl}
\texttt{type}\ \mathbb{A} & & \\
\texttt{type}\ \mathit{atactic} & = & \mathbb{A}^{\mathbb{N}} \to \mathit{tactic}
\end{array}$$

2.1.1 Atomic tactics and their continuity

Left sequent rules can be coded as so-called *atomic tactics*, tactics which consume a stream of names. In fact, every such tactic is *continuous* in a specific sense. For an atomic sequence $\alpha \in \mathbb{A}^{\mathbb{N}}$ and a natural number $n \in \mathbb{N}$, let $\overline{\alpha}[n]$ be the initial segment of α of length n; let $[n]\overline{\alpha}$ be infinite suffix of α got by chopping off the initial prefix $\overline{\alpha}[n]$. Let $M \approx N$ be *observational equivalence*: M and N evaluate to equal values, or they both diverge.

Then, for any atomic tactic T and judgment J, we can calculate a modulus of continuity:

$$\forall \alpha \in \mathbb{A}^{\mathbb{N}}.\exists n \in \mathbb{N}.\forall \beta \in \mathbb{A}^{\mathbb{N}}.\overline{\alpha}(n) = \overline{\beta}(n) \implies T(\alpha, J) \approx T(\beta, J) \tag{continuity}$$

This calculation can be realized computationally in our metalanguage in a number of ways, but for our purposes it suffices to remark that it is a fact concerning all computable stream processors. Let $\mathsf{M}(T, J) \in \mathbb{A}^{\mathbb{N}} \to \mathbb{N}$ calculate the modulus of continuity for an atomic tactic $T \in \mathit{atactic}$ at goal J.

2.2 Nominal LCF

2.2.1 Static Semantics

We will define the **Nominal LCF** language by specifying an abt signature for it; at its heart is decomposition of the various sequencing tacticals `THEN`, `THENL`, etc. of LCF into a single sequencing tactical combined a separate notion of *multi-tactic*.

First, we define sorts for tactics, atomic tactics, binding tactics and multi-tactics respectively:

$$\frac{}{\mathtt{tac}\ \mathit{sort}} \qquad \frac{}{\mathtt{atac}\ \mathit{sort}} \qquad \frac{}{\mathtt{btac}\ \mathit{sort}} \qquad \frac{}{\mathtt{mtac}\ \mathit{sort}}$$

We also provide a sort to classify hypotheses:

$$\frac{}{\mathtt{hyp}\ \mathit{sort}}$$

Now, we'll define the operators of **Nominal LCF**; note that the operators of sort `atac` are arbitrary and are provided only for the sake of illustration. We will consider the definition of **Nominal LCF** as over some signature Σ of atomic tactics.

$$\frac{}{\Upsilon \Vdash \mathtt{id} : ()\,\mathtt{atac}} \qquad \frac{}{\Upsilon \Vdash \mathtt{fail} : ()\,\mathtt{atac}} \qquad \frac{}{\Upsilon, a : \mathtt{hyp} \Vdash \mathtt{elim}[a] : ()\,\mathtt{atac}}$$

$$\frac{n \in \mathbb{N}}{\Upsilon \Vdash \mathtt{seq}_n : (.\,\mathtt{btac}, \{\mathtt{hyp}^n\}.\,\mathtt{mtac})\,\mathtt{tac}}$$

$$\frac{n \in \mathbb{N}}{\Upsilon \Vdash \mathtt{smash} : (.\,\mathtt{atac}, .\,\mathtt{atac})\,\mathtt{btac}}$$

$$\frac{}{\Upsilon \Vdash \mathtt{all} : (.\,\mathtt{tac})\,\mathtt{mtac}} \qquad \frac{n \in \mathbb{N}}{\Upsilon \Vdash \mathtt{each}_n : (.\,\mathtt{tac}^n)\,\mathtt{mtac}} \qquad \frac{i \in \mathbb{N}}{\Upsilon \Vdash \mathtt{some}_i : (.\,\mathtt{tac})\,\mathtt{mtac}}$$

For the sake of clarity, we introduce the following notational abbreviations for tactic expressions:

$$\begin{aligned}
t_1; t_2 &\triangleq \mathtt{seq}_0(.\,t_1; .\,t_2)\\
a_0, \ldots, a_n \leftarrow t_1; t_2 &\triangleq \mathtt{seq}_n(.\,t_1; \{a_0, \ldots, a_n\}.\,t_2)\\
t_1 \mathrel{\ni\!\!\!\!-} t_2 &\triangleq \mathtt{smash}(.\,t_1; .\,t_n)\\
\Box t &\triangleq \mathtt{all}(.\,t)\\
(\!| t_1, \ldots, t_n |\!) &\triangleq \mathtt{each}_n(.\,t_1; \ldots; .\,t_n)\\
\Diamond_i t &\triangleq \mathtt{some}_i(.\,t)
\end{aligned}$$

2.2.2 Dynamic Semantics

For a signature of atomic tactics Σ, we will define a Σ-model $\mathcal{M}$ to be an interpretation of each atomic tactic $t \in \Sigma$ into some $[\![t]\!]_{\mathcal{M}} \in \mathit{atactic}$; additionally, the model shall come equipped with a free choice sequence of atoms $\alpha_{\mathcal{M}} \in \mathbb{A}^{\mathbb{N}}$, such that each neighborhood $\vec{u} \ni \alpha_{\mathcal{M}}$ shall contain only distinct atoms.[1] Then, we can interpret all of **Nominal LCF** into $\mathcal{M}$, by defining an elaboration judgment $\mathcal{M} \models_\rho \Upsilon \parallel \Gamma \vdash t \mathbin{@} J \Longrightarrow P$ with $J \in \mathit{judgment}$ and $P \in \mathit{state}$, such that $\rho(x) \in \mathit{atactic}$ for each $x : \mathtt{atac} \in \Gamma$.

First, we define the elaboration uniformly by appealing to an auxiliary judgment:

$$\frac{\mathcal{M} \models_\rho \Upsilon \parallel \Gamma \vdash t \mathbin{@} J \overset{\mu}{\Longrightarrow} P \ll \alpha_{\mathcal{M}}}{\mathcal{M} \models_\rho \Upsilon \parallel \Gamma \vdash t \mathbin{@} J \Longrightarrow P}$$

This auxiliary form of judgment is made with respect to a free choice sequence of names α, and synthesizes the modulus of continuity μ of the tactic under consideration. When it leads to no ambiguity, we will write $\mathcal{M} \models_\rho t \mathbin{@} J \overset{\mu}{\Longrightarrow} P \ll \alpha$ instead of the more verbose $\mathcal{M} \models_\rho \Upsilon \parallel \Gamma \vdash t \mathbin{@} J \overset{\mu}{\Longrightarrow} P \ll \alpha$; we will explain this form of judgment for t of sorts `atac`, `btac`, and `tac`. To start with, we will give the elaboration rules for constants (atomic tactics) and variables denoting atomic tactics:

$$\frac{\mathsf{M}([\![t]\!]_{\mathcal{M}}, J)(\alpha) \equiv \mu}{\mathcal{M} \models_\rho t \mathbin{@} J \overset{\mu}{\Longrightarrow} [\![t]\!]_{\mathcal{M}}(\alpha, J) \ll \alpha} \qquad \frac{\mathsf{M}(\rho(x), J)(\alpha) \equiv \mu}{\mathcal{M} \models_\rho x \mathbin{@} J \overset{\mu}{\Longrightarrow} \rho(x)(\alpha, J) \ll \alpha}$$

Let $\mathsf{split}(\vec{x}, \vec{n})$ be the operation that splits the list $\vec{x}$ into a list sublists of $\vec{x}$ of length n for each $n \in \vec{n}$. Then, for a proof state $P \in \mathit{state}$ and a vector of proof states $Q_i \in \mathit{state}$ $(i < \|P\|)$, we can form a new proof state collapses the two layers of subgoals into one, as follows:

$$\mathsf{split}_i^n(\vec{x}, Q_i) \triangleq \mathsf{split}(\vec{x}, [\|Q_i\| \mid i < n])$$

$$\bigtriangledown_i^P Q_i \triangleq \left(\bigoplus_{i \in \|P\|} |Q_i|, \lambda \vec{x}.\ P \star \left[Q_i \star \vec{y} \mid (\vec{y}, i) \in \left(\mathsf{split}_i^{\|P\|}(\vec{x}, Q_i), \mathbb{N}_{\|P\|} \right) \right] \right)$$

The rule for the *smash* tactical $t_1 \mathbin{\ni\!\!\!\!\supset} t_2$ exploits the continuity of atomic tactics to divide up the name store α between t_1 and t_2:

$$\frac{\begin{array}{l}\mathcal{M} \models_\rho t_1 \mathbin{@} J \overset{\mu}{\Longrightarrow} P \ll \alpha \\ \mathcal{M} \models_\rho t_2 \mathbin{@} |P|_i \overset{\mu_i}{\Longrightarrow} Q_i \ll [\mu]\overline{\alpha} \quad (i < \|P\|)\end{array} \qquad \mu + \bigsqcup_{i < \|P\|} \mu_i \equiv \mu'}{\mathcal{M} \models_\rho t_1 \mathbin{\ni\!\!\!\!\supset} t_2 \mathbin{@} J \overset{\mu'}{\Longrightarrow} \bigtriangledown_i^P Q_i \ll \alpha}$$

Next, we define the behavior of the sequencing tactical in a single rule by exploiting an auxiliary form of judgment concerning the action of multi-tactics on proof states, $\mathcal{M} \models_\rho t \mathbin{@} P \overset{\mu}{\Longrightarrow} Q \ll \alpha$ (with $P, Q \in \mathit{state}$):

$$\frac{\begin{array}{c}\mathcal{M} \models_\rho t_1 \mathbin{@} J \overset{\mu_1}{\Longrightarrow} P \ll \vec{u} \oplus \alpha \\ \mathcal{M} \models_\rho t_2 \mathbin{@} J_i \overset{\mu_2}{\Longrightarrow} Q \ll [\mu_1]\overline{\vec{u} \oplus \alpha}\end{array}}{\mathcal{M} \models_\rho \vec{u} \leftarrow t_1; t_2 \mathbin{@} J \overset{\mu_1 + \mu_2}{\Longrightarrow} Q \ll \alpha}$$

Finally, the action of the multi-tactics is defined:

$$\frac{\mathcal{M} \models_\rho t \mathbin{@} |P|_i \overset{\mu_i}{\Longrightarrow} Q_i \ll \alpha \quad (i < \|P\|) \quad \bigsqcup_{i < \|P\|} \mu_i \equiv \mu}{\mathcal{M} \models_\rho \Box t \mathbin{@} P \overset{\mu}{\Longrightarrow} \bigtriangledown_i^P Q_i \ll \alpha}$$

[1] This is an example of a *spread law* (see [1]).

$$\frac{\mathcal{M} \models_\rho t_i \mathbin{@} |P|_i \xRightarrow{\mu_i} Q_i \ll \alpha \quad (i < \|P\|) \qquad \bigsqcup_{i<\|P\|} \equiv \mu}{\mathcal{M} \models_\rho (\!| t_0, \ldots, t_{\|P\|} |\!) \mathbin{@} P \xRightarrow{\mu} \mathop{\nabla}\limits_{i}^{P} Q_i \ll \alpha}$$

$$\frac{\mathcal{M} \models_\rho t \mathbin{@} |P|_i \xRightarrow{\mu} Q \ll \alpha}{\mathcal{M} \models_\rho \Diamond_i t \mathbin{@} P \xRightarrow{\mu} \mathop{\nabla}\limits_{j}^{P} \left\{ \begin{array}{lcl} j = i & \mapsto & Q \\ j \neq i & \mapsto & \left([|P|_j], \lambda[x].x\right) \end{array} \right\} \ll \alpha}$$

Bibliography

[1] L. E. J. Brouwer and Dirk van Dalen. *Brouwer's Cambridge lectures on intuitionism.* Cambridge University Press Cambridge, 1981.

[2] R. L. Constable, S. F. Allen, H. M. Bromley, W. R. Cleaveland, J. F. Cremer, R. W. Harper, D. J. Howe, T. B. Knoblock, N. P. Mendler, P. Panangaden, J. T. Sasaki, and S. F. Smith. *Implementing Mathematics with the Nuprl Proof Development System.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1986.