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

The JonPRL Group

January 25, 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.

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

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

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

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

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

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

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 [1]), 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:

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

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.

Now, frequently a tactic may need to consume names from a *name store*, which is an infinite stream of *atoms* or *symbols*. We will call such a tactic a *nominal tactic*:

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

2.1.1 Nominal tactics and their continuity

Left sequent rules can be coded nominal tactics, pull the names for their bound variables from the supplied choice sequence of atoms In fact, every such tactic is *continuous* in a specific sense. For a choice 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 nominal tactic T and judgment J, we can calculate a modulus of continuity:

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

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 $\mathsf{M}(T) \in \mathit{judgment} \to \mathbb{A}^{\mathbb{N}} \to \mathbb{N}$ calculate the modulus of continuity for a nominal tactic $T \in \mathit{ntactic}$ with name store α.

2.2 Nominal LCF

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 nominal tactics and multi-tactics respectively:

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

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 tac 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 tactical constants.

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

$$\frac{}{\Upsilon \Vdash \mathtt{fix} : ([\mathtt{tac}].\,\mathtt{tac})\,\mathtt{tac}} \qquad \frac{}{\Upsilon \Vdash \mathtt{let} : (.\,\mathtt{tac}, [\mathtt{tac}].\,\mathtt{tac})\,\mathtt{tac}}$$

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

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

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

$$\begin{aligned}
t_1; t_2 &\triangleq \mathtt{seq}_0(.\,t_1; .\,t_2) \\
a_0, \ldots, a_n \leftarrow t_1; t_2 &\triangleq \mathtt{seq}_n(.\,t_1; \{a_0, \ldots, a_n\}.\,t_2) \\
t_1 \ni t_2 &\triangleq \mathtt{smoosh}(.\,t_1; .\,t_n) \\
\Box t &\triangleq \mathtt{all}(.\,t) \\
(\!|t_1, \ldots, t_n|\!) &\triangleq \mathtt{each}_n(.\,t_1; \ldots; .\,t_n) \\
\Diamond_i t &\triangleq \mathtt{some}_i(.\,t) \\
\mu x.\mathfrak{t}[x] &\triangleq \mathtt{fix}([x].\,\mathfrak{t}[x]) \\
\mathit{let}\ x := t_1\ \mathit{in}\ \mathfrak{t}_2[x] &\triangleq \mathtt{let}(.\,t_1; [x].\,\mathfrak{t}_2[x])
\end{aligned}$$

2.2.1 Denotational Semantics

We will now give a denotational semantics for **Nominal LCF** by interpreting every tactic expression into an LCF nominal tactic. The intepretation $\mathcal{M}\llbracket \Upsilon \parallel \Gamma \vdash t \rrbracket_\rho \equiv T$ is defined by recursion on $t : \texttt{tac}$ such that $T \in \mathit{ntactic}$, presupposing that $\rho(x) \in \mathit{ntactic}$ for each $x : \texttt{tac} \in \Gamma$. To start with, variable tactics are simply projected from the environment ρ:

$$\frac{}{\mathcal{M}\llbracket \Upsilon \parallel \Gamma \vdash x \rrbracket_\rho \equiv \rho(x)}$$

The basic tacticals are interpreted as follows (we omit the interpretation of $\texttt{elim}[a]$, which will depend on the logic):

$$\frac{}{\mathcal{M}\llbracket \Upsilon \parallel \Gamma \vdash \texttt{id} \rrbracket_\rho \equiv \lambda\alpha.\texttt{ID}} \qquad \frac{}{\mathcal{M}\llbracket \Upsilon \parallel \Gamma \vdash \texttt{fail} \rrbracket_\rho \equiv \lambda\alpha.\texttt{FAIL}}$$

$$\frac{\mathcal{M}\llbracket \Upsilon \parallel \Gamma \vdash t_1 \rrbracket_\rho \equiv T_1 \quad \mathcal{M}\llbracket \Upsilon \parallel \Gamma, x : \texttt{tac} \vdash \mathsf{t}_2[x] \rrbracket_{\rho, x \mapsto T_1} \equiv T_2}{\mathcal{M}\llbracket \Upsilon \parallel \Gamma \vdash \mathit{let}\ x := t_1\ \mathit{in}\ \mathsf{t}_2[x] \rrbracket_\rho \equiv T_2}$$

$$\frac{}{\mathcal{M}\llbracket \Upsilon \parallel \Gamma \vdash \mu x.\mathsf{t}[x] \rrbracket_\rho \equiv \texttt{fix}\left(\lambda T.\mathcal{M}\llbracket \Upsilon \parallel \Gamma, x : \texttt{tac} \vdash \mathsf{t}[x] \rrbracket_{\rho, x \mapsto T}\right)}$$

The rule for the *smoosh* tactical $t_1 \Supset t_2$ exploits the continuity of nominal tactics to divide up the name store between t_1 and t_2:

$$\frac{\mathcal{M}\llbracket \Upsilon \parallel \Gamma \vdash t_1 \rrbracket_\rho \equiv T_1 \quad \mathcal{M}\llbracket \Upsilon \parallel \Gamma \vdash t_2 \rrbracket_\rho \equiv T_2}{\mathcal{M}\llbracket \Upsilon \parallel \Gamma \vdash t_1 \Supset t_2 \rrbracket_\rho \equiv \lambda\alpha.\texttt{THEN}(T_1(\alpha), \lambda J.T_2([\mathsf{M}(T_1)(J,\alpha)]\overline{\alpha}, J))}$$

Next, we define the behavior of the sequencing tactical, by case on the multi-tactical:

$$\frac{\mathcal{M}\llbracket \Upsilon \parallel \Gamma \vdash t_1 \rrbracket_\rho \equiv T_1 \quad \mathcal{M}\llbracket \Upsilon, \vec{u} : \texttt{hyp} \parallel \Gamma \vdash t_2 \rrbracket_\rho \equiv T_2}{\mathcal{M}\llbracket \Upsilon \parallel \Gamma \vdash \vec{u} \leftarrow t_1; \Box t_2 \rrbracket_\rho \equiv \lambda\alpha.\texttt{THEN}(T_1(\vec{u} \oplus \alpha), \lambda J.T_2([\mathsf{M}(T_1)(J, \vec{u} \oplus \alpha)]\overline{\vec{u} \oplus \alpha}, J))}$$

$$\frac{\mathcal{M}\llbracket \Upsilon \parallel \Gamma \vdash t \rrbracket_\rho \equiv T \quad \mathcal{M}\llbracket \Upsilon, \vec{u} : \texttt{hyp} \parallel \Gamma \vdash t_i \rrbracket_\rho \equiv T_i \quad (i < n)}{\mathcal{M}\llbracket \Upsilon \parallel \Gamma \vdash \vec{u} \leftarrow t; (\!| t_0, \ldots, t_n |\!) \rrbracket_\rho \equiv \lambda\alpha.\texttt{THENL}(T(\vec{u} \oplus \alpha), \left[\lambda J.T_i([\mathsf{M}(T)(J, \vec{u} \oplus \alpha)]\overline{\vec{u} \oplus \alpha}, J) \mid i < n\right])}$$

$$\frac{\mathcal{M}\llbracket \Upsilon \parallel \Gamma \vdash t_1 \rrbracket_\rho \equiv T_1 \quad \mathcal{M}\llbracket \Upsilon, \vec{u} : \texttt{hyp} \parallel \Gamma \vdash t_2 \rrbracket_\rho \equiv T_2}{\mathcal{M}\llbracket \Upsilon \parallel \Gamma \vdash \vec{u} \leftarrow t_1; \Diamond_i t_2 \rrbracket_\rho \equiv \lambda\alpha.\texttt{THENF}(T_1(\vec{u} \oplus \alpha), i, \lambda J.T_2([\mathsf{M}(T_1)(J, \vec{u} \oplus \alpha)]\overline{\vec{u} \oplus \alpha}, J))}$$

Bibliography

[1] 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.