Coq as a Metatheory for Nuprl with Bar Induction

Vincent Rahli and Mark Bickford
http://www.nuprl.org

securityandtrust.lu

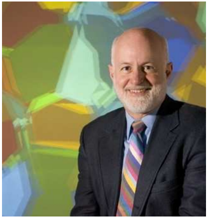
October 7, 2015

Overall Story

Mark Bickford

Luitzen Egbertus Jan Brouwer

Robert L. Constable



Nuprl in a Nutshell

Similar to Coq and Agda

Extensional Intuitionistic Type Theory for partial functions

Consistency proof in Coq:
`https://github.com/vrahli/NuprlInCoq`

Cloud based & virtual machines: `http://www.nuprl.org`

JonPRL: `http://www.jonprl.org`

Nuprl Stack

Tactics

Refiner

Inference rules

Allen's PER semantics

Howe's computational equality

An untyped applied lambda-calculus

Howe's Computational Equality

$\preccurlyeq$ is a simulation relation

Greatest fixpoint of the following relation: $t\ [R]\ u$ if whenever t computes to a value $\theta(\overline{b})$, then u also computes to a value $\theta(\overline{b'})$ such that $\overline{b}\ R\ \overline{b'}$.

Examples: $\perp \preccurlyeq 1$, $\langle \perp, 1 \rangle \preccurlyeq \langle 1, 1 \rangle$

$\sim$ is a bisimulation relation ($a \sim b = a \preccurlyeq b \ \wedge\ b \preccurlyeq a$)

Purely by computation:

$$\mathtt{map}(f\,,\mathtt{map}(g\,,l)) \ \sim\ \mathtt{map}(f \circ g\,,l)$$

$\preccurlyeq$ and $\sim$ are congruences

Howe's Computational Equality

Type checking and type inference are undecidable

Proving that terms are well-formed can be cumbersome

$\sim$ saves us from having to prove well-formedness

It turned out that many equalities could be stated using $\sim$

Nupri Types

Based on Martin-Löf's extensional type theory

Equality: $a = b \in T$

Dependent product: $a{:}A \rightarrow B[a]$

Dependent sum: $a{:}A \times B[a]$

Universe: $\mathbb{U}_i$

Nupl Types

Less "conventional types"

Partial: $\overline{A}$

Disjoint union: $A+B$

Intersection: $\cap a{:}A.B[a]$

Union: $\cup a{:}A.B[a]$

Subset: $\{a : A \mid B[a]\}$

Quotient: $T//E$

Domain: $\texttt{Base}$

Simulation: $t_1 \preccurlyeq t_2$

$(\texttt{Void} = 0 \preccurlyeq 1$ and $\texttt{Unit} = 0 \preccurlyeq 0)$

Bisimulation: $t_1 \sim t_2$

Image: $\texttt{Img}(A, f)$

PER: $\texttt{per}(R)$

Nuprl Types

Image type (Nogin & Kopylov)

Subset: $\{a : A \mid B[a]\} \triangleq \mathtt{Img}(a{:}A \times B[a], \pi_1)$

Union: $\cup a{:}A.B[a] \triangleq \mathtt{Img}(a{:}A \times B[a], \pi_2)$

Nuprl Types

PER type (inspired by Allen)

$$\texttt{Top} = \texttt{per}(\lambda\_, \_.0 \preccurlyeq 0)$$

$$\texttt{halts}(t) = \star \preccurlyeq (\texttt{let } x := t \texttt{ in } \star)$$

$$A \sqcap B = \cap x{:}\texttt{Base}. \cap y{:}\texttt{halts}(x).\texttt{isaxiom}(x, A, B)$$

$$T//E = \texttt{per}(\lambda x, y.(x \in T) \sqcap (y \in T) \sqcap (E\ x\ y))$$

Nuprl Types

Squashing

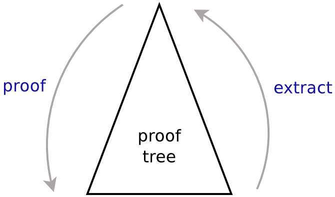
$\downarrow T$	$\{\mathtt{Unit} \mid T\}$ $\mathtt{Img}(T, \lambda\_.\star)$	$\mathtt{per}(\lambda x.\lambda y.\star \preccurlyeq x \sqcap \star \preccurlyeq y \sqcap T)$
$\downarrow T$	$T//\mathtt{True}$	$\mathtt{per}(\lambda x.\lambda y.x \in T \sqcap y \in T)$
$\Downarrow T$	$\mathtt{Top}//T$	$\mathtt{per}(\lambda\_.\lambda\_.T)$

Nuprl Refinements

Nuprl's proof engine is called a refiner (TB)

A generic goal directed reasoner:

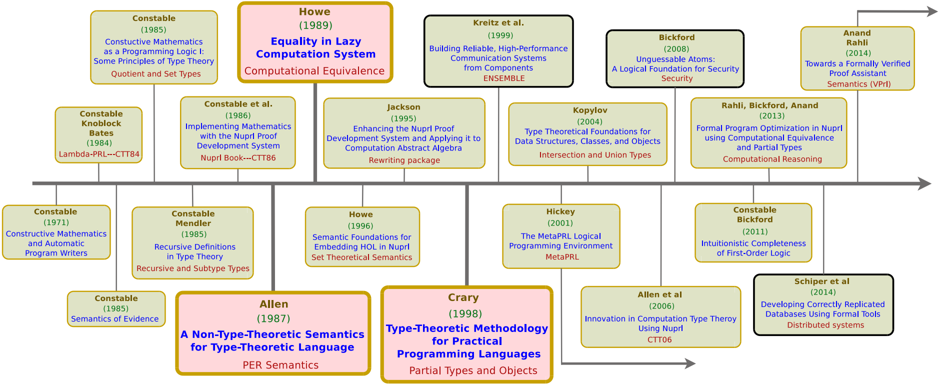
- **a rule interpreter**
- **a proof manager**



Example of a rule

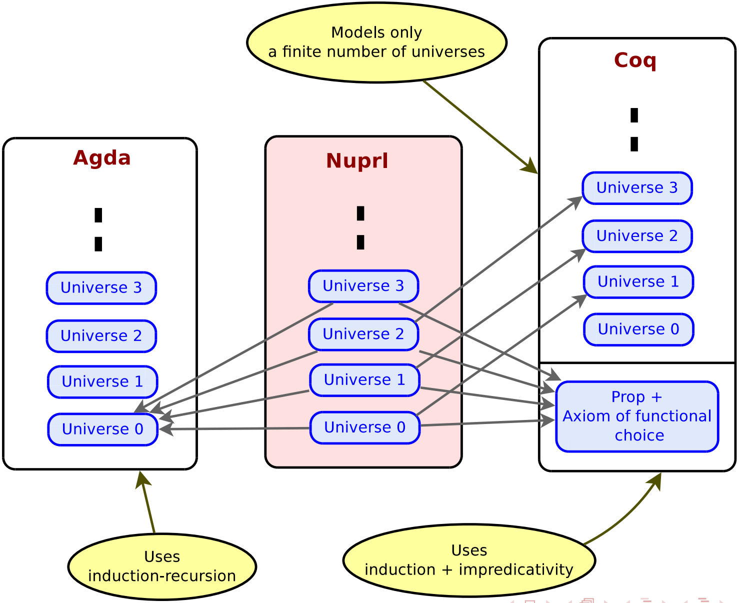
$$H \vdash a{:}A \rightarrow B[a] \; \lfloor \mathbf{ext}\ \lambda x.b \rfloor$$
$$\text{BY } [\texttt{lambdaFormation}]$$
$$H, x : A \vdash B[x] \; \lfloor \mathbf{ext}\ b \rfloor$$
$$H \vdash A \in \mathbb{U}_i \; \lfloor \mathbf{ext}\ \star \rfloor$$

Nuprl PER Semantics Implemented in Coq



Stuart Allen had his own meta-theory that was meant to be meaningful on its own and needs not be framed into type theory. We chose to use Coq and Agda.

Nuprl PER Semantics Implemented in Coq



The More Inference Rules the Better!

All verified

Expose more of the metatheory

Encode Mathematical knowledge

Intuitionistic Type Theory

We've proved these rules correct using our Coq model:

Brouwer's Continuity Principle for numbers

$$\mathbf{\Pi} F{:}\mathcal{B} \rightarrow \mathbb{N}.\mathbf{\Pi} f{:}\mathcal{B}.\downharpoonleft \mathbf{\Sigma} n{:}\mathbb{N}.\mathbf{\Pi} g{:}\mathcal{B}.f =_{\mathbb{N}^{\mathbb{N}_n}} g \rightarrow F(f) =_{\mathbb{N}} F(g)$$

$$(\mathcal{B} = \mathbb{N}^{\mathbb{N}} = \mathbb{N} \rightarrow \mathbb{N})$$

Bar induction

- On free choice sequences of closed terms without atoms
- We can build indexed W types

Weak Continuity

False in Nuprl (following Escardó and Xu)

$$\Pi F{:}\mathcal{B} \to \mathbb{N}.\Pi f{:}\mathcal{B}.\Sigma n{:}\mathbb{N}.\Pi g{:}\mathcal{B}.f =_{\mathbb{N}^{\mathbb{N}_n}} g \to F(f) =_{\mathbb{N}} F(g)$$

Easy in Coq model (almost purely by computation) because it doesn't have computational content

$$\Pi F{:}\mathcal{B} \to \mathbb{N}.\Pi f{:}\mathcal{B}.\downarrow\Sigma n{:}\mathbb{N}.\Pi g{:}\mathcal{B}.f =_{\mathbb{N}^{\mathbb{N}_n}} g \to F(f) =_{\mathbb{N}} F(g)$$

Harder in Coq because it has computational content: uses named exceptions + ν (following Longley's method)

$$\Pi F{:}\mathcal{B} \to \mathbb{N}.\Pi f{:}\mathcal{B}.\downarrow\Sigma n{:}\mathbb{N}.\Pi g{:}\mathcal{B}.f =_{\mathbb{N}^{\mathbb{N}_n}} g \to F(f) =_{\mathbb{N}} F(g)$$

Strong Continuity

Actually what we proved in Coq is essentially

$$
\begin{array}{l}
\mathbf{\Pi} F{:}\mathcal{B} \to \mathbb{N}. \\
\quad \downharpoonleft \mathbf{\Sigma} M{:}(\mathbf{\Pi} n{:}\mathbb{N}.\mathbb{N}^{\mathbb{N}_n} \to \mathbb{N}+\mathtt{Unit}). \\
\qquad \mathbf{\Pi} f{:}\mathcal{B}.\mathbf{\Sigma} n{:}\mathbb{N}. \quad M\ n\ f =_{\mathbb{N}+\mathtt{Unit}} \mathtt{inl}(F(f)) \\
\qquad\qquad\qquad\qquad \wedge\ \mathbf{\Pi} m{:}\mathbb{N}.\mathtt{isl}(M\ m\ f) \to m =_{\mathbb{N}} n
\end{array}
$$

which is equivalent to weak continuity because (standard)

$$\mathsf{AC}_{1,0\downharpoonleft} \Rightarrow (\mathsf{WCP}_{\downharpoonleft} \iff \mathsf{SCP}_{\downharpoonleft})$$

Axiom of Choice

Trivial

$$\Pi a{:}A.\Sigma b{:}B.P\ a\ b \Rightarrow \Sigma f{:}B^{A}.\Pi a{:}A.P\ a\ f(a)$$

Harder to prove ($AC_{0,0}$) in Coq: uses the axiom of choice and free choice sequences

$$\Pi a{:}\mathbb{N}.\downarrow\Sigma b{:}\mathbb{N}.P\ a\ b \Rightarrow \downarrow\Sigma f{:}\mathbb{N}^{\mathbb{N}}.\Pi a{:}\mathbb{N}.P\ a\ f(a)$$

Non-trivial to prove ($AC_{0,n}$ and $AC_{1,n}$) in Nuprl

$$\Pi a{:}\mathbb{N}.\downarrow\Sigma b{:}B.P\ a\ b \Rightarrow \downarrow\Sigma f{:}B^{\mathbb{N}}.\Pi a{:}\mathbb{N}.P\ a\ f(a)$$

$$\Pi a{:}\mathcal{B}.\downarrow\Sigma b{:}B.P\ a\ b \Rightarrow \downarrow\Sigma f{:}B^{\mathcal{B}}.\Pi a{:}\mathcal{B}.P\ a\ f(a)$$

Uniform Continuity

Follows from the Fan Theorem (every decidable bar is uniform) and Weak Continuity (standard)

$$\mathbf{\Pi} F{:}\mathcal{C} \to \mathbb{N}.\downharpoonleft \mathbf{\Sigma} n{:}\mathbb{N}.\mathbf{\Pi} f, g{:}\mathcal{C}.f =_{2^{\mathbb{N}_n}} g \to F(f) =_{\mathbb{N}} F(g)$$

$$(\mathcal{C} = 2^{\mathbb{N}})$$

Following Escardó and Xu:

$$\mathbf{\Pi} F{:}\mathcal{C} \to \mathbb{N}.\mathbf{\Sigma} n{:}\mathbb{N}.\mathbf{\Pi} f, g{:}\mathcal{C}.f =_{2^{\mathbb{N}_n}} g \to F(f) =_{\mathbb{N}} F(g)$$

Bar Induction

Fan Theorem follows from Bar Induction on Decidable Bars (BID)

$H \vdash \downarrow(X\ 0\ c)$

BY [BID]

$(\mathtt{dec})\quad H, n : \mathbb{N}, s : \mathbb{N}^{\mathbb{N}_n} \vdash B\ n\ s\ \vee\ \neg B\ n\ s$

$(\mathtt{bar})\quad H, s : \mathbb{N}^{\mathbb{N}} \vdash \downarrow\exists n : \mathbb{N}.\ B\ n\ s$

$(\mathtt{imp})\quad H, n : \mathbb{N}, s : \mathbb{N}^{\mathbb{N}_n}, m : B\ n\ s \vdash X\ n\ s$

$(\mathtt{ind})\quad H, n : \mathbb{N}, s : \mathbb{N}^{\mathbb{N}_n}, x : (\forall m : \mathbb{N}.\ X\ (n+1)\ \mathtt{ext}(s, n, m)) \vdash X\ n\ s$

Bar Induction

We proved BID for free choice sequences of numbers in Coq following Dummett's "standard" classical proof (easy)

We added free choice sequences of numbers to Nuprl's model: all Coq functions from $\mathbb{N}$ to $\mathbb{N}$

What about sequences of terms?

Bar Induction

We proved BID for free choice sequences of closed terms without names (in Coq following Dummett's "standard" classical proof)

Harder because we had to turn our terms into a big W type: a function from $\mathbb{N}$ to terms is now a term!

Why without names?

ν picks fresh names and we can't compute the collection of all names anymore (still doable I think)

Law of Excluded Middle

LEM is false in Nuprl (Anand)

$$\mathbf{\Pi} P{:}\mathbb{P}.P \ \vee \ \neg P$$

Follows from: $\neg\mathbf{\Pi} t{:}\mathtt{Base}.t\Downarrow \ \vee \neg t\Downarrow$ (call the function magic)
We can prove:
$\mathtt{if}\ \mathit{magic}(\bot)\ \mathtt{then}\ \bot\ \mathtt{else}\ \star \preccurlyeq \mathtt{if}\ \mathit{magic}(\star)\ \mathtt{then}\ \bot\ \mathtt{else}\ \star$
We get: $\star \preccurlyeq \bot$

Squashed version is true in Coq (using LEM in Coq)

$$\mathbf{\Pi} P{:}\mathbb{P}.\downarrow(P \ \vee \ \neg P)$$

Questions

Can we prove continuity for sequences of terms instead of $\mathcal{B}$?

Can we prove BID/BIM on sequences of terms with atoms?

What does that give us? + proof-theoretic strength?

Can I hope to be able to prove BID in Coq/Agda without LEM/AC?