Proven-Correct Provers

Vincent Rahli

(in collaboration with Mark Bickford (Cornell), Robert L. Constable (Cornell), Liron Cohen (Cornell), Ivana Vukotic (SnT), Marcus Völp (SnT), Brandon Bohrer (CMU), Ándre Platzer (CMU))

May 22, 2017

What are we going to cover?

Turning Nuprl into an Intuitionistic Type Theory

- Verified validity of inference rules
- Added Intuitionistic axioms (continuity and bar induction)
- Added named exception
- Added some sort of choice sequences

Verification of KeYmaera X's core

- Verified validity of inference rules
- Built a proof checker in Coq
- Enhanced a real analysis library

Nupirl?

Nuprl in a Nutshell

Similar to Coq and Agda

Extensional Constructive Type Theory with partial functions

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

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

Extensional CTT with partial functions?

Extensional

$$(\forall a : A.\ f(a) = g(a) \in B) \rightarrow f = g \in A \rightarrow B$$

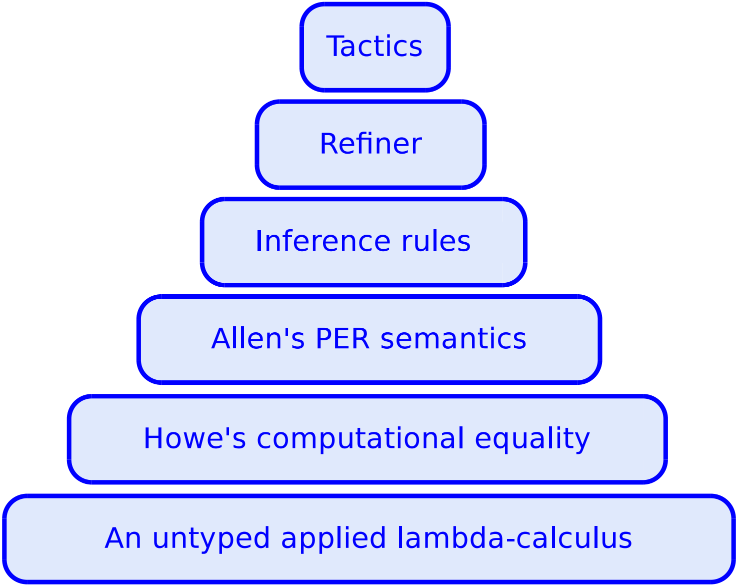
Constructive

$(A \rightarrow A)$ true because inhabited by $(\lambda x.x)$

Partial functions

$\texttt{fix}(\lambda x.x)$ inhabits $\overline{\mathbb{N}}$

Nuprl Stack


Tactics
Refiner
Inference rules
Allen's PER semantics
Howe's computational equality
An untyped applied lambda-calculus

Nuprl Types—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: $\mathbb{U}_i$

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

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

Nupl Types—PER type (inspired by Allen)

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

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

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

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

Nuprl Types—Squashing

Proof erasure (1):

$$\{\texttt{Unit} \mid T\}$$

$\downarrow T$ $\quad$ $\texttt{per}(\lambda x.\lambda y.\star \preccurlyeq x \sqcap \star \preccurlyeq y \sqcap T)$

$$\texttt{Img}(T, \lambda\_.\star)$$

Proof irrelevance:

$\downarrow T$ $\quad$ $T//\texttt{True}$ $\quad$ $\texttt{per}(\lambda x.\lambda y.x \in T \sqcap y \in T)$

Proof erasure (2):

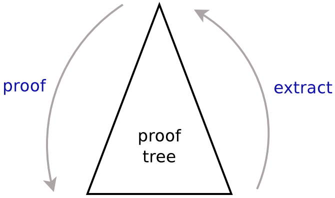
$\Downarrow T$ $\quad$ $\texttt{Top}//T$ $\quad$ $\texttt{per}(\lambda\_.\lambda\_.T)$

Nuprl Refinements

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

A generic goal directed reasoner:

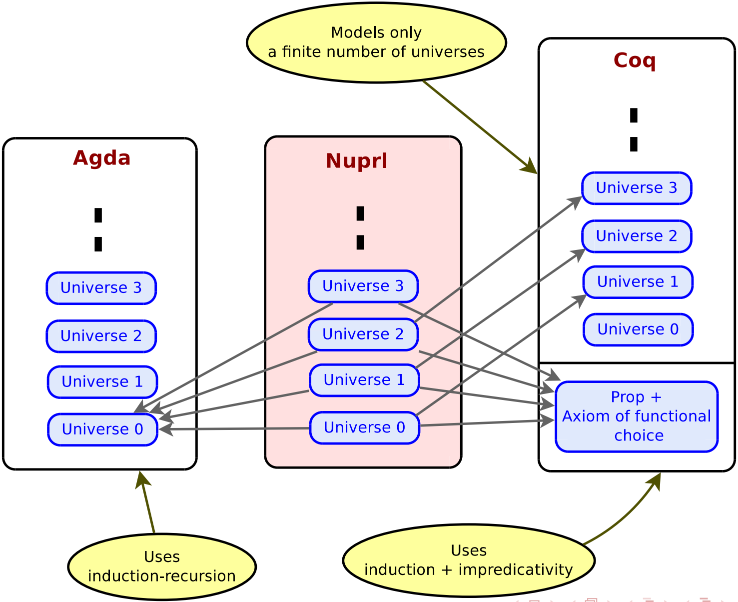
- **a rule interpreter**
- **a proof manager**



Example of a rule

$$
\begin{array}{l}
H \vdash a{:}A \rightarrow B[a] \ \lfloor \textbf{ext} \ \lambda x.b \rfloor \\
\quad \text{BY } \texttt{[lambdaFormation]} \\
\quad H, x : A \vdash B[x] \ \lfloor \textbf{ext} \ b \rfloor \\
\quad H \vdash A \in \mathbb{U}_i \ \lfloor \textbf{ext} \ \star \rfloor
\end{array}
$$

Nuprl PER Semantics Implemented in Coq



The More Inference Rules the Better!

All verified

Expose more of the metatheory

Encode Mathematical knowledge

Let's now see how far we got towards turning Nuprl into an intuitionistic type theory

Intuitionism

- First act: Intuitionistic logic is based on our **inner consciousness of time**, which gives rise to the **two-ity**.
- As opposed to Platonism, it's about **constructions in the mind** and not objects that exist independently of us. There are no mathematical truths outside human thought.
- A statement is true when we have an appropriate construction, and false when no construction is possible.

Intuitionism

- Second act: New mathematical entities can be created through **more or less freely proceeding sequences** of mathematical entities.
- Also by defining new mathematical species (types, sets) that respect equality of mathematical entities.
- Gives rise to (never finished) choice sequences. Could be lawlike or lawless. Laws can be 1st order, 2nd order...
- The continuum is captured by choice sequences of nested rational intervals.

Intuitionism—The creative subject

Brouwer introduced procedures that depend on the mental activity of an idealized mathematician

$$\mathrm{CS}_1 \qquad \forall x.(\vdash_x A \ \vee\ \neg \vdash_x A)$$

$$\mathrm{CS}_2 \qquad \forall x, y.(\vdash_x A \ \Rightarrow \vdash_{x+y} A)$$

$$\mathrm{CS}_3 \qquad (\exists x. \vdash_n A) \iff A$$

Intuitionism—a non-classical logic

1. Take p a predicate on numbers such that $p(n)$ is decidable for all n but $(\forall n : \mathbb{N}.\ p(n))$ is not known, e.g., GC.
2. Define the choice sequence α (real number) as follows:

$\alpha(0) = 2^{-0}$	$\alpha(1) = 2^{-1}$	$\alpha(2) = 2^{-2}$	$\alpha(3) = 2^{-3}$	$\alpha(4) = 2^{-4}$	$\alpha(5) = 2^{-4}$	$\alpha(6) = 2^{-4}$	$\alpha(7) = 2^{-4}$	$\cdots$
$p(0)$	$p(1)$	$p(2)$	$p(3)$	$p(4)$	$\neg p(5)$	_	_	

3. We have $\alpha = 0 \iff \forall n : \mathbb{N}.\ p(n)$
4. Therefore, $\alpha = 0$ is not decidable

Intuitionism—lawless sequences

"Absolutely free choice sequences"—think of the 2nd order restriction that forbids 1st order restrictions

We'll write s for finite sequences and α for lawless sequences.
We write $\alpha \in s$ if s is an initial segment of α.
$\equiv$ stands for intensional equality.
We write $\overline{\alpha}x$ for the initial segment of α of length x.

LS_1	$\forall s.\exists \alpha.\alpha \in s$
LS_2	$\forall \alpha, \beta.(\alpha \equiv \beta \ \vee \ \neg \alpha \equiv \beta)$
LS_3	$A(\alpha) \ \Rightarrow \ \exists x.\forall \beta.(\overline{\alpha}x = \overline{\beta}x \ \Rightarrow \ A(\beta))$

Intuitionism—continuity

What can we do with these sequences if they are never finished?

Brouwer's answer: one never needs the whole sequence.

His **continuity axiom for numbers** says that functions from sequences to numbers only need initial segments

$$\forall F : \mathbb{N}^{\mathcal{B}}.\ \forall f : \mathcal{B}.\ \exists n : \mathbb{N}.\ \forall g : \mathcal{B}.\ f =_{\mathcal{B}_n} g \rightarrow F(f) =_{\mathbb{N}} F(g)$$

From which his **uniform continuity theorem** follows: Let f be of type $[\alpha, \beta] \rightarrow \mathbb{R}$, then

$$\begin{aligned}&\mathrm{CONT}(f, \alpha, \beta)\\ &= \forall \epsilon > 0. \exists \delta > 0. \forall x, y : [\alpha, \beta].\ |x - y| \leqslant \delta \rightarrow |f(x) - f(y)| \leqslant \epsilon\end{aligned}$$

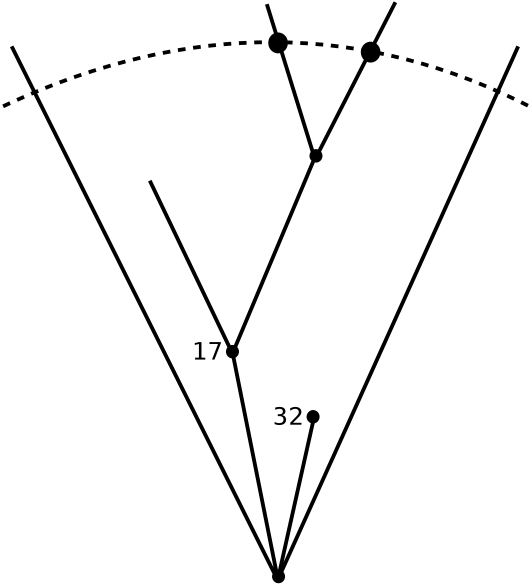
Intuitionism—bar induction

To prove his **uniform continuity theorem**, Brouwer also used the **Fan theorem**.

The fan theorem says that if for each branch α of a binary tree T, a property A is true about some initial segment of α, then **there is a uniform bound** on the depth at which A is met.

The fan theorem follows from **bar induction**.

Bar Induction—the intuition



Bar Induction—on decidable bars

$H \vdash P(0, c)$

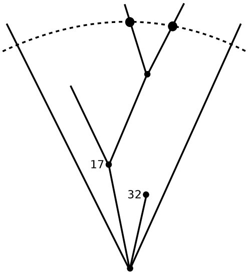
BY [BID]

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

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

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

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

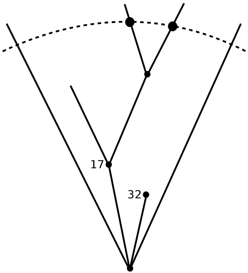


Bar Induction—on monotone bars

$H \vdash \downharpoonleft P(0, c)$

BY [BIM]

$$
\begin{array}{ll}
(\texttt{mon}) & H, n : \mathbb{N}, s : \mathbb{N}^{\mathbb{N}_n} \vdash \forall m : \mathbb{N}.\ B(n, s) \ \Rightarrow\ B(n+1, s \oplus_n m) \\
(\texttt{bar}) & H, s : \mathbb{N}^{\mathbb{N}} \vdash \downharpoonleft \exists n : \mathbb{N}.\ B(n, s) \\
(\texttt{imp}) & H, n : \mathbb{N}, s : \mathbb{N}^{\mathbb{N}_n}, m : B(n, s) \vdash P(n, s) \\
(\texttt{ind}) & H, n : \mathbb{N}, s : \mathbb{N}^{\mathbb{N}_n}, x : (\forall m : \mathbb{N}.\ P((n+1), s \oplus_n m)) \vdash P(n, s)
\end{array}
$$



Why the squashing operator?

As proved by Kreisel, Troelstra, and Escardó and Xu,
continuity is false in Martin-Löf-like type theories
when not $\downarrow$-squashed

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

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

From which we derived:
BIM is false when not $\downarrow$-squashed

Bar Induction

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

We added "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/BIM for sequences of closed terms without names (in Coq following "standard" classical proof)

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

Why without names?

ν picks fresh names and we can't compute the collection of all names anymore

Questions

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

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

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

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

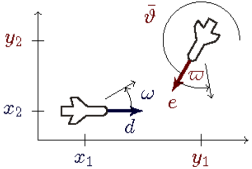
What Axioms Have We Validated So Far?

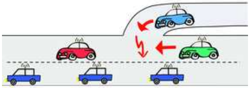
Name	Formula	Where	Comments
$\mathtt{WCP}_{1,0}$	$\neg\mathbf{\Pi}F{:}\mathbb{N}^{\mathcal{B}}.\mathbf{\Pi}f{:}\mathcal{B}.\mathbf{\Sigma}n{:}\mathbb{N}.\mathbf{\Pi}g{:}\mathcal{B}.f =_{\mathcal{B}_n} g \rightarrow F(f) =_{\mathbb{N}} F(g)$	Nuprl	
$\mathtt{WCP}_{1,0_{\downharpoonleft}}$	$\mathbf{\Pi}F{:}\mathbb{N}^{\mathcal{B}}.\mathbf{\Pi}f{:}\mathcal{B}.\downharpoonleft\mathbf{\Sigma}n{:}\mathbb{N}.\ \mathbf{\Pi}g{:}\mathcal{B}.f =_{\mathcal{B}_n} g \rightarrow F(f) =_{\mathbb{N}} F(g)$	Coq	uses named exceptions
$\mathtt{WCP}_{1,0_{\downarrow}}$	$\mathbf{\Pi}F{:}\mathbb{N}^{\mathcal{B}}.\mathbf{\Pi}f{:}\mathcal{B}.\downarrow\mathbf{\Sigma}n{:}\mathbb{N}.\mathbf{\Pi}g{:}\mathcal{B}.f =_{\mathcal{B}_n} g \rightarrow F(f) =_{\mathbb{N}} F(g)$	Coq	uses $\perp$
$\mathtt{WCP}_{1,1}$	$\neg\mathbf{\Pi}P{:}\mathcal{B} \rightarrow \mathbb{P}^{\mathcal{B}}.(\mathbf{\Pi}a{:}\mathcal{B}.\mathbf{\Sigma}b{:}\mathcal{B}.P(a,b)) \rightarrow \mathbf{\Sigma}c{:}\mathbb{N}^{\mathcal{B}}.\mathtt{CONT}(c) \ \wedge\ \mathbf{\Pi}a{:}\mathcal{B}.\mathtt{shift}(c,a)$	Nuprl	
$\mathtt{WCP}_{1,1_{\downharpoonleft}}$	$\overset{?}{\neg}\mathbf{\Pi}P{:}\mathcal{B} \rightarrow \mathbb{P}^{\mathcal{B}}.(\mathbf{\Pi}a{:}\mathcal{B}.\mathbf{\Sigma}b{:}\mathcal{B}.P(a,b)) \rightarrow \downharpoonleft\mathbf{\Sigma}c{:}\mathbb{N}^{\mathcal{B}}.\ \mathtt{CONT}(c)_{\downharpoonleft} \ \wedge\ \mathbf{\Pi}a{:}\mathcal{B}.\mathtt{shift}(c,a)$	?	
$\mathtt{WCP}_{1,1_{\downarrow}}$	$\overset{?}{\neg}\mathbf{\Pi}P{:}\mathcal{B} \rightarrow \mathbb{P}^{\mathcal{B}}.(\mathbf{\Pi}a{:}\mathcal{B}.\mathbf{\Sigma}b{:}\mathcal{B}.P(a,b)) \rightarrow \downarrow\mathbf{\Sigma}c{:}\mathbb{N}^{\mathcal{B}}.\mathtt{CONT}(c)_{\downarrow} \ \wedge\ \mathbf{\Pi}a{:}\mathcal{B}.\mathtt{shift}(c,a)$	?	
$\mathtt{AC}_{0,0}$	$\mathbf{\Pi}P{:}\mathbb{N} \rightarrow \mathbb{P}^{\mathbb{N}}.(\mathbf{\Pi}n{:}\mathbb{N}.\mathbf{\Sigma}m{:}\mathbb{N}.P(n,m)) \rightarrow \mathbf{\Sigma}f{:}\mathcal{B}.\mathbf{\Pi}n{:}\mathcal{B}.P(n,f(n))$	Nuprl	
$\mathtt{AC}_{0,0_{\downharpoonleft}}$	$\mathbf{\Pi}P{:}\mathbb{N} \rightarrow \mathbb{P}^{\mathbb{N}}.(\mathbf{\Pi}n{:}\mathbb{N}.\downharpoonleft\mathbf{\Sigma}m{:}\mathbb{N}.\ P(n,m)) \rightarrow \downharpoonleft\mathbf{\Sigma}f{:}\mathcal{B}.\ \mathbf{\Pi}n{:}\mathcal{B}.P(n,f(n))$	Nuprl	
$\mathtt{AC}_{0,0_{\downarrow}}$	$\mathbf{\Pi}P{:}\mathbb{N} \rightarrow \mathbb{P}^{\mathbb{N}}.(\mathbf{\Pi}n{:}\mathbb{N}.\downarrow\mathbf{\Sigma}m{:}\mathbb{N}.P(n,m)) \rightarrow \downarrow\mathbf{\Sigma}f{:}\mathcal{B}.\mathbf{\Pi}n{:}\mathcal{B}.P(n,f(n))$	Coq	uses classical logic
$\mathtt{AC}_{1,0}$	$\mathbf{\Pi}P{:}\mathcal{B} \rightarrow \mathbb{P}^{\mathbb{N}}.(\mathbf{\Pi}f{:}\mathcal{B}.\mathbf{\Sigma}n{:}\mathbb{N}.P(f,n)) \rightarrow \mathbf{\Sigma}F{:}\mathbb{N}^{\mathcal{B}}.\mathbf{\Pi}f{:}\mathcal{B}.P(f,F(f))$	Nuprl	
$\mathtt{AC}_{1,0_{\downharpoonleft}}$	$\mathbf{\Pi}P{:}\mathcal{B} \rightarrow \mathbb{P}^{\mathbb{N}}.(\mathbf{\Pi}f{:}\mathcal{B}.\downharpoonleft\mathbf{\Sigma}n{:}\mathbb{N}.\ P(f,n)) \rightarrow \downharpoonleft\mathbf{\Sigma}F{:}\mathbb{N}^{\mathcal{B}}.\ \mathbf{\Pi}f{:}\mathcal{B}.P(f,F(f))$	Nuprl	
$\mathtt{AC}_{1,0_{\downarrow}}$	$\overset{?}{\neg}\mathbf{\Pi}P{:}\mathcal{B} \rightarrow \mathbb{P}^{\mathbb{N}}.(\mathbf{\Pi}f{:}\mathcal{B}.\downarrow\mathbf{\Sigma}n{:}\mathbb{N}.P(f,n)) \rightarrow \downarrow\mathbf{\Sigma}F{:}\mathbb{N}^{\mathcal{B}}.\mathbf{\Pi}f{:}\mathcal{B}.P(f,F(f))$	?	
$\mathtt{AC}_{2,0}$	$\mathbf{\Pi}P{:}\mathbb{N}^{\mathcal{B}} \rightarrow \mathbb{P}^{\mathbb{N}}.(\mathbf{\Pi}f{:}\mathbb{N}^{\mathcal{B}}.\mathbf{\Sigma}n{:}T.P(f,n)) \rightarrow \mathbf{\Sigma}F{:}T^{(\mathbb{N}^{\mathcal{B}})}.\mathbf{\Pi}f{:}\mathbb{N}^{\mathcal{B}}.P(f,F(f))$	Nuprl	
$\mathtt{AC}_{2,0_{\downharpoonleft}}$	$\neg(\mathbf{\Pi}P{:}\mathbb{N}^{\mathcal{B}} \rightarrow \mathbb{P}^{T}.(\mathbf{\Pi}f{:}\mathbb{N}^{\mathcal{B}}.\downharpoonleft\mathbf{\Sigma}n{:}T.\ P(f,n)) \rightarrow \downharpoonleft\mathbf{\Sigma}F{:}T^{(\mathbb{N}^{\mathcal{B}})}.\ \mathbf{\Pi}f{:}\mathbb{N}^{\mathcal{B}}.P(f,F(f)))$	Nuprl	contradicts continuity
$\mathtt{AC}_{2,0_{\downarrow}}$	$\neg(\mathbf{\Pi}P{:}\mathbb{N}^{\mathcal{B}} \rightarrow \mathbb{P}^{T}.(\mathbf{\Pi}f{:}\mathbb{N}^{\mathcal{B}}.\downharpoonleft\mathbf{\Sigma}n{:}T.\ P(f,n)) \rightarrow \downarrow\mathbf{\Sigma}F{:}T^{(\mathbb{N}^{\mathcal{B}})}.\mathbf{\Pi}f{:}\mathbb{N}^{\mathcal{B}}.P(f,F(f)))$	Nuprl	contradicts continuity
LEM	$\neg\mathbf{\Pi}P{:}\mathbb{P}.P \ \vee\ \neg P$	Nuprl	
$\mathrm{LEM}_{\downharpoonleft}$	$\neg\mathbf{\Pi}P{:}\mathbb{P}.\downharpoonleft(P \ \vee\ \neg P)$	Nuprl	
$\mathrm{LEM}_{\downarrow}$	$\mathbf{\Pi}P{:}\mathbb{P}.\downarrow(P \ \vee\ \neg P)$	Coq	uses classical logic
MP	$\mathbf{\Pi}P{:}\mathbb{P}^{\mathbb{N}}.(\mathbf{\Pi}n{:}\mathbb{N}.P(n) \ \vee\ \neg P(n)) \rightarrow (\neg\mathbf{\Pi}n{:}\mathbb{N}.\neg P(n)) \rightarrow \mathbf{\Sigma}n{:}\mathbb{N}.P(n)$	Nuprl	uses $\mathrm{LEM}_{\downarrow}$
KS	$\neg\mathbf{\Pi}A{:}\mathbb{P}.\mathbf{\Sigma}a{:}\mathcal{B}.((\mathbf{\Sigma}x{:}\mathbb{N}.a(x) =_{\mathbb{N}} 1) \iff A)$	Nuprl	uses MP
$\mathrm{KS}_{\downharpoonleft}$	$\neg\mathbf{\Pi}A{:}\mathbb{P}.\downharpoonleft\mathbf{\Sigma}a{:}\mathcal{B}.((\mathbf{\Sigma}x{:}\mathbb{N}.a(x) =_{\mathbb{N}} 1) \iff A)$	Nuprl	uses MP
$\mathrm{KS}_{\downarrow}$	$\mathbf{\Pi}A{:}\mathbb{P}.\downarrow\mathbf{\Sigma}a{:}\mathcal{B}.((\mathbf{\Sigma}x{:}\mathbb{N}.a(x) =_{\mathbb{N}} 1) \iff A)$	Coq	uses classical logic
$\mathrm{BI}_{\downarrow}$	$\mathtt{WF}(B) \rightarrow \mathtt{BAR}_{\downharpoonleft}(B) \rightarrow \mathtt{BASE}(B,P) \rightarrow \mathtt{IND}(P) \rightarrow \downarrow P(0, \perp\!\!\!\perp)$	Coq	uses classical logic
BID	$\mathtt{WF}(B) \rightarrow \mathtt{BAR}_{\downharpoonleft}(B) \rightarrow \mathtt{DEC}(B) \rightarrow \mathtt{BASE}(B,P) \rightarrow \mathtt{IND}(P) \rightarrow P(0, \perp\!\!\!\perp)$	Nuprl	uses $\mathrm{BI}_{\downarrow}$
$\mathrm{BIM}_{\downharpoonleft}$	$\mathtt{WF}(B) \rightarrow \mathtt{BAR}_{\downharpoonleft}(B) \rightarrow \mathtt{MON}(B) \rightarrow \mathtt{BASE}(B,P) \rightarrow \mathtt{IND}(P) \rightarrow \downharpoonleft P(0, \perp\!\!\!\perp)$	Nuprl	uses $\mathrm{BI}_{\downarrow}$
BIM	$\neg\mathbf{\Pi}B,P{:}(\mathbf{\Pi}n{:}\mathbb{N}.\mathbb{P}^{\mathcal{B}_n}).\mathtt{BAR}_{\downharpoonleft}(B) \rightarrow \mathtt{MON}(B) \rightarrow \mathtt{BASE}(B,P) \rightarrow \mathtt{IND}(P) \rightarrow P(0, \perp\!\!\!\perp)$	Nuprl	contradicts continuity

We verified the core of another prover: KeYmaera X

(some of that material comes from http://symbolaris.com/)
(thanks to Ivana for some of the material)

KeYmaera—a theorem prover for hybrid systems

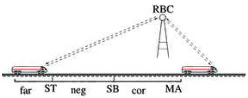


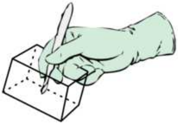


CPSs combine digital and physical components

Hybrid systems model discrete and continuous effects of CPSs

Combination of computation and control





KeYmaera—a theorem prover for hybrid systems

discrete dynamics specified using assignments

$$a := 1$$

(set acceleration to 1)

continuous dynamic specified using differential equations

$$x' = v, v' = a$$

(derivative of position = velocity,
derivative of velocity = acceleration)

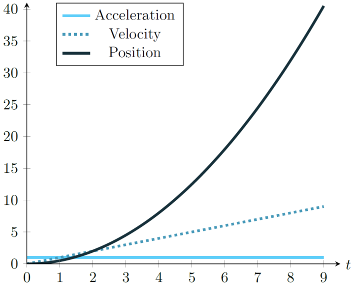
KeYmaera—example 1

Example 1 Safety property of an uncontrolled continuous car model

$$\text{init} \;\rightarrow\; [plant]\;(\text{req}) \tag{5}$$
$$\text{init} \equiv v \geq 0 \wedge A > 0 \tag{6}$$
$$plant \equiv p' = v, v' = A \tag{7}$$
$$\text{req} \equiv v \geq 0 \tag{8}$$

alter-native

$$\text{init} \equiv v \geq 0 \wedge A > 0 \wedge p_0 = p \tag{9}$$
$$\text{req} \equiv p \geq p_0 \tag{10}$$



KeYmaera—example 2

Example 2 Safety property of a hybrid car model

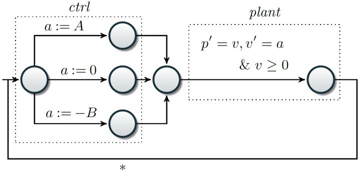
$$\text{init} \rightarrow [(\mathit{ctrl}; \mathit{plant})^*]\,(\text{req}) \tag{11}$$

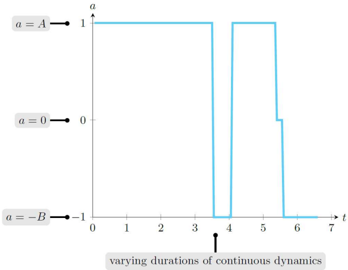
$$\text{init} \equiv v \geq 0 \land A > 0 \land \boldsymbol{B > 0} \tag{12}$$

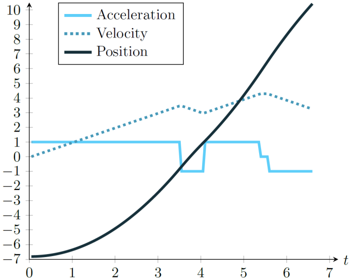
$$\boldsymbol{\mathit{ctrl} \equiv a := A \cup a := 0 \cup a := -B} \tag{13}$$

$$\mathit{plant} \equiv p' = v, v' = a \;\boldsymbol{\&\; v \geq 0} \tag{14}$$

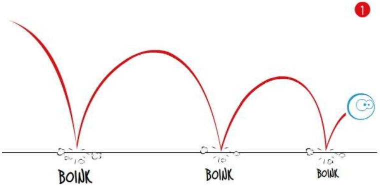
$$\text{req} \equiv v \geq 0 \tag{15}$$







KeYmaera—example 3



$$(h' = v, v' = -g \& h \geqslant 0; \texttt{if}\ (h = 0)\ \texttt{then}\ v := -cv\ \texttt{fi})*$$

(g: gravity force; c: damping factor)

KeYmaera—verified cores

Core implemented in Scala

- dL
- Axioms
- Uniform substitution
- Renaming

Verified using Coq and Isabelle

KeYmaera—uniform substitution?

concrete axioms instead of schemata

$$[x := e]p(x) \iff p(e)$$

instantiated using substitutions

$$[x := e]x \geqslant 0 \iff e \geqslant 0$$

all side conditions are handled by a
uniform admissibility condition on substitutions

e.g., x shouldn't get captured

KeYmaera—verified cores

We formalized in Coq:

- dL's syntax
- dL's static and dynamic semantics
- dL's axioms
- uniform substitution
- renaming
- proof checker

Brandon Bohrer found a bug
in the implementation of renaming!

KeYmaera—Picard-Lindelöf

1 gap: The real analysis library we used doesn't provide the Picard-Lindelöf theorem

Existence and uniqueness of solutions to first-order equations with initial condition

$$\begin{cases} y'(t) = f(t, y(t)) \\ y(0) = y_0 \end{cases}$$

where f is Lipschitz continuous

KeYmaera—Faà di Bruno

We thought we would have to compute the n^{th}-derivatives of standard operations, such as **composition**

Chain rule for 1st derivative:

$$(f \circ g)' = (f' \circ g) \cdot g'$$

Faà di Bruno generalizes the chain rule (1855–1857):

$$(f \circ g)^{(n)} = \sum_{n=1k_1+\cdots+nk_n} \frac{n!}{k_1! \cdots k_n!} \cdot f^{(k_1+\cdots+k_n)}(g) \cdot \prod_{j=1}^{n} (\frac{g^{(j)}}{j!})^{k_j}$$

KeYmaera—Faà di Bruno

Faà di Bruno generalizes the chain rule:

$$(f \circ g)^{(n)} = \sum_{n=1k_1+\cdots+nk_n} \frac{n!}{k_1!\cdots k_n!} \cdot f^{(k_1+\cdots+k_n)}(g) \cdot \prod_{j=1}^{n} (\frac{g^{(j)}}{j!})^{k_j}$$

So far, we only proved **McKiernan's formula**
(see: "On the nth Derivative of Composite Functions"):

$$(f \circ g)^{(n)} = \sum_{r=1}^{n} f^{(r)}(g) \cdot \sum_{s=0}^{r} \frac{(-1)^{r-s}}{s!(r-s)!} g^{r-s}(g^s)^{(n)}$$

KeYmaera—Faà di Bruno

Many hand-written proofs

See Johnson's "The Curious History of Faà di Bruno's Formula"

Formally verified?