

About Formalisations which use the Nominal Datatype Package

Christian Urban, TU Munich

Some Formalisations

The nominal datatype package has been developed to reason about syntax with binders. Several formalisations have already been done:

- CR and SN in the lambda-calculus; various classic results in SOS
- Jesper Bengtson formalised a significant part of the pi-calculus theory
- a chapter by Karl Cray in the Advanced Type-Systems book; this chapter presents a completeness proof for equivalence checking...

Two Health Warnings

I found that theorem provers should come with two health warnings:

- Theorem provers are addictive!

(Xavier Leroy: "Building [proof] scripts is surprisingly addictive, in a videogame kind of way...")

Two Health Warnings

I found that theorem provers should come with two health warnings:

- Theorem provers are addictive!

(Xavier Leroy: "Building [proof] scripts is surprisingly addictive, in a videogame kind of way...")

- Theorem provers cause you to lose faith in your proofs done by hand!

(Michael Norrish, Mike Gordon, me, very possibly others)

Two Health Warnings

I found that theorem provers should come with two health warnings:

- Theorem provers are addictive!

(Xavier Leroy: "Building [proof] scripts is surprisingly addictive, in a videogame kind of way...")

- Theorem provers cause you to lose faith in your proofs done by hand!

(Michael Norrish, Mike Gordon, me, very possibly others)

Question: should we trust our informal reasoning in the first place when proofs are about syntax?

My Answer: No!

- I formalised my PhD about a strong normalisation result for cut-elimination

$$\begin{aligned} \text{trm} = & Ax \text{ name coname} \\ & | \text{Cut } \langle\langle \text{coname} \rangle\rangle \text{trm } \langle\langle \text{name} \rangle\rangle \text{trm} \\ & | \text{And}_R \langle\langle \text{coname} \rangle\rangle \text{trm } \langle\langle \text{coname} \rangle\rangle \text{trm } \text{coname} \\ & | \text{And}_L^i \langle\langle \text{name} \rangle\rangle \text{trm } \text{name} \\ & \dots \\ & | \text{Imp}_L \langle\langle \text{coname} \rangle\rangle \text{trm } \langle\langle \text{name} \rangle\rangle \text{trm } \text{name} \\ & | \text{Imp}_R \langle\langle \text{coname} \rangle\rangle \langle\langle \text{name} \rangle\rangle \text{trm } \text{coname} \end{aligned}$$

- it has two kinds of binders; substitution is a form of cut-elimination
- it is SN, but one central lemma was wrong; as a result another had to be generalised; definitions had to be restated.

Substitution Lemma: If $x \neq y$ and $x \notin FV(L)$, then

$$M[x := N][y := L] \equiv M[y := L][x := N[y := L]].$$

Proof: By induction on the structure of M .

- **Case 1:** M is a variable.

Case 1.1 $M = x$. Then both sides equal $N[y := L]$ since $x \neq y$.

Case 1.2 $M = y$. Then both sides equal L .

Case 1.3 $M = z$. Then both sides equal z .

- **Case 2:** $M \equiv \lambda z.M_1$. We need to show that $z \neq x, y$ and z is not free in N, L . Then by induction hypothesis

$$(\lambda z.M_1)[x := N][y := L]$$

$$\equiv \lambda z.(M_1[x := N][y := L])$$

$$\equiv \lambda z.(M_1[y := L][x := N[y := L]])$$

$$\equiv (\lambda z.M_1)[y := L][x := N[y := L]].$$

- **Case 3:** $M \equiv M_1M_2$. The statement follows again from the induction hypothesis. □

Substitution Lemma: If $x \neq y$ and $x \notin FV(L)$, then

$$M[x := N][y := L] \equiv M[y := L][x := N[y := L]].$$

Proof: By induction on the structure of M .

- **Case 1:** M is a variable.

Case 1.1. $M \equiv x$. Then both sides equal $N[y := L]$ since $x \neq y$.

Case 1.2. $M \equiv y$. Then both sides equal L , for $x \notin FV(L)$
implies $L[x := \dots] \equiv L$.

Case 1.3. $M \equiv z \neq x, y$. Then both sides equal z .

- **Case 2:** $M \equiv \lambda z.M_1$. By the variable convention we may assume that $z \neq x, y$ and z is not free in N, L . Then by induction hypothesis

$$\begin{aligned} & (\lambda z.M_1)[x := N][y := L] \\ & \equiv \lambda z.(M_1[x := N][y := L]) \\ & \equiv \lambda z.(M_1[y := L][x := N[y := L]]) \\ & \equiv (\lambda z.M_1)[y := L][x := N[y := L]]. \end{aligned}$$

- **Case 3:** $M \equiv M_1M_2$. The statement follows again from the induction hypothesis. □

Informal Reasoning

For example Karl Crary in the textbook on Advanced Type-Systems:

“As usual, we will identify terms that differ only in the names of bound variables, and our substitution is capture avoiding.”

This seems to suggest it is enough to provide:

- nominal datatypes (named α -equivalence classes) + simple reasoning infrastructure,
- recursion over α -equivalence classes
- and structural inductions.

Informal Reasoning

For example Karl Cray in the textbook on
Advanced Type-Systems:

"As usual, we will identify terms that differ
only
subs

**We found this is not enough:
one needs induction principles
that have the usual variable
convention already built in.**

This s

- nominal data types (named α -equivalence classes) + simple reasoning infrastructure,
- recursion over α -equivalence classes
- and structural inductions.

Typing Judgements

■ Typing-rules for the λ -calculus are specified as:

$$\frac{(x:\tau) \in \Gamma \text{ valid } \Gamma}{\Gamma \vdash x : \tau}$$

$$\frac{\Gamma \vdash M : \sigma \rightarrow \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash M N : \tau}$$

$$\frac{x \# \Gamma \quad \{x:\sigma\} \cup \Gamma \vdash M : \tau}{\Gamma \vdash \lambda x.M : \sigma \rightarrow \tau}$$

$$\frac{}{\text{valid } \emptyset}$$

$$\frac{x \# \Gamma \quad \text{valid } \Gamma}{\text{valid } \{x:\tau\} \cup \Gamma}$$

■ Weakening: If $\Gamma_1 \vdash M : \tau$ and $\text{valid } \Gamma_2, \Gamma_1 \subseteq \Gamma_2$
then $\Gamma_2 \vdash M : \tau$.

Induction “For Free”

- The induction principle that comes with this definition is as follows:

$$\forall \Gamma x \tau. (x : \tau) \in \Gamma \wedge \text{valid } \Gamma \Rightarrow P \Gamma (x) \tau$$

$$\forall \Gamma M N \sigma \tau.$$

$$P \Gamma M (\sigma \rightarrow \tau) \wedge P \Gamma N \sigma \Rightarrow P \Gamma (M N) \sigma$$

$$\forall \Gamma x M \sigma \tau.$$

$$x \notin \Gamma \wedge$$

$$P (\{x : \sigma\} \cup \Gamma) M \tau \Rightarrow P \Gamma (\lambda x. t) (\sigma \rightarrow \tau)$$

$$\Gamma \vdash M : \tau \Rightarrow P \Gamma M \tau$$

Note that it says “for all x ...”

Proof for Weakening

$$\frac{x \# \Gamma \quad \{x:\sigma\} \cup \Gamma \vdash M : \tau}{\Gamma \vdash \lambda x.M : \sigma \rightarrow \tau}$$

■ If $\Gamma_1 \vdash M : \tau$ then $\forall \Gamma_2. \text{valid } \Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : \tau$

■ We know:

$$\forall \Gamma_2. \text{valid } \Gamma_2 \wedge \{x:\sigma\} \cup \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : \tau$$

$x \# \Gamma_1$

■ We have to show:

$$\forall \Gamma_2. \text{valid } \Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash \lambda x.M : \sigma \rightarrow \tau$$

Proof for Weakening

$$\frac{x \# \Gamma \quad \{x:\sigma\} \cup \Gamma \vdash M : \tau}{\Gamma \vdash \lambda x.M : \sigma \rightarrow \tau}$$

■ If $\Gamma_1 \vdash M : \tau$ then $\forall \Gamma_2. \text{valid } \Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : \tau$

For all Γ_1, x, M, σ and τ :

■ We know:

$$\forall \Gamma_2. \text{valid } \Gamma_2 \wedge \{x:\sigma\} \cup \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : \tau$$

$x \# \Gamma_1$

■ We have to show:

$$\forall \Gamma_2. \text{valid } \Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash \lambda x.M : \sigma \rightarrow \tau$$

Proof for Weakening

$$\frac{x \# \Gamma \quad \{x:\sigma\} \cup \Gamma \vdash M : \tau}{\Gamma \vdash \lambda x.M : \sigma \rightarrow \tau}$$

■ If $\Gamma_1 \vdash M : \tau$ then $\forall \Gamma_2. \text{valid } \Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : \tau$

For all Γ_1, x, M, σ and τ :

■ We know:

$$\forall \Gamma_2. \text{valid } \Gamma_2 \wedge \{x:\sigma\} \cup \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : \tau$$

$$x \# \Gamma_1$$

$$\text{valid } \Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2$$

■ We have to show:

$$\Gamma_2 \vdash \lambda x.M : \sigma \rightarrow \tau$$

Proof for Weakening

$$\frac{x \# \Gamma \quad \{x:\sigma\} \cup \Gamma \vdash M : \tau}{\Gamma \vdash \lambda x.M : \sigma \rightarrow \tau}$$

■ If $\Gamma_1 \vdash M : \tau$ then $\forall \Gamma_2. \text{valid } \Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : \tau$

For all Γ_1, x, M, σ and τ :

■ We know:

$$\forall \Gamma_2. \text{valid } \Gamma_2 \wedge \{x:\sigma\} \cup \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : \tau$$

$$x \# \Gamma_1$$

$$\text{valid } \Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2$$

■ We have to show:

$$\Gamma_2 \vdash \lambda x.M : \sigma \rightarrow \tau$$

Proof for Weakening

$$\frac{x \# \Gamma \quad \{x:\sigma\} \cup \Gamma \vdash M : \tau}{\Gamma \vdash \lambda x.M : \sigma \rightarrow \tau}$$

■ If $\Gamma_1 \vdash M : \tau$ then $\forall \Gamma_2. \text{valid } \Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : \tau$

For all Γ_1, x, M, σ and τ :

■ We know:

$$\Gamma_2 \mapsto \{x:\sigma\} \cup \Gamma_2$$

$$\forall \Gamma_2. \text{valid } \Gamma_2 \wedge \{x:\sigma\} \cup \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : \tau$$

$$x \# \Gamma_1$$

$$\text{valid } \Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2$$

■ We have to show:

$$\Gamma_2 \vdash \lambda x.M : \sigma \rightarrow \tau$$

Proof for Weakening

$$\frac{x \# \Gamma \quad \{x:\sigma\} \cup \Gamma \vdash M : \tau}{\Gamma \vdash \lambda x.M : \sigma \rightarrow \tau}$$

■ If $\Gamma_1 \vdash M : \tau$ then $\forall \Gamma_2. \text{valid } \Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : \tau$

For all Γ_1, x, M, σ and τ :

■ We know:

$$\Gamma_2 \mapsto \{x:\sigma\} \cup \Gamma_2$$

$$\forall \Gamma_2. \text{valid } \Gamma_2 \wedge \{x:\sigma\} \cup \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : \tau$$

$$x \# \Gamma_1$$

$$\text{valid } \Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2 \Rightarrow \{x:\sigma\} \cup \Gamma_1 \subseteq \{x:\sigma\} \cup \Gamma_2$$

■ We have to show:

$$\Gamma_2 \vdash \lambda x.M : \sigma \rightarrow \tau$$

Proof for Weakening

$$\frac{x \# \Gamma \quad \{x:\sigma\} \cup \Gamma \vdash M : \tau}{\Gamma \vdash \lambda x.M : \sigma \rightarrow \tau}$$

■ If $\Gamma_1 \vdash M : \tau$ then $\forall \Gamma_2. \text{valid } \Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : \tau$

For all Γ_1, x, M, σ and τ :

■ We know:

$$\Gamma_2 \mapsto \{x:\sigma\} \cup \Gamma_2$$

$$\forall \Gamma_2. \text{valid } \Gamma_2 \wedge \{x:\sigma\} \cup \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : \tau$$

$$x \# \Gamma_1$$

$$\text{valid } \Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2 \Rightarrow \{x:\sigma\} \cup \Gamma_1 \subseteq \{x:\sigma\} \cup \Gamma_2$$

$$\text{valid } \{x:\sigma\} \cup \Gamma_2 \text{ ???}$$

■ We have to show:

$$\Gamma_2 \vdash \lambda x.M : \sigma \rightarrow \tau$$

- The usual proof of strong normalisation for simply-typed lambda terms establishes first:

Lemma: If for all reducible s , $t[x := s]$ is reducible, then $\lambda x.t$ is reducible.

- Then one shows for a closing (simultaneous) substitution:

Theorem: If $\Gamma \vdash t : \tau$, then for all closing substitutions θ containing reducible terms only, $\theta(t)$ is reducible.

Lambda-Case: By induction we know $(x \mapsto s \cup \theta)(t)$ with s being reducible. This is equal* to $(\theta(t))[x := s]$. Consequently, we can apply the lemma and get $\lambda x.(\theta(t))$. Because this is equal* to $\theta(\lambda x.t)$ we are done.

*take a deep breath

- The usual proof of strong normalisation for simply-typed lambda terms establishes first:

Lemma: If for all reducible s , $t[x := s]$ is reducible, then $\lambda x.t$ is reducible.

- Girard writes in P&T (roughly): case u is $\lambda x.t$ of type $\sigma \rightarrow \tau$: by induction hypothesis $(x \mapsto s \cup \theta)(t)$ is reducible for all s of type σ . Lemma 6.3.2 says that $\theta(u) = \lambda x.\theta(t)$ is reducible.

Lambda-Case: By induction we know $(x \mapsto s \cup \theta)(t)$ with s being reducible. This is equal* to $(\theta(t))[x := s]$. Consequently, we can apply the lemma and get $\lambda x.(\theta(t))$. Because this is equal* to $\theta(\lambda x.t)$ we are done.

*take a deep breath

Strong Induction

- Instead of using the for-free (weak) induction principle:

$$\forall \Gamma x \tau. (x : \tau) \in \Gamma \wedge \text{valid } \Gamma \Rightarrow P \Gamma (x) \tau$$

$$\forall \Gamma M N \sigma \tau.$$

$$P \Gamma M (\sigma \rightarrow \tau) \wedge P \Gamma N \sigma \Rightarrow P \Gamma (M N) \sigma$$

$$\forall \Gamma x M \sigma \tau.$$

$$x \# \Gamma \wedge$$

$$P (\{x : \sigma\} \cup \Gamma) M \tau \Rightarrow P \Gamma (\lambda x.t) (\sigma \rightarrow \tau)$$

$$\Gamma \vdash M : \tau \Rightarrow P \Gamma M \tau$$

Strong Induction

■ we use the following strong induction principle

$$\forall c \Gamma x \tau. (x : \tau) \in \Gamma \wedge \text{valid } \Gamma \Rightarrow P c \Gamma (x) \tau$$

$$\forall c \Gamma M N \sigma \tau.$$

$$(\forall d. P d \Gamma M (\sigma \rightarrow \tau)) \wedge (\forall d. P d \Gamma N \sigma) \\ \Rightarrow P c \Gamma (M N) \sigma$$

$$\forall c \Gamma x M \sigma \tau.$$

$$x \# \Gamma \wedge x \# c \wedge$$

$$(\forall d. P d (\{x : \sigma\} \cup \Gamma) M \tau) \Rightarrow P c \Gamma (\lambda x. t) (\sigma \rightarrow \tau)$$

$$\Gamma \vdash M : \tau \Rightarrow P c \Gamma M \tau$$

■ and set up the induction so that it “avoids” Γ_2 (in case of the weakening lemma) or “avoids” θ (in case of SN)

Weakening (Again)

$$\frac{x \# \Gamma \quad \{x:\sigma\} \cup \Gamma \vdash M : \tau}{\Gamma \vdash \lambda x.M : \sigma \rightarrow \tau}$$

■ If $\Gamma_1 \vdash M : \tau$ then valid $\Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : \tau$
avoiding Γ_2

For all Γ_1, x, M, σ and τ :

■ We know:

$$\forall \Gamma_2. \text{valid } \Gamma_2 \wedge \{x:\sigma\} \cup \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : \tau$$

$$x \# \Gamma_1$$

$$\text{valid } \Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2$$

$$x \# \Gamma_2$$

■ We have to show:

$$\Gamma_2 \vdash \lambda x.M : \sigma \rightarrow \tau$$

Weakening (Again)

$$\frac{x \# \Gamma \quad \{x:\sigma\} \cup \Gamma \vdash M : \tau}{\Gamma \vdash \lambda x.M : \sigma \rightarrow \tau}$$

■ If $\Gamma_1 \vdash M : \tau$ then valid $\Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : \tau$
avoiding Γ_2

For all Γ_1, x, M, σ and τ :

■ We know:

$$\forall \Gamma_2. \text{valid } \Gamma_2 \wedge \{x:\sigma\} \cup \Gamma_1 \subseteq \Gamma_2 \Rightarrow \Gamma_2 \vdash M : \tau$$

$$x \# \Gamma_1$$

$$\text{valid } \Gamma_2 \wedge \Gamma_1 \subseteq \Gamma_2 \Rightarrow \{x:\sigma\} \cup \Gamma_1 \subseteq \{x:\sigma\} \cup \Gamma_2$$

$$x \# \Gamma_2 \Rightarrow \text{valid } \{x:\sigma\} \cup \Gamma_2$$

■ We have to show:

$$\Gamma_2 \vdash \lambda x.M : \sigma \rightarrow \tau$$

SN (Again)

Theorem: If $\Gamma \vdash t : \tau$, then for all closing substitutions θ containing reducible terms only, $\theta(t)$ is reducible.

■ Since we say that the strong induction should avoid θ , we get the assumption $x \# \theta$; then

■ in the Lambda-Case:

By induction we know $(x \mapsto s \cup \theta)(t)$ with s being reducible. This is equal to $(\theta(t))[x := s]$.

Consequently, we can apply the lemma and get $\lambda x.(\theta(t))$. Because this is equal to $\theta(\lambda x.t)$ we are done.

$$x \# \theta \Rightarrow (x \mapsto s \cup \theta)(t) = (\theta(t))[x := s]$$
$$\theta(\lambda x.t) = \lambda x.(\theta(t))$$

Honest Toil, No Theft

- Remember the sacred principle of HOL:

“The method of ‘postulating’ what we want has many advantages; they are the same as the advantages of theft over honest toil.”

B. Russell, Introduction of Mathematical Philosophy

- I will show next that the weak, for-free induction principle implies the strong induction principle.

(I am only going to show the lambda-case.)

The Strong Induction Principle

■ Remember I am going to show you now:

$$\forall c \Gamma x \tau. (x : \tau) \in \Gamma \wedge \text{valid } \Gamma \Rightarrow P c \Gamma (x) \tau$$

$$\forall c \Gamma M N \sigma \tau.$$

$$(\forall d. P d \Gamma M (\sigma \rightarrow \tau)) \wedge (\forall d. P d \Gamma N \sigma) \\ \Rightarrow P c \Gamma (M N) \sigma$$

$$\forall c \Gamma x M \sigma \tau.$$

$$x \# \Gamma \wedge x \# c \wedge$$

$$(\forall d. P d (\{x : \sigma\} \cup \Gamma) M \tau) \Rightarrow P c \Gamma (\lambda x. t) (\sigma \rightarrow \tau)$$

$$\Gamma \vdash M : \tau \Rightarrow P c \Gamma M \tau$$

Proof

■ We prove $P \text{ c } \Gamma \text{ t } \tau$

Proof

■ We prove $\forall \pi c. P c (\pi \bullet \Gamma) (\pi \bullet t) \tau$

Proof

■ We prove $\forall \pi c. P c (\pi \bullet \Gamma) (\pi \bullet t) \tau$

■ I.e., we show $P c (\pi \bullet \Gamma) (\pi \bullet (\lambda x.t)) \sigma \rightarrow \tau$

Proof

■ We prove $\forall \pi c. P c (\pi \bullet \Gamma) (\pi \bullet t) \tau$

■ I.e., we show $P c (\pi \bullet \Gamma) (\lambda \pi \bullet x. \pi \bullet t) \sigma \rightarrow \tau$

Proof

- We prove $\forall \pi c. P c (\pi \bullet \Gamma) (\pi \bullet t) \tau$
- I.e., we show $P c (\pi \bullet \Gamma) (\lambda \pi \bullet x. \pi \bullet t) \sigma \rightarrow \tau$
- We have $\forall \pi d. P d (\pi \bullet (x : \sigma \cup \Gamma)) (\pi \bullet t) \tau$ and $x \notin \Gamma$ by induction

Proof

- We prove $\forall \pi c. P c (\pi \bullet \Gamma) (\pi \bullet t) \tau$
- I.e., we show $P c (\pi \bullet \Gamma) (\lambda \pi \bullet x. \pi \bullet t) \sigma \rightarrow \tau$
- We have $\forall \pi d. P d (\pi \bullet (x : \sigma \cup \Gamma)) (\pi \bullet t) \tau$ and $\pi \bullet x \# \pi \bullet \Gamma$ by induction

Proof

- We prove $\forall \pi c. P c (\pi \bullet \Gamma) (\pi \bullet t) \tau$
- I.e., we show $P c (\pi \bullet \Gamma) (\lambda \pi \bullet x. \pi \bullet t) \sigma \rightarrow \tau$
- We have $\forall \pi d. P d (\pi \bullet (x : \sigma \cup \Gamma)) (\pi \bullet t) \tau$ and $\pi \bullet x \neq \pi \bullet \Gamma$ by induction
- Our weaker precondition says that:
$$\forall c \Gamma x t \sigma \tau. x \neq \Gamma \wedge x \neq c \wedge$$
$$(\forall d. P d (x : \tau \cup \Gamma) t \tau) \Rightarrow P c \Gamma (\lambda x. t) \sigma \rightarrow \tau$$

Proof

- We prove $\forall \pi c. P c (\pi \bullet \Gamma) (\pi \bullet t) \tau$
- I.e., we show $P c (\pi \bullet \Gamma) (\lambda \pi \bullet x. \pi \bullet t) \sigma \rightarrow \tau$
- We have $\forall \pi d. P d (\pi \bullet (x : \sigma \cup \Gamma)) (\pi \bullet t) \tau$ and $\pi \bullet x \# \pi \bullet \Gamma$ by induction
- Our weaker precondition says that:
$$\forall c \Gamma x t \sigma \tau. x \# \Gamma \wedge x \# c \wedge$$
$$(\forall d. P d (x : \tau \cup \Gamma) t \tau) \Rightarrow P c \Gamma (\lambda x. t) \sigma \rightarrow \tau$$
- We choose a fresh y s.t. $y \# (\pi \bullet x, \pi \bullet t, \pi \bullet \Gamma, c)$

Proof

- We prove $\forall \pi c. P c (\pi \bullet \Gamma) (\pi \bullet t) \tau$
- I.e., we show $P c (\pi \bullet \Gamma) (\lambda \pi \bullet x. \pi \bullet t) \sigma \rightarrow \tau$
- We have $\forall \pi d. P d (\pi \bullet (x : \sigma \cup \Gamma)) (\pi \bullet t) \tau$ and $\pi \bullet x \# \pi \bullet \Gamma$ by induction
- Our weaker precondition says that:
$$\forall c \Gamma x t \sigma \tau. x \# \Gamma \wedge x \# c \wedge$$
$$(\forall d. P d (x : \tau \cup \Gamma) t \tau) \Rightarrow P c \Gamma (\lambda x. t) \sigma \rightarrow \tau$$
- We choose a fresh y s.t. $y \# (\pi \bullet x, \pi \bullet t, \pi \bullet \Gamma, c)$
- Now we can use
$$\forall d. P d ((y \pi \bullet x) :: \pi \bullet (x : \sigma \cup \Gamma)) ((y \pi \bullet x) :: \pi \bullet t) \tau$$

Proof

■ We prove $\forall \pi c. P c (\pi \bullet \Gamma) (\pi \bullet t) \tau$

■ I.e., we show $P c (\pi \bullet \Gamma) (\lambda \pi \bullet x. \pi \bullet t) \sigma \rightarrow \tau$

■ We have $\forall \pi d. P d (\pi \bullet (x : \sigma \cup \Gamma)) (\pi \bullet t) \tau$ and $\pi \bullet x \# \pi \bullet \Gamma$ by induction

■ Our weaker precondition says that:

$$\forall c \Gamma x t \sigma \tau. x \# \Gamma \wedge x \# c \wedge (\forall d. P d (x : \tau \cup \Gamma) t \tau) \Rightarrow P c \Gamma (\lambda x. t) \sigma \rightarrow \tau$$

■ We choose a fresh y s.t. $y \# (\pi \bullet x, \pi \bullet t, \pi \bullet \Gamma, c)$

■ Now we can use

$$\forall d. P d ((y \pi \bullet x) \bullet \pi \bullet (x : \sigma \cup \Gamma)) ((y \pi \bullet x) \bullet \pi \bullet t) \tau$$
$$P c (y \pi \bullet x) \bullet \pi \bullet \Gamma (\lambda y. ((y \pi \bullet x) \bullet \pi \bullet t)) \sigma \rightarrow \tau$$

Proof

■ We prove $\forall \pi c. P c (\pi \bullet \Gamma) (\pi \bullet t) \tau$

■ I.e., we show $P c (\pi \bullet \Gamma) (\lambda \pi \bullet x. \pi \bullet t) \sigma \rightarrow \tau$

■ We have $\forall \pi d. P d (\pi \bullet (x : \sigma \cup \Gamma)) (\pi \bullet t) \tau$ and $\pi \bullet x \# \pi \bullet \Gamma$ by induction

■ Our weaker precondition says that:

$$\forall c \Gamma x t \sigma \tau. x \# \Gamma \wedge x \# c \wedge (\forall d. P d (x : \tau \cup \Gamma) t \tau) \Rightarrow P c \Gamma (\lambda x. t) \sigma \rightarrow \tau$$

■ We choose a fresh y s.t. $y \# (\pi \bullet x, \pi \bullet t, \pi \bullet \Gamma, c)$

■ Now we can use

$$\forall d. P d ((y \pi \bullet x) \bullet \pi \bullet (x : \sigma \cup \Gamma)) ((y \pi \bullet x) \bullet \pi \bullet t) \tau$$
$$P c (y \pi \bullet x) \bullet \pi \bullet \Gamma ((y \pi \bullet x) \bullet \lambda \pi \bullet x. \pi \bullet t) \sigma \rightarrow \tau$$

Proof

■ We prove $\forall \pi c. P c (\pi \bullet \Gamma) (\pi \bullet t) \tau$

■ I.e., we show $P c (\pi \bullet \Gamma) (\lambda \pi \bullet x. \pi \bullet t) \sigma \rightarrow \tau$

■ We have $\forall \pi d. P d (\pi \bullet (x : \sigma \cup \Gamma)) (\pi \bullet t) \tau$ and $\pi \bullet x \# \pi \bullet \Gamma$ by induction

■ Our weaker precondition says that:

$$\forall c \Gamma x t \sigma \tau. x \# \Gamma \wedge x \# c \wedge (\forall d. P d (x : \tau \cup \Gamma) t \tau) \Rightarrow P c \Gamma (\lambda x. t) \sigma \rightarrow \tau$$

■ We choose a fresh y s.t. $y \# (\pi \bullet x, \pi \bullet t, \pi \bullet \Gamma, c)$

■ Now we can use

$$\forall d. P d ((y \pi \bullet x) \bullet \pi \bullet (x : \sigma \cup \Gamma)) ((y \pi \bullet x) \bullet \pi \bullet t) \tau$$
$$P c (y \pi \bullet x) \bullet \pi \bullet \Gamma ((y \pi \bullet x) \bullet \lambda \pi \bullet x. \pi \bullet t) \sigma \rightarrow \tau$$

■ However $(y \pi \bullet x) \bullet \pi \bullet \Gamma = \pi \bullet \Gamma$ and $(y \pi \bullet x) \bullet \lambda \pi \bullet x. \pi \bullet t = \lambda \pi \bullet x. \pi \bullet t$

Proof

■ We prove $\forall \pi c. P c (\pi \bullet \Gamma) (\pi \bullet t) \tau$

■ I.e., we show $P c (\pi \bullet \Gamma) (\lambda \pi \bullet x. \pi \bullet t) \sigma \rightarrow \tau$

■ We have $\forall \pi d. P d (\pi \bullet (x : \sigma \cup \Gamma)) (\pi \bullet t) \tau$ and $\pi \bullet x \# \pi \bullet \Gamma$ by induction

■ Our weaker precondition says that:

$$\forall c \Gamma x t \sigma \tau. x \# \Gamma \wedge x \# c \wedge (\forall d. P d (x : \tau \cup \Gamma) t \tau) \Rightarrow P c \Gamma (\lambda x. t) \sigma \rightarrow \tau$$

■ We choose a fresh y s.t. $y \# (\pi \bullet x, \pi \bullet t, \pi \bullet \Gamma, c)$

■ Now we can use

$$\forall d. P d ((y \pi \bullet x) \bullet \pi \bullet (x : \sigma \cup \Gamma)) ((y \pi \bullet x) \bullet \pi \bullet t) \tau$$
$$P c (\pi \bullet \Gamma) (\lambda \pi \bullet x. \pi \bullet t) \sigma \rightarrow \tau$$

■ However $(y \pi \bullet x) \bullet \pi \bullet \Gamma = \pi \bullet \Gamma$ and $(y \pi \bullet x) \bullet \lambda \pi \bullet x. \pi \bullet t = \lambda \pi \bullet x. \pi \bullet t$

Conclusions

- The nominal datatype package automatically derives the strong structural induction principles for all nominal datatypes (not just the lambda-calculus).
- Also for rule inductions (though they have to satisfy some reasonability constraints).
- They are easy to use: you just have to think carefully what the variable convention should be.
- We can explore the "dark" corners of the variable convention: when and where it can be used.

Conclusions

- The nominal datatype package automatically derives the strong structural induction principles for all nominal datatypes (not just the lambda-calculus).
- Also for rule inductions (though they have to satisfy some reasonability constraints).
- They are easy to use: you just have to think carefully what the variable convention should be.
- We can explore the "dark" corners of the variable convention: when and where it can be used.
- **Point to take home:** Actually these proofs using the variable convention are all trivial / obvious / routine... **provided** you use the nominal datatype package ;o)

Quiz

Imagine...

Var "name"

App "lam" "lam"

Lam "«name»lam"

Foo "«name»«name»lam" "«name»«name»lam"

That means roughly:

$Foo (\lambda x.y.t_1) (\lambda z.u.t_2)$

- What does the variable convention look like for **Foo**?
- What does the clause for capture-avoiding substitution look like?

Answers: Download the nominal datatype package and try it out.
<http://isabelle.in.tum.de/nominal>