

# Representations of Binding: Local Representations

Randy Pollack

Version of October 26, 2011

## Outline

Naive Syntax Trees: Problems with binding

Local Representation

- Variable-Closed Sexprs

- Relations on terms

- Reasoning about Relations on Terms

Canonical Local Representations

- Locally Nameless Representation

- Sato Representation

- Adequacy of the Representation

- Example

# Outline

## Naive Syntax Trees: Problems with binding

### Local Representation

Variable-Closed Sexprs

Relations on terms

Reasoning about Relations on Terms

### Canonical Local Representations

Locally Nameless Representation

Sato Representation

Adequacy of the Representation

Example

## Pure Lambda terms: Raw Syntax

- ▶ Countable set  $\mathbb{X}$  of *atoms* used for *variables*:  $X, Y, Z$ .
  - ▶ Only relation needed on  $\mathbb{X}$  is decidable equality.
- ▶ Datatype of *terms* ranged over by  $M, N, P, Q$ :

$$M ::= X \mid P.Q \mid [X]M$$

- ▶ **Substitution**; naive definition as in “Software Foundations” and Greg’s Coq file for STLC:

$$\begin{aligned} [M/Y]X &:= \text{if } Y = X \text{ then } M \text{ else } X \\ [M/Y]N_1.N_2 &:= ([M/Y]N_1).[M/Y]N_2 \\ [M/Y]([X]N) &:= [X](\text{if } Y = X \text{ then } N \text{ else } [M/Y]N) \end{aligned}$$

- ▶ This is a definition by structural recursion.
- ▶ **Well known that naive substitution is wrong: allows capture**

$$[X/Y]([X]Y) = [X]X.$$

## Substitution isn't the only problem: Typing

- ▶ Let  $A, B, \dots$  be *simple types* (implicational propositions).
- ▶ *Valid contexts*  $(\Gamma, \Delta)$  are lists of uniquely labelled assumptions:

$\Gamma ::= X_1, A_1, \dots, X_n, A_n$     where the  $X_i$  are pairwise distinct

- ▶ Consider the rules for Simply Typed Lambda Calculus (STLC):

$$\frac{\Gamma \text{ valid} \quad X:A \in \Gamma}{\Gamma \vdash X : A} \quad (\text{ELIM}) \frac{\Gamma \vdash b : A \rightarrow B \quad \Gamma \vdash a : A}{\Gamma \vdash ba : B}$$

$$(\text{INTRO}) \frac{\Gamma, Y:A \vdash b : B}{\Gamma \vdash [Y]b : A \rightarrow B}$$

- ▶ Does this system accept the shadowing judgement

$$\vdash [Y]([Y]Y) : A \rightarrow A \rightarrow A ?$$

## Substitution isn't the only problem: Typing (2)

Greg's Coq STLC does the rules differently to accept shadowing:

$$\frac{\text{LIFO lookup } (X, \Gamma) = A}{\Gamma \vdash X : A} \quad (\text{ELIM}) \frac{\Gamma \vdash b : A \rightarrow B \quad \Gamma \vdash a : A}{\Gamma \vdash b a : B}$$

$$(\text{INTRO}) \frac{\Gamma, Y:A \vdash b : B}{\Gamma \vdash [Y]b : A \rightarrow B}$$

- ▶ No validity requirement for contexts; LIFO lookup instead.
- ▶ Accepts the judgement  $\vdash [Y]([Y]Y) : A \rightarrow A \rightarrow A$ .
- ▶ But this system doesn't accept context permutation
  - ▶ which the previous system does.
- ▶ Worse, LIFO lookup cannot handle dependent types.

## Shadowing in Dependent Types

- ▶ Consider a naive INTRO rule for dependent types

$$\text{INTRO} \frac{\Gamma, X:A \vdash M : B}{\Gamma \vdash \lambda X:A. M : \Pi X:A. B}$$

- ▶ The following correct judgement with shadowing is not derivable:

$$A:\star, P:A \rightarrow \star \vdash \lambda X:A. (\lambda X:P.X. X) : \Pi X:A. (\Pi Y:P.X. P.X) .$$

- ▶ The stacking of dependencies in terms and types can differ.
- ▶ LIFO lookup doesn't solve the shadowing problem for dependent types.

## What is the problem?

- ▶ The raw syntax datatype doesn't respect our idea of binding:
  - ▶  $[X]X \neq [Y]Y$  as elements of the datatype.
  - ▶ Structural induction doesn't give the right induction hypothesis.
- ▶ Informal solution: quotient syntax by  $\alpha$ -equivalence
  - ▶ but  $\alpha$ -equivalence is usually defined in terms of substitution
    - ▶ Smallest congruence relation containing

$$\frac{Y \neq X \quad Y \notin \text{FV}(M)}{[X]M =_{\alpha} [Y]([Y/X]M)}$$

- ▶ so first define substitution on raw syntax, then define  $\alpha$ -equivalence and show substitution respects it.

Not very pretty: our game is to avoid defining  $\alpha$ -equivalence at all.

## Aside: $\alpha$ without substitution

- ▶ Let  $(X Y)\bullet M$  be the operation that swaps (permutes)  $X$  and  $Y$  in term  $M$ .
  - ▶ Viewing  $M$  as raw syntax: no binding,
- ▶  $\alpha$ -equivalence is the smallest congruence containing

$$\frac{Z \sharp (X, Y, M, N) \quad (X Z)\bullet M =_{\alpha} (Y Z)\bullet N}{[X]M =_{\alpha} [Y]N}$$

- ▶ where  $Z \sharp M$  means  $Z$  doesn't appear in the raw syntax  $M$ .
- ▶ The operations of swapping  $\bullet$  and freshness  $\sharp$  are definable on raw syntax by structural recursion.

Name permutation on raw syntax is the basic idea of nominal sets and nominal logic.

- ▶ This technique was already observed in [McKinna/Pollack 1993].

## Back to the standard approach: Carefully define substitution on raw syntax

- ▶ Commonly used in modern presentations since Church; Curry and Feys.

$$[c/X]Y \quad := \quad \text{if } X = Y \text{ then } c \text{ else } Y$$

$$[c/X](b_1 \cdot b_2) := ([c/X]b_1) \cdot ([c/X]b_2)$$

$$[c/X]([Y]b) := [Z][c/X][Z/Y]b \quad Z \text{ sufficiently fresh}$$

- ▶  $[-/-]b$  is defined by recursion on **length** of  $b$ ,
  - ▶ not on *structure* of  $b$ , since  $[Z/Y]b$  is not a subterm of  $[Y]b$ .
- ▶ Arbitrary choice of fresh  $Z$  could be made canonical ...
  - ▶ e.g. “first name not occurring ...”
  - ▶ this definition is deterministic given a choice function over names.

## Aside: Simultaneous Substitution is Structural

- ▶ A *simultaneous substitution*,  $\rho$ , is a finite partial function from  $\mathbb{X}$  to terms.

$$\begin{aligned} x\rho &:= \rho x \\ (a \cdot b)\rho &:= (a\rho) \cdot (b\rho) \\ ([X]b)\rho &:= [Z](b(\rho, X=Z)) \quad Z \text{ sufficiently fresh} \end{aligned}$$

Primitive recursion with variable parameter.

- ▶ The choice of fresh  $Z$  can be canonical ...
  - ▶ then applying any substitution alpha-normalizes (Stoughton)
  - ▶ so testing alpha-equivalence becomes a test for identity.
- ▶ Note: this operation has funny properties:

$$[X/X]M \neq M$$

Back to our quest to avoid  $\alpha$ -equivalence.

## Outline

Naive Syntax Trees: Problems with binding

### Local Representation

Variable-Closed Sexprs

Relations on terms

Reasoning about Relations on Terms

### Canonical Local Representations

Locally Nameless Representation

Sato Representation

Adequacy of the Representation

Example

## Distinct species of names

Why is it natural to identify bound names with free names?

- ▶ Example:  $\rightarrow$ -intro rule for simple types

$$\text{(INTRO)} \frac{\Gamma, X:A \vdash b : B}{\Gamma \vdash [X]b : A \rightarrow B}$$

- ▶ 'X' really occurs in the premise (free, global).
- ▶ 'X' does not occur in the conclusion (locally bound).

This suggests:

- ▶ **Syntactically separate local (bound) variables from global (free) variables.**
- ▶ Not a new idea: Frege, Gentzen and Prawitz all informally used different species of names.

## Syntax of locally named pre-terms for pure $\lambda$

As in McKinna/Pollack [TLCA 1993, JAR 1999].

- ▶ Countable set  $\mathbb{V}$  of atoms used for local *variables*:  $x, y, z$ .
- ▶ Countable set  $\mathbb{X}$  of atoms, used for global *parameters*:  $X, Y, Z, p, q$ .
- ▶ Only relation needed on  $\mathbb{V}, \mathbb{X}$  is decidable equality.

Symbolic Expressions ( $\mathbb{S}$ ):

- ▶ Datatype of pre-terms ranged over by  $M, N, P, Q$ :

$$M ::= x \mid X \mid P \cdot Q \mid [x]M$$

- ▶ No way to bind global names,  $X$ .
- ▶ In general, may be other classes of variables, parameters and expressions
  - ▶ e.g. types and terms in System F.

## Occurrences of Names

- ▶ Occurrences of global names (parameters)
  - ▶  $X \# A$  means “ $X$  does not occur syntactically in  $A$ ”.
    - ▶ Easily defined by structural recursion
  - ▶ Corresponds to nominal freshness (also written  $\#$ ).

## Substitution, Concretely

- ▶ Concretely defined by *structural* recursion:

$$\begin{aligned}
 [M/X]x &= x \\
 [M/X]Y &= \text{if } X = Y \text{ then } M \text{ else } Y \\
 [M/X]N \cdot N &= ([M/X]N) \cdot [M/X]N \\
 [M/X]([x]N) &= [x][M/X]N
 \end{aligned}$$

- ▶ Deterministic: no choosing arbitrary names.
  - ▶ Thus has natural properties; e.g.

$$\begin{aligned}
 [X/X]M &= M. \\
 X \# M &\implies [P/X]M = M.
 \end{aligned}$$

- ▶ **Does not prevent capture**, e.g.  $[x/X][x]X = [x]x$ .
  - ▶ Will only be used in safe ways.
- ▶ Substitution is a B-algebra homomorphism; see Pollack and Sato (J. Symb. Comp.).

## Not Substitution: a purely technical operation

- ▶ Used to fill a “hole” (free variable) created by going under a binder.
- ▶ Defined by structural recursion:

$$\begin{aligned}
 [M/y]x &= \text{if } y = x \text{ then } M \text{ else } x \\
 [M/y]X &= X \\
 [M/y]N_1 \cdot N_2 &= ([M/y]N_1) \cdot [M/y]N_2 \\
 [M/y]([x]N) &= [x](\text{if } y = x \text{ then } N \text{ else } [M/y]N)
 \end{aligned}$$

- ▶ Respects intended scope of binding.
- ▶ **Does not prevent capture**, e.g.  $[x/y][x]y = [x]x$ .

## Overview: Symbolic expressions vs $\lambda$ -terms

Sexprs do not faithfully represent  $\lambda$ -terms for two reasons.

1. Local variables may appear unbound in sexprs.
  - ▶ 'x' is an sexpr, not intended to represent any  $\lambda$ -term.
    - ▶ Remark: 'X' is an sexpr representing a  $\lambda$ -term with one (particular) global variable.
  - ▶ The fix: select the set of sexprs with no unbound local variables.
    - ▶ Call this subset *vclosed* for *variable closed*.
2. Different sexprs may represent the same  $\lambda$ -term.
  - ▶ '[x]x' and '[y]y'; **not canonical**.
  - ▶ Ignore this for the moment.
    - ▶ We show how to reason correctly with *vclosed* expressions.

## Overview: Symbolic expressions vs $\lambda$ -terms

Sexprs do not faithfully represent  $\lambda$ -terms for two reasons.

1. Local variables may appear unbound in sexprs.
  - ▶ 'x' is an sexpr, not intended to represent any  $\lambda$ -term.
    - ▶ Remark: 'X' is an sexpr representing a  $\lambda$ -term with one (particular) global variable.
  - ▶ The fix: select the set of sexprs with no unbound local variables.
    - ▶ Call this subset *vclosed* for *variable closed*.
2. Different sexprs may represent the same  $\lambda$ -term.
  - ▶ '[x]x' and '[y]y'; **not canonical**.
  - ▶ Ignore this for the moment.
    - ▶ We show how to reason correctly with *vclosed* expressions.

## Overview: Symbolic expressions vs $\lambda$ -terms

Sexprs do not faithfully represent  $\lambda$ -terms for two reasons.

1. Local variables may appear unbound in sexprs.
  - ▶ 'x' is an sexpr, not intended to represent any  $\lambda$ -term.
    - ▶ Remark: 'X' is an sexpr representing a  $\lambda$ -term with one (particular) global variable.
  - ▶ The fix: select the set of sexprs with no unbound local variables.
    - ▶ Call this subset *vclosed* for *variable closed*.
2. Different sexprs may represent the same  $\lambda$ -term.
  - ▶ '[x]x' and '[y]y'; **not canonical**.
  - ▶ Ignore this for the moment.
    - ▶ We show how to reason correctly with *vclosed* expressions.

## Variable-Closed Sexprs

A predicate meaning “no free variables”.

$$\frac{}{vclosed\ X} \qquad \frac{vclosed\ M \quad vclosed\ N}{vclosed\ M \cdot N} \qquad \frac{vclosed\ [X/y]M}{vclosed\ [y]M}$$

- ▶ *vclosed* terms have no unbound local variables.
- ▶ An abstraction is *vclosed* when . . . .
- ▶ Every parameter is *vclosed* and no variable is *vclosed* .
- ▶ Use *vclosed* induction instead of *sexpr* structural induction . . .
- ▶ . . . **no case for unbound variables.**

## Variable-Closed and Substitution

- ▶ Operations  $[M/X]N$  and  $[M/x]N$  are capture free on  $vclosed$ .
  - ▶ There are no free local names to get captured!
- ▶  $vclosed$  is trivially closed under substitution:

$$vclosed M \wedge vclosed N \implies vclosed [M/X]N$$

- ▶ Think of  $vclosed$  as a “weak typing judgement”.
  - ▶  $vclosed$  terms behave well for substitution, just as well-typed terms behave well for computation.

## Aside: Alternative definitions of Variable-Closed

Only the rules for abstraction differ:

$$\frac{X \# M \quad \text{vclosed } [X/y]M}{\text{vclosed } [y]M}$$

$$\frac{\text{vclosed } [X/y]M}{\text{vclosed } [y]M}$$

$$\frac{\text{vclosed } M}{\text{vclosed } [x][x/X]M}$$

$$\frac{\forall X. X \# M \implies \text{vclosed } [X/y]M}{\text{vclosed } [y]M}$$

$$\frac{\forall X. \text{vclosed } [X/y]M}{\text{vclosed } [y]M}$$

$$\frac{\forall Y. (Y \# X \implies Y \# M) \implies \text{vclosed } [Y/X]M}{\text{vclosed } [x][x/X]M}$$

- ▶ These 6 relations are pairwise *extensionally* equivalent:
  - ▶ i.e. they derive the same judgments,
  - ▶ but their derivations are different,
  - ▶ **they behave differently for induction.**
- ▶ Each of the relations in the left hand column have infinitely many derivations of each derivable judgement.
- ▶ Each of the relations in the right hand column have at most one derivation of any judgement.

## Simply Typed Lambda Calculus (STLC)

- ▶ Let *simple types*  $(A, B, \dots)$  and *valid contexts*  $(\Gamma, \Delta)$  be as above

$$\frac{\Gamma \text{ valid} \quad p:A \in \Gamma}{\Gamma \vdash p : A} \quad (\text{ELIM}) \frac{\Gamma \vdash b : A \rightarrow B \quad \Gamma \vdash a : A}{\Gamma \vdash ba : B}$$

$$(\text{INTRO}) \frac{\Gamma, p:A \vdash [p/y]b : B \quad p \# b}{\Gamma \vdash [y]b : A \rightarrow B}$$

- ▶ When going under a binder, substitute a **suitably fresh** parameter in the hole created.
  - ▶ **The choice of  $p$  is arbitrary**; we will have more to say.
- ▶ Why no mention of *vclosed* ?
  - ▶ **lemma:**  $\Gamma \vdash b : B \implies \text{vclosed } b$ .

## Simply Typed Lambda Calculus (2)

$$\frac{\Gamma \text{ valid} \quad p:A \in \Gamma}{\Gamma \vdash p : A} \quad (\text{ELIM}) \frac{\Gamma \vdash b : A \rightarrow B \quad \Gamma \vdash a : A}{\Gamma \vdash ba : B}$$

$$(\text{INTRO}) \frac{\Gamma, p:A \vdash [p/y]b : B \quad p \# b}{\Gamma \vdash [y]b : A \rightarrow B}$$

- ▶ The side condition is needed in rule INTRO to prevent too many judgements being derivable:
  - ▶ if  $p \in b$  then  $p \in \lambda y.b$  in the conclusion, where  $p$  is not bound in the context  $\Gamma$ .
- ▶ Validity side conditions are *not* required in rules INTRO and ELIM because they follow from the premises.
- ▶ This definition of  $\vdash$  is easily formalised in Coq, Isabelle/HOL, ...

## Beta Reduction

$$(\beta) \frac{vclosed [x]b \quad vclosed s}{([x]b) s > [s/x]b}$$

$$(\xi) \frac{[p/x]s > [p/y]t \quad p \# (s, t)}{[x]s > [y]t}$$

$$\frac{s_1 > t \quad vclosed s_2}{s_1 s_2 > t s_2} \qquad \frac{s_2 > t \quad vclosed s_1}{s_1 s_2 > s_1 t}$$

- ▶ In rule  $(\xi)$  we must let  $x$  and  $y$  possibly be different ...
  - ▶ or else some correct reduction judgements won't be derivable.
- ▶ In  $\beta$ , we must restrict to  $(vclosed s)$  for safety.
  - ▶ Otherwise free variables in  $s$  might be captured in  $[s/x]b$ .
- ▶ The other  $vclosed$  restrictions are for hygiene:
  - ▶ **lemma:**  $a > b \implies vclosed a \wedge vclosed b$ .

## Dependent Types

- ▶ Now we see how to handle the problem of dependent types

$$\text{(INTRO)} \frac{\Gamma, p:A \vdash [p/x]M : [p/y]B \quad p \# (M, B)}{\Gamma \vdash \lambda x:A. M : \Pi y:A. B}$$

- ▶ Allowing different names to be bound in different parts of the judgement accounts for different dependencies in terms vs. types.

## What can we do with these relations?

- ▶ Definitions and statements of lemmas are natural using names.
- ▶ All the expected judgements are derivable:
  - ▶ The set of derivable judgments is closed under  $\alpha$ -conversion and renaming.
- ▶ The standard metatheory can be developed:
  - ▶ Weakening, substitution lemma, subject reduction . . .
- ▶ We **almost** never need to define or reason about  $\alpha$ -conversion.
  - ▶ Church-Rosser for  $\beta$ -reduction does not hold on-the-nose for these *vclosed* terms: **not canonical**.
  - ▶ Church-Rosser does hold for Tait/Martin-Löf parallel reduction.
- ▶ We can use the Locally Nameless representation or the Sato representation to get canonicity.

## Weakening for STLC

- ▶ Define **subcontext**:

$$\Gamma \sqsubseteq \Delta \quad \text{iff} \quad \forall p, A . p:A \in \Gamma \implies p:A \in \Delta$$

$\Delta$  contains every assumption occurring in  $\Gamma$ .

### Lemma (Weakening)

$$\Gamma \vdash a : A \implies (\forall \Delta . \Gamma \sqsubseteq \Delta \wedge \Delta \text{ valid} \implies \Delta \vdash a : A).$$

### Remark

- ▶ *We trivially have the lemma:  $\Gamma \vdash a : A \implies \Gamma$  valid.*
- ▶ *... so the premise '  $\Delta$  valid ' is needed in the statement of weakening.*

## Prove weakening

$$\Gamma \vdash a : A \implies (\forall \Delta . \Gamma \sqsubseteq \Delta \wedge \Delta \text{ valid} \implies \Delta \vdash a : A).$$

**Proof:** Attempt proof by induction on the derivation of  $\Gamma \vdash a : A$

- ▶ Consider case for rule INTRO: 
$$\frac{\Gamma, p:A \vdash [p/y]b : B \quad p \# b}{\Gamma \vdash [y]b : A \rightarrow B}$$
- ▶ The goal is  $\Delta \vdash [y]b : A \rightarrow B$
- ▶ The IH is:

$$\forall \Phi . (\Gamma, p_0:A \sqsubseteq \Phi \wedge \Phi \text{ valid}) \implies \Phi \vdash [p_0/y]b : B$$

for some particular  $p_0 \# b$ .

- ▶ By rule INTRO we need to show

$$\Delta, q:A \vdash [q/y]b : B \quad \text{for some } q \# b.$$

- ▶ It seems we want to instantiate  $\Phi$  in IH with  $\Delta, p_0:A \dots$
- ▶ ... but  $\Delta, p_0:A$  may not be valid, as  $p_0$  may occur in  $\Delta$ .

## Proof of weakening (contd)

- ▶ Since  $p_0$  may not be fresh enough, we want to exchange it for a fresh parameter.
- ▶ Let  $(p\ q)\bullet b$  mean *permute all occurrences of  $p$  and  $q$  in  $b$* .
- ▶ As a lemma (**equivariance**), show

$$\Gamma \vdash a : A \implies \forall p\ q . (p\ q)\bullet \Gamma \vdash (p\ q)\bullet a : A. \quad (1)$$

- ▶ This is easy to prove, but even better ...
- ▶ **nominal Isabelle defines polymorphic permutations and proves equivariance automatically.**
- ▶ Now, pick  $q \# (\Delta, b, \Gamma)$ . It suffices to show

$$\Delta, q:A \vdash [q/y]b : B$$

which, by (1) and IH is difficult but possible.

## We have a proof; what's the problem?

- ▶ We must use name swapping explicitly (as in the *weakening* proof above) to handle each example where eigenvariable problems appear.

**Better: we can package this swapping reasoning for each relation (typing, reduction, ...) once and for all.**

- ▶ This technique from McKinna/Pollack (1993).

## A more uniform solution to eigenvariable problems

The following judgements are equivalent:

- ▶ Arbitrary choice of  $p$  in INTRO: judgements may have infinitely many derivations:

$$\frac{\Gamma \text{ valid} \quad p:A \in \Gamma}{\Gamma \vdash p : A} \quad (\text{ELIM}) \quad \frac{\Gamma \vdash b : A \rightarrow B \quad \Gamma \vdash a : A}{\Gamma \vdash ba : B}$$

$$(\text{INTRO}) \quad \frac{\Gamma, p:A \vdash [p/y]b : B \quad p \# b}{\Gamma \vdash [y]b : A \rightarrow B}$$

- ▶ No arbitrary choices: judgements have at most one derivation:

$$\frac{\Gamma \text{ valid} \quad p:A \in \Gamma}{\Gamma \Vdash p : A} \quad (\text{ELIM}) \quad \frac{\Gamma \Vdash b : A \rightarrow B \quad \Gamma \Vdash a : A}{\Gamma \Vdash ba : B}$$

$$(\text{INTRO}) \quad \frac{\forall p. p \# \Gamma \implies \Gamma, p:A \Vdash [p/y]b : B}{\Gamma \Vdash [y]b : A \rightarrow B}$$

## A more uniform solution to eigenvariable problems

- ▶  $\vdash$  is the “official” relation,  $\Vdash$  is an auxiliary notion.
- ▶  $\vdash$  is ordinary syntax: formalizable in Primitive Recursive Arithmetic.
  - ▶  $\vdash$  derivations are well-founded, finitely branching trees.
- ▶  $\Vdash$ , defined by **generalized inductive definition**, is not formalizable in PRA.
  - ▶  $\Vdash$  derivations are well-founded but **infinitely branching** trees.
- ▶ By induction on the derivation of  $\Gamma \Vdash a : A$ , it is trivial that

$$\Gamma \Vdash a : A \implies \Gamma \vdash a : A .$$

The other direction takes some work.

## Why are we interested in this equivalence?

- ▶ With this equivalence we can prove weakening by rule induction without name swapping.
- ▶ **The equivalence of  $\vdash$  and  $\text{If}$  “packages” the eigenvariable reasoning that we need for all examples.**
  - ▶ Introduction of  $\vdash$  is easy: only need property for **one** fresh variable.
  - ▶ Elimination of  $\text{If}$  is powerful: get the IH for **all** sufficiently fresh variables.
  - ▶ We use  $\text{If}$  as a “derived induction principle” for  $\vdash$
- ▶ Instead of using swapping arguments for every rule induction on every relation (typing, reduction, ...)
- ▶ ... we use it **once for each relation**, to prove strengthened induction principle.
- ▶ There is much more to say about this: cofinite quantification, nominal freshness contexts, ...

## Applying the equivalence to prove weakening

- ▶ It is easy to prove weakening:

$$\Gamma \Vdash a : A \implies (\Gamma \sqsubseteq \Delta \wedge \Delta \text{ valid} \implies \Delta \vdash a : A).$$

hence, equivalently, weakening for  $\vdash$ .

- ▶ **Proof:** by induction on the derivation of  $\Gamma \Vdash a : A$ .

- ▶ Consider case for INTRO: 
$$\frac{\forall p. p \# \Gamma \implies \Gamma, p:A \Vdash [p/y]b : B}{\Gamma \Vdash [y]b : A \rightarrow B}$$

- ▶ By rule INTRO (for  $\vdash$ ) we need to show

$$\Delta, p:A \vdash [p/y]b : B \quad \text{for some } p \# b.$$

using IH:

$$\forall \Phi \forall p. (p \# \Gamma \wedge \Gamma, p:A \sqsubseteq \Phi \wedge \Phi \text{ valid}) \implies \Phi \vdash [p/y]b : B.$$

- ▶ Select  $p_0 \# (b, \Gamma, \Phi)$  and instantiate  $\Phi$  in IH with  $\Delta, p_0:A$ .

## Proof of the equivalence of $\vdash$ and $\Vdash$

**Lemma**  $\Gamma \vdash a : A \implies \Gamma \Vdash a : A.$

- ▶ Proof by induction on the derivation of  $\Gamma \vdash a : A.$
- ▶ Consider the case of rule INTRO.
- ▶ Any derivation of  $\vdash$  will use a particular parameter, say  $p_0.$
- ▶ The IH for this case is

$$\Gamma, p_0 : A \Vdash [p_0/y]b : B \quad (p_0 \# b) \quad (\text{also } p_0 \# \Gamma)$$

but to use the INTRO rule for  $\Vdash$  we need the premise

$$\forall p . p \# \Gamma \implies \Gamma, p : A \Vdash [p/y]b : B$$

- ▶ How to reason from a particular parameter to all parameters?

Proof continued:  $\Gamma \vdash a : A \implies \Gamma \Vdash a : A$

We solve the problem using swapping, as in the *weakening* proof.

- ▶ As a lemma, have **equivariance** of  $\Vdash$  :

$$\Gamma \Vdash a : A \implies \forall p q . (p q) \bullet \Gamma \Vdash (p q) \bullet a : A. \quad (2)$$

Nominal Isabelle can prove this automatically, and provides the lemmas about  $\#$  and  $(- -) \bullet -$  that we need.

- ▶ We are trying to prove

$$\forall p . p \# \Gamma \implies \Gamma, p : A \Vdash [p/y]b : B.$$

So pick  $p \# \Gamma$ . (Hence  $p \# b$ .)

- ▶ From IH and (2) have

$$(p p_0) \bullet (\Gamma, p_0 : A) \Vdash (p p_0) \bullet ([p_0/y]b) : B$$

i.e.  $\Gamma, p : A \Vdash [p/y]b : B$  as required.

## Aside: Stronger inversion principles

- ▶ Rule INTRO 
$$\frac{\Gamma, p:A \vdash [p/y]b : B \quad p \# b}{\Gamma \vdash [y]b : A \rightarrow B}$$

gives rise (by induction) to an *inversion principle*:

$$\Gamma \vdash [y]b : T \implies \exists A, B, p. \Gamma, p:A \vdash [p/y]b : B \wedge p \# b \wedge T = A \rightarrow B.$$

- ▶ Rule INTRO 
$$\frac{\forall p. p \# \Gamma \implies \Gamma, p:A \Vdash [p/y]b : B}{\Gamma \Vdash [y]b : A \rightarrow B}$$

gives a stronger inversion principle (using  $\vdash \Leftrightarrow \Vdash$ ):

$$\Gamma \vdash [y]b : T \implies \exists A, B. \forall p \# \Gamma. \Gamma, p:A \vdash [p/y]b : B \wedge p \# b \wedge T = A \rightarrow B.$$

- ▶ Some proofs need stronger inversion.

## Outline

Naive Syntax Trees: Problems with binding

Local Representation

Variable-Closed Sexprs

Relations on terms

Reasoning about Relations on Terms

Canonical Local Representations

Locally Nameless Representation

Sato Representation

Adequacy of the Representation

Example

## Limitations of *vclosed* representation

- ▶ Feasible way to work with *representatives* of  $\alpha$  classes ...
  - ▶ Never needed to define  $\alpha$ -conversion in [McKinna/Pollack].
  - ▶ Can prove parallel- $\beta$ -conversion is confluent.
- ▶ ...but Church–Rosser for  $\beta$ -reduction fails on-the-nose for *vclosed* .
  - ▶ If we wanted to reason about  $\beta$ -reduction we would need to reason about  $\alpha$ -conversion.
- ▶ *vclosed* representation not canonical:  $[x]x \neq [y]y$  .
- ▶ We can get canonical representations:
  - ▶ using de Bruijn indexes (locally nameless)
  - ▶ or by choosing bound variable names canonically (Sato representation).
- ▶ In both these approaches, use strengthened induction principles as above.

## Locally Nameless Representation

- ▶ This is described in detail [POPL 2008] and in a JAR paper by Arthur Charguéraud.
- ▶ Charguéraud's paper comes with a Coq library and many examples.
  - ▶ Arthur's scripts work in Coq 8.3pl2 (26 Oct 2011), but there are some assumptions waiting to be filled in with proofs.
- ▶ Use de Bruijn indexes (numbers,  $n$ ) for local (bound) variables.
- ▶ Use  $X, Y, \dots$  for global parameters, as above.
- ▶ Datatype of raw *terms*:  $M ::= X \mid n \mid P \cdot Q \mid \lambda M$

## Locally Nameless

- ▶ Substitution,  $[M/p]N$  is defined as before.
  - ▶ **No need to lift the free indexes in  $M$**  because  $vclosed M$  have no free indexes.
- ▶ “Hole filling”,  $[M/n]N$ , is defined by structural recursion on  $N$

$$[M/k]i = \text{if } k = i \text{ then } M \text{ else } i$$

$$[M/k]X = X$$

$$[M/k]N_1 \cdot N_2 = ([M/k]N_1) \cdot [M/k]N_2$$

$$[M/k](\lambda N) = \lambda([M/k+1]N)$$

- ▶ This index arithmetic does make reasoning more difficult.

$$\frac{}{vclosed X}$$

$$\frac{vclosed M \quad vclosed N}{vclosed M \cdot N}$$

$$\frac{vclosed [X/0]M}{vclosed \lambda M}$$

- ▶ I leave you to look up the details.

## A Canonical Locally Named Representation

- ▶ Consider again the *vclosed* rules:

$$\frac{}{\text{vclosed } X} \quad \frac{\text{vclosed } M \quad \text{vclosed } N}{\text{vclosed } M \cdot N} \quad \frac{\text{vclosed } M}{\text{vclosed } [x][x/X]M}$$

Local variable 'x' not determined in the rule for abstraction.

- ▶ To define a canonical subset  $\mathbb{L}_F$ , **choose 'x' deterministically**:

$$\frac{}{X : \mathbb{L}_F} \quad \frac{M : \mathbb{L}_F \quad N : \mathbb{L}_F}{M \cdot N : \mathbb{L}_F} \quad \frac{M : \mathbb{L}_F \quad x = F_X(M)}{[x][x/X]M : \mathbb{L}_F}$$

parameterized by a **height function**  $F : \mathbb{X} \times \mathbb{S} \rightarrow \mathbb{V}$ .

- ▶ Clearly  $M : \mathbb{L}_F \implies \text{vclosed } M$ , so substitution is capture free.
- ▶ **Not obvious that  $\mathbb{L}_F$  is closed under substitution.**
- ▶ Still to do: **specify  $F$  such that  $\mathbb{L}_F$  well behaved.**

## Improve Notation

- ▶ Everything is parameterised by a height function  $F$ ; drop explicit subscript.
- ▶ Considering rule:

$$\frac{M : \mathbb{L} \quad x = F_X(M)}{[x][x/X]M : \mathbb{L}}$$

define “abstraction”:

$$\text{abs}_X M \triangleq [F_X(M)][F_X(M)/X]M.$$

- ▶ Abstraction rule can now be written more abstractly.

$$\frac{}{X : \mathbb{L}} \qquad \frac{M : \mathbb{L} \quad N : \mathbb{L}}{M \cdot N : \mathbb{L}} \qquad \frac{M : \mathbb{L}}{\text{abs}_X M : \mathbb{L}}$$

- ▶  $\text{abs}_X M$  will behave like informal ‘ $\lambda X.M$ ’.
  - ▶  $X$  does not occur in  $\text{abs}_X M$ ;  $\text{abs}_X X \equiv \text{abs}_Y Y$ .

## A Datatype of Lambda Terms?

Assume good properties for  $F$  (to be discussed below).

- ▶  $\mathbb{L}$  can be formalized as a **type**
  - ▶  $\mathbb{L}$  is a decidable predicate on  $\mathbb{S}$  ;
    - ▶ can be a proof irrelevant  $\Sigma$ -type in Type Theory.
  - ▶  $\mathbb{L}$  is a non-empty predicate on  $\mathbb{S}$  ;
    - ▶ can be a defined type in HOL.
- ▶ But no amount of clever indexing  $\dots$ , inductive-recursive  $\dots$  can make  $\mathbb{L}$  into a **datatype**  $\dots$
- ▶  $\dots$  because ‘abs’ isn’t injective.

Another topic for another talk.

## Three Good Properties of $F : \mathbb{X} \times \mathbb{S} \rightarrow \mathbb{V}$

(HE)  $F$  is equivariant: Let  $\pi$  be a permutation over  $\mathbb{X}$ , then

$$M : \mathbb{L} \implies F_X(M) = F_{\pi \cdot X}(\pi \cdot M).$$

(HP)  $F$  is preserved by substitution:

$$M : \mathbb{L} \wedge Q : \mathbb{L} \wedge X \neq Y \wedge X \not\# Q \implies F_X(M) = F_X([Q/Y]M).$$

(HF)  $F_X(M)$  does not occur in binding position on any path from the root of  $M$  to any occurrence of  $X$  in  $M$ .

$$M : \mathbb{L} \implies F_X(M) \notin E_X(M)$$

where  $E_X(M) : \mathbb{X} \times \mathbb{S} \rightarrow (\mathbb{V} \text{ set})$  is defined:

$$\begin{aligned} E_X(\alpha) &\triangleq \{\} && \text{if } \alpha \text{ is atomic} \\ E_X(M \cdot N) &\triangleq E_X(M) \cup E_X(N) \\ E_X([x]M) &\triangleq \begin{cases} \{\} & \text{if } X \# M \\ \{x\} \cup E_X(M) & \text{otherwise} \end{cases} \end{aligned}$$

## Consistency and independence of goodness

- ▶ **Consistency:** A good height function exists.
  - ▶ Interpret  $\mathbb{V}$  as natural numbers,  $\mathbb{N}$ .
  - ▶  $H : \mathbb{X} \times \mathbb{S} \rightarrow \mathbb{N}$  defined by structural recursion:

$$H_X(Y) \triangleq \begin{cases} 1 & \text{if } X = Y \\ 0 & \text{if } X \neq Y \end{cases}$$

$$H_X(x) \triangleq 0$$

$$H_X(M \cdot N) \triangleq \max(H_X(M), H_X(N))$$

$$H_X([x]M) \triangleq \begin{cases} H_X(M) & \text{if } H_X(M) = 0 \text{ or } H_X(M) > x \\ x + 1 & \text{otherwise} \end{cases}$$

- ▶ **Independence:** No two of (HE), (HP) and (HF) imply the third.

**Proof** by examples; see JAR paper.

## $\text{abs}_X M$ behaves like abstraction should

We develop a theory of good  $F$ . For example:

- ▶ From (HE)

$$“\lambda X.M \overset{\alpha}{\sim} \lambda Y.N” \implies \text{abs}_X M = \text{abs}_Y N$$

- ▶ From (HF)

$$\text{abs}_X M = \text{abs}_Y N \implies “\lambda X.M \overset{\alpha}{\sim} \lambda Y.N”$$

- ▶ From (HP), substitution “under a binder”

$$X \neq Y \wedge X \# Q \implies [Q/Y]\text{abs}_X M = \text{abs}_X [Q/Y]M$$

Together, (HE), (HF) and (HP) show ‘ $\text{abs}$ ’ behaves correctly for  $\alpha$ -conversion and substitution.

## Is the representation “adequate”?

**Formal vs informal** Relationship between a formal thing and an informal thing is not formalizable.

**Formal vs formal** Adequacy of representation of one formal thing by another formal thing depends on which properties we intend to preserve.

We show formally that  $\mathbb{L}$  is an adequate representation of pure  $\lambda$ -terms in Nominal Isabelle.

- ▶ Let  $A, B, C$  range over nominal terms

$$A ::= X \mid B \cdot C \mid [X]A$$

- ▶ One more definition: ‘instantiation’ (nominal and Sato terms).

$$([x]M) \nabla N \triangleq [N/x]M \qquad ([X]A) \blacktriangledown B \triangleq A[X ::= B]$$

## Isomorphism with Nominal Lambda Terms

- ▶ Define a *representation function* by “structural recursion”:

$$\begin{aligned} !X &\triangleq X \\ !(A \cdot B) &\triangleq !A \cdot !B \\ ![X]A &\triangleq \text{abs}_X !A \end{aligned}$$

- ▶ Need (HE) (F equivariant), to show ! is a function.
- ▶ Assuming F is good, ! is a bijection that preserves substitution and instantiation:

$$\begin{aligned} M : \mathbb{L} &\implies \exists A. !A = M && ! \text{ is surjective,} \\ !A = !B &\implies A = B && ! \text{ is injective,} \\ !(A[X ::= B]) &= [!B/X]!A && ! \text{ respects substitution,} \\ !([X]A) \blacktriangledown Y &= (\text{abs}_X !A) \blacktriangledown Y && ! \text{ respects instantiation} \\ &&& \text{by parameters.} \end{aligned}$$

## A Converse

- ▶ Assume  $\sigma$  is a bijection that preserves instantiation. Then (HE), (HP) and (HF) hold.
- ▶ (Thus preservation of substitution is not independent of the other properties. There is still something to figure out here.)

## Example: $\beta$ -reduction

$$\frac{P : \mathbb{L} \quad N : \mathbb{L}}{(\text{abs}_X P) \cdot N \rightarrow (\text{abs}_X P) \nabla N} \quad (\beta)$$

$$\frac{M_1 \rightarrow M_2 \quad N : \mathbb{L}}{M_1 \cdot N \rightarrow M_2 \cdot N} \quad \frac{M : \mathbb{L} \quad N_1 \rightarrow N_2}{M \cdot N_1 \rightarrow M \cdot N_2}$$

$$\frac{M \rightarrow N}{\text{abs}_X M \rightarrow \text{abs}_X N} \quad (\xi)$$

- ▶  $\rightarrow$  is well behaved, e.g.
  - ▶  $\rightarrow$  is equivariant.
  - ▶ Reduction and well-formedness:  $M \rightarrow N \implies M : \mathbb{L} \wedge N : \mathbb{L}$ .
  - ▶ Reduction respects representation:  $A \rightarrow B \Leftrightarrow !A \rightarrow !B$