An Introduction to Nominal Sets

Andrew Pitts

UNIVERSITY OF CAMBRIDGE

Computer Science & Technology

EWSCS 2020

Lecture 3

Outline

L1 Structural recursion and induction in the presence of name-binding operations.

L2 Introducing the category of nominal sets.

L3 Nominal algebraic data types and α-structural recursion.

L4 Dependently typed λ-calculus with locally fresh names and name-abstraction.

References:

AMP, *Nominal Sets: Names and Symmetry in Computer Science*, CUP 2013

AMP, *Alpha-Structural Recursion and Induction*, JACM 53(2006)459-506.

AMP, J. Matthiesen and J. Derikx,
A Dependent Type Theory with Abstractable Names, ENTCS 312(2015)19-50.

Recall: Alpha-equivalence

Smallest binary relation $=_\alpha$ on Tr closed under the rules:

$$\frac{a \in \mathbb{A}}{\mathtt{V}\,a =_\alpha \mathtt{V}\,a} \qquad \frac{t_1 =_\alpha t_1' \qquad t_2 =_\alpha t_2'}{\mathtt{A}(t_1, t_2) =_\alpha \mathtt{A}(t_1', t_2')}$$

$$\boxed{\frac{(a\ b)\cdot t =_\alpha (a'\ b)\cdot t' \qquad b \notin \{a, a'\} \cup \mathrm{var}(t) \cup \mathrm{var}(t')}{\mathtt{L}(a, t) =_\alpha \mathtt{L}(a', t')}}$$

E.g. $\mathtt{A}(\mathtt{L}(a, \mathtt{A}(\mathtt{V}\,a, \mathtt{V}\,b)), \mathtt{V}\,c) \;=_\alpha\; \mathtt{A}(\mathtt{L}(c, \mathtt{A}(\mathtt{V}\,c, \mathtt{V}\,b)), \mathtt{V}\,c)$
$\neq_\alpha\; \mathtt{A}(\mathtt{L}(b, \mathtt{A}(\mathtt{V}\,b, \mathtt{V}\,b)), \mathtt{V}\,c)$

Fact: $=_\alpha$ is transitive (and reflexive & symmetric). [Ex. 1]

Freshness

For each nominal set X, we can define a relation $\# \subseteq \mathbb{A} \times X$ of freshness:

$$a \# x \;\triangleq\; a \notin \operatorname{supp} x$$

more constructively,
a is fresh for x if
there is some finite
support set A for x
not containing a

Freshness

For each nominal set X, we can define a relation $\# \subseteq \mathbb{A} \times X$ of freshness:

$$a \# x \quad \triangleq \quad a \notin \operatorname{supp} x$$

- In $\mathbb{N}$, $a \# n$ always.
- In $\mathbb{A}$, $a \# b$ iff $a \neq b$.
- In Λ, $a \# t$ iff $a \notin \operatorname{fv} t$.
- In $X \times Y$, $a \# (x, y)$ iff $a \# x$ and $a \# y$.
- In $X \to_{\mathrm{fs}} Y$, $a \# f$ can be subtle!
  (and hence ditto for $\mathrm{P}_{\mathrm{fs}} X$)

Freshness Quantifier

If $\varphi(a)$ is a property of atoms $a \in \mathbb{A}$, we write $\boxed{Иa,\ \varphi(a)}$ to mean: $\{a \in \mathbb{A} \mid \neg\varphi(a)\}$ is finite, i.e. $\varphi(a)$ holds for all but finitely many a.

Freshness Quantifier

If $\varphi(a)$ is a property of atoms $a \in \mathbb{A}$, we write $\boxed{\text{И}a,\ \varphi(a)}$ to mean: $\{a \in \mathbb{A} \mid \neg\varphi(a)\}$ is finite, i.e. $\varphi(a)$ holds for all but finitely many a.

Theorem. Writing $S = \{a \in \mathbb{A} \mid \varphi(a)\}$, then t.f.a.e.
(1) $\text{И}a,\ \varphi(a)$
(2) $S \in \mathrm{P}_{\mathrm{fs}}\mathbb{A}$ and $\exists a \in \mathbb{A},\ a \# S \wedge \varphi(a)$
(3) $S \in \mathrm{P}_{\mathrm{fs}}\mathbb{A}$ and $\forall b \in \mathbb{A},\ b \# S \Rightarrow \varphi(b)$
So can read $\text{И}a,\ \varphi(a)$ as
"for some/any fresh a, $\varphi(a)$ holds"

Proof.

Freshness Quantifier

If $\varphi(a)$ is a property of atoms $a \in \mathbb{A}$, we write $\boxed{\text{И}a,\ \varphi(a)}$ to mean: $\{a \in \mathbb{A} \mid \neg\varphi(a)\}$ is finite, i.e. $\varphi(a)$ holds for all but finitely many a.

Theorem. Writing $S = \{a \in \mathbb{A} \mid \varphi(a)\}$, then t.f.a.e.

(1) $\text{И}a,\ \varphi(a)$

(2) $S \in \mathrm{P}_{\mathrm{fs}}\mathbb{A}$ and $\exists a \in \mathbb{A},\ a \mathrel{\#} S \wedge \varphi(a)$

(3) $S \in \mathrm{P}_{\mathrm{fs}}\mathbb{A}$ and $\forall b \in \mathbb{A},\ b \mathrel{\#} S \Rightarrow \varphi(b)$

So can read $\text{И}a,\ \varphi(a)$ as

"for some/any fresh a, $\varphi(a)$ holds"

Proof. If (1), then $A \triangleq \mathbb{A} - S$ is finite and necessarily supports S w.r.t. action of $\mathbb{A}$ on subsets of atoms. Since $\mathbb{A}$ is infinite and A finite, there is some $a \in S = \mathbb{A} - A$; and $a \mathrel{\#} S$ because $a \notin A$. So (2) holds.

Freshness Quantifier

If $\varphi(a)$ is a property of atoms $a \in \mathbb{A}$, we write $\boxed{\text{И}a,\ \varphi(a)}$ to mean: $\{a \in \mathbb{A} \mid \neg\varphi(a)\}$ is finite, i.e. $\varphi(a)$ holds for all but finitely many a.

Theorem. Writing $S = \{a \in \mathbb{A} \mid \varphi(a)\}$, then t.f.a.e.
(1) $\text{И}a,\ \varphi(a)$
(2) $S \in \mathrm{P}_{\mathrm{fs}}\mathbb{A}$ and $\exists a \in \mathbb{A},\ a \mathrel{\#} S \wedge \varphi(a)$
(3) $S \in \mathrm{P}_{\mathrm{fs}}\mathbb{A}$ and $\forall b \in \mathbb{A},\ b \mathrel{\#} S \Rightarrow \varphi(b)$
So can read $\text{И}a,\ \varphi(a)$ as
"for some/any fresh a, $\varphi(a)$ holds"

Proof. If (2), say $a \in S$ and $a \mathrel{\#} S$, then for any b with $b \mathrel{\#} S$, we have $(a\ b) \cdot S = S$, so $b = (a\ b) \cdot a \in (a\ b) \cdot S = S$. So (3) holds.

Freshness Quantifier

If $\varphi(a)$ is a property of atoms $a \in \mathbb{A}$, we write $\boxed{\text{И} a,\ \varphi(a)}$ to mean: $\{a \in \mathbb{A} \mid \neg\varphi(a)\}$ is finite, i.e. $\varphi(a)$ holds for all but finitely many a.

Theorem. Writing $S = \{a \in \mathbb{A} \mid \varphi(a)\}$, then t.f.a.e.
(1) $\text{И} a,\ \varphi(a)$
(2) $S \in \mathrm{P_{fs}}\mathbb{A}$ and $\exists a \in \mathbb{A},\ a \# S \wedge \varphi(a)$
(3) $S \in \mathrm{P_{fs}}\mathbb{A}$ and $\forall b \in \mathbb{A},\ b \# S \Rightarrow \varphi(b)$
So can read $\text{И} a,\ \varphi(a)$ as
"for some/any fresh a, $\varphi(a)$ holds"

Proof. If (3), then there is some finite $A \subseteq \mathbb{A}$ supporting S w.r.t. action of $\mathbb{A}$ on subsets of atoms. Since A is finite, to prove (1) it suffices to show $\mathbb{A} - S \subseteq A$, i.e. $\mathbb{A} - A \subseteq S$. But if $b \notin A$, then because A supports S, we have $b \# S$ and so by (3) we do have $b \in S$. $\square$

Name abstraction

Each $X \in \mathbf{Nom}$ yields a nominal set $\boxed{[\mathbb{A}]X}$ of

name-abstractions $\langle a\rangle x$ are $\sim$-equivalence classes of pairs $(a, x) \in \mathbb{A} \times X$, where

$$(a, x) \sim (a', x') \iff \text{И}b,\ (b\ a) \cdot x = (b\ a') \cdot x'$$

The $\mathrm{Perm}\,\mathbb{A}$-action on $[\mathbb{A}]X$ is well-defined by

$$\pi \cdot \langle a\rangle x = \langle \pi(a)\rangle(\pi \cdot x)$$

Fact: $\mathrm{supp}(\langle a\rangle x) = \mathrm{supp}\, x - \{a\}$, so that

$$b \mathrel{\#} \langle a\rangle x \iff b = a \ \vee\ b \mathrel{\#} x$$

Name abstraction

Each $X \in \mathbf{Nom}$ yields a nominal set $\boxed{[\mathbb{A}]X}$ of

name-abstractions $\langle a \rangle x$ are $\sim$-equivalence classes of pairs $(a, x) \in \mathbb{A} \times X$, where

$$(a, x) \sim (a', x') \;\Leftrightarrow\; \text{И}b,\ (b\ a) \cdot x = (b\ a') \cdot x'$$

We get a functor $[\mathbb{A}](-) : \mathbf{Nom} \to \mathbf{Nom}$ sending $f \in \mathbf{Nom}(X, Y)$ to $[\mathbb{A}]f \in \mathbf{Nom}([\mathbb{A}]X, [\mathbb{A}]Y)$ where

$$[\mathbb{A}]f\,(\langle a \rangle x) = \langle a \rangle (f\,x)$$

Name abstraction

$[\mathbb{A}](-) : \mathbf{Nom} \to \mathbf{Nom}$ is a kind of (affine) function space—it is right adjoint to the functor $\mathbb{A} \otimes (-) : \mathbf{Nom} \to \mathbf{Nom}$ sending X to $\mathbb{A} \otimes X = \{(a, x) \mid a \mathrel{\#} x\}$.

Co-unit of the adjunction is 'concretion' of an abstraction

$$\_ \mathbin{@} \_ : ([\mathbb{A}]X) \otimes \mathbb{A} \longrightarrow X$$

defined by computation rule:

$$\forall a, x, \text{И} b,\ (\langle a \rangle x) \mathbin{@} b = (b\ a) \cdot x$$

[Ex. 6]

Name abstraction

Generalising concretion, we have the following characterization of morphisms out of $[\mathbb{A}]X$

Theorem. $f \in (\mathbb{A} \times X) \to_{\mathrm{fs}} Y$ factors through the subquotient $\mathbb{A} \times X \supseteq \{(a, x) \mid a \# f\} \twoheadrightarrow [\mathbb{A}]X$ to give a unique element of $\overline{f} \in ([\mathbb{A}]X) \to_{\mathrm{fs}} Y$ satisfying

$$\mathsf{N}a, \forall x,\ \overline{f}(\langle a \rangle x) = f(a, x)$$

iff f satisfies: $\mathsf{N}a, \forall x,\ a \# f(a, x)$.

Initial algebras

- $[\mathbb{A}](-)$ has excellent exactness properties. It can be combined with $\times$, $+$ and $X \to_{\mathrm{fs}} (-)$ to give functors $\mathrm{T} : \mathbf{Nom} \to \mathbf{Nom}$ that have initial algebras $I : \mathrm{T}\, D \to D$

$$
\begin{tikzcd}
\mathrm{T}\, D \arrow[d, "I"'] & & \mathrm{T}\, X \arrow[d, "\text{for all}"', "F"] \\
D & & X
\end{tikzcd}
$$

Initial algebras

- $[\mathbb{A}](-)$ has excellent exactness properties. It can be combined with $\times$, $+$ and $X \to_{\mathrm{fs}} (-)$ to give functors $\mathrm{T} : \mathbf{Nom} \to \mathbf{Nom}$ that have initial algebras $I : \mathrm{T}\,D \to D$

$$
\begin{tikzcd}
\mathrm{T}\,D \arrow[r, dashed, "\mathrm{T}\,\hat{F}"] \arrow[d, "I"'] & \mathrm{T}\,X \arrow[d, "F"] \\
D \arrow[r, dashed, "\text{exists unique}", "\hat{F}"'] & X
\end{tikzcd}
$$

Initial algebras

- $[\mathbb{A}](-)$ has excellent exactness properties. It can be combined with $\times$, $+$ and $X \to_{\mathrm{fs}} (-)$ to give functors $\mathrm{T} : \mathbf{Nom} \to \mathbf{Nom}$ that have initial algebras $I : \mathrm{T}\, D \to D$
- For a wide class of such functors (nominal algebraic functors) the initial algebra D coincides with ASTs/α-equivalence.
  E.g. Λ is the initial algebra for

$$\mathrm{T}(-) \triangleq \mathbb{A} + (- \times -) + [\mathbb{A}](-)$$

Nominal algebraic signatures

- Sorts $\mathtt{S} \;::=\; \mathtt{N}$ name-sort (here just one, for simplicity)

$$
\begin{array}{rll}
\mathtt{S} ::= & \mathtt{N} & \text{name-sort} \\
\mid & \mathtt{D} & \text{data-sorts} \\
\mid & \mathtt{1} & \text{unit} \\
\mid & \mathtt{S}\,,\,\mathtt{S} & \text{pairs} \\
\mid & \mathtt{N}\,.\,\mathtt{S} & \text{name-binding}
\end{array}
$$

- Typed operations $\mathtt{op} : \mathtt{S} \to \mathtt{D}$

Signature Σ is specified by the stuff in red.

Nominal algebraic signatures

Example: λ-calculus

name-sort $\mathtt{Var}$ for variables, data-sort $\mathtt{Term}$ for terms, and operations

$$\mathtt{V} : \mathtt{Var} \longrightarrow \mathtt{Term}$$
$$\mathtt{A} : \mathtt{Term} \, , \mathtt{Term} \longrightarrow \mathtt{Term}$$
$$\mathtt{L} : \mathtt{Var} \, . \, \mathtt{Term} \longrightarrow \mathtt{Term}$$

Nominal algebraic signatures

Example: π-calculus

name-sort $\mathtt{Chan}$ for channel names, data-sorts $\mathtt{Proc}$, $\mathtt{Pre}$ and $\mathtt{Sum}$ for processes, prefixed processes and summations, and operations

$$
\begin{aligned}
\mathtt{S} &: \mathtt{Sum} \longrightarrow \mathtt{Proc}\\
\mathtt{Comp} &: \mathtt{Proc}\,,\mathtt{Proc} \longrightarrow \mathtt{Proc}\\
\mathtt{Nu} &: \mathtt{Chan}\,.\,\mathtt{Proc} \longrightarrow \mathtt{Proc}\\
\mathtt{!} &: \mathtt{Proc} \longrightarrow \mathtt{Proc}\\
\mathtt{P} &: \mathtt{Pre} \longrightarrow \mathtt{Sum}\\
\mathtt{O} &: 1 \longrightarrow \mathtt{Sum}\\
\mathtt{Plus} &: \mathtt{Sum}\,,\mathtt{Sum} \longrightarrow \mathtt{Sum}\\
\mathtt{Out} &: \mathtt{Chan}\,,\mathtt{Chan}\,,\mathtt{Proc} \longrightarrow \mathtt{Pre}\\
\mathtt{In} &: \mathtt{Chan}\,,(\mathtt{Chan}\,.\,\mathtt{Proc}) \longrightarrow \mathtt{Pre}\\
\mathtt{Tau} &: \mathtt{Proc} \longrightarrow \mathtt{Pre}\\
\mathtt{Match} &: \mathtt{Chan}\,,\mathtt{Chan}\,,\mathtt{Pre} \longrightarrow \mathtt{Pre}
\end{aligned}
$$

Nominal algebraic signatures

Closely related notions:

- *binding signatures* of Fiore, Plotkin & Turi (LICS 1999)
- *nominal algebras* of Honsell, Miculan & Scagnetto (ICALP 2001)

N.B. all these notions of signature restrict attention to iterated, but *unary* name-binding—there are other kinds of lexically scoped binder (e.g. see Pottier's Cαml language, or Blanchette *et al* POPL 2019.)

$\Sigma(\mathtt{S})$ = raw terms over Σ of sort $\mathtt{S}$

$$\frac{a \in \mathbb{A}}{a \in \Sigma(\mathtt{N})} \qquad \frac{t \in \Sigma(\mathtt{S}) \qquad \mathtt{op} : \mathtt{S} \to \mathtt{D}}{\mathtt{op}\, t \in \Sigma(\mathtt{D})} \qquad \frac{}{() \in \Sigma(\mathtt{1})}$$

$$\frac{t_1 \in \Sigma(\mathtt{S}_1) \qquad t_2 \in \Sigma(\mathtt{S}_2)}{t_1 \,,\, t_2 \in \Sigma(\mathtt{S}_1 \,,\, \mathtt{S}_2)} \qquad \frac{a \in \mathbb{A} \qquad t \in \Sigma(\mathtt{S})}{a \,.\, t \in \Sigma(\mathtt{N} \,.\, \mathtt{S})}$$

Each $\Sigma(\mathtt{S})$ is a nominal set once equipped with the obvious $\mathrm{Perm}\,\mathbb{A}$-action—any finite set of atoms containing all those occurring in t supports $t \in \Sigma(\mathtt{S})$.

Alpha-equivalence $=_\alpha \subseteq \Sigma(\mathrm{S}) \times \Sigma(\mathrm{S})$

$$\frac{a \in \mathbb{A}}{a =_\alpha a} \qquad \frac{t =_\alpha t'}{\mathsf{op}\, t =_\alpha \mathsf{op}\, t'} \qquad \frac{}{() =_\alpha ()}$$

$$\frac{t_1 =_\alpha t_1' \qquad t_2 =_\alpha t_2'}{t_1\, ,\, t_2 =_\alpha t_1'\, ,\, t_2'}$$

$$\frac{(a_1\ a) \cdot t_1 =_\alpha (a_2\ a) \cdot t_2 \qquad a \# (a_1, t_1, a_2, t_2)}{a_1\,.\,t_1 =_\alpha a_2\,.\,t_2}$$

Alpha-equivalence $=_\alpha \subseteq \Sigma(\mathrm{S}) \times \Sigma(\mathrm{S})$

Fact: $=_\alpha$ is equivariant ($t_1 =_\alpha t_2 \;\Rightarrow\; \pi \cdot t_1 =_\alpha \pi \cdot t_2$) and each quotient

$$\Sigma_\alpha(\mathrm{S}) \triangleq \{[t]_\alpha \mid t \in \Sigma(\mathrm{S})\}$$

is a nominal set with

$$\begin{array}{rcl} \pi \cdot [t]_\alpha & = & [\pi \cdot t]_\alpha \\ \mathrm{supp}\,[t]_\alpha & = & \mathit{fn}\,t \\ & \text{where} & \\ \mathit{fn}(a \,.\, t) & = & \mathit{fn}\,t - \{a\} \\ \mathit{fn}(t_1 \,,\, t_2) & = & \mathit{fn}\,t_1 \cup \mathit{fn}\,t_2 \\ & \text{etc.} & \end{array}$$

Theorem. Given a nominal algebraic signature Σ (for simplicity, assume Σ has a single data-sort $\mathtt{D}$ as well as a single name-sort $\mathtt{N}$)
$\Sigma_\alpha(\mathtt{D})$ is an initial algebra for the associated functor $T_\Sigma : \mathbf{Nom} \longrightarrow \mathbf{Nom}$.

Theorem. Given a nominal algebraic signature Σ (for simplicity, assume Σ has a single data-sort $\mathtt{D}$ as well as a single name-sort $\mathtt{N}$)
$\Sigma_\alpha(\mathtt{D})$ is an initial algebra for the associated functor $T_\Sigma : \mathbf{Nom} \longrightarrow \mathbf{Nom}$.

$$T_\Sigma(-) = [\![\mathtt{S}_1]\!](-) + \cdots + [\![\mathtt{S}_n]\!](-)$$

where Σ has operations $\mathtt{op}_i : \mathtt{S}_i \longrightarrow D$ $(i = 1..n)$

and $[\![\mathtt{S}]\!](-) : \mathbf{Nom} \longrightarrow \mathbf{Nom}$ is defined by:

$$\begin{aligned}
[\![\mathtt{N}]\!](-) &= \mathbb{A} \\
[\![\mathtt{D}]\!](-) &= (-) \\
[\![\mathtt{1}]\!](-) &= 1 \\
[\![\mathtt{S}_1 \,,\, \mathtt{S}_2]\!](-) &= [\![\mathtt{S}_1]\!](-) \times [\![\mathtt{S}_2]\!](-) \\
[\![\mathtt{N} \,.\, \mathtt{S}]\!](-) &= [\mathbb{A}]([\![\mathtt{S}]\!](-))
\end{aligned}$$

Theorem. Given a nominal algebraic signature Σ (for simplicity, assume Σ has a single data-sort $\mathtt{D}$ as well as a single name-sort $\mathtt{N}$)
$\Sigma_\alpha(\mathtt{D})$ is an initial algebra for the associated functor $T_\Sigma : \mathbf{Nom} \longrightarrow \mathbf{Nom}$.

E.g. for the λ-calculus signature with operations

$\mathtt{V} : \mathtt{Var} \longrightarrow \mathtt{Term}$

$\mathtt{A} : \mathtt{Term}\,,\mathtt{Term} \longrightarrow \mathtt{Term}$

$\mathtt{L} : \mathtt{Var}\,.\,\mathtt{Term} \longrightarrow \mathtt{Term}$

we have

$T_\Sigma(-) = \mathbb{A} + (- \times -) + [\mathbb{A}](-)$

α-Structural recursion

For λ-terms:

Theorem.
Given any $X \in \mathbf{Nom}$ and $\begin{cases} f_1 & \in & \mathbb{A} \to_{\mathrm{fs}} X \\ f_2 & \in & X \times X \to_{\mathrm{fs}} X \\ f_3 & \in & [\mathbb{A}]X \to_{\mathrm{fs}} X \end{cases}$

$\exists !\ \hat{f} \in \Lambda \to_{\mathrm{fs}} X$ s.t. $\begin{cases} \hat{f}\ a = f_1\ a \\ \hat{f}\ (e_1\ e_2) = f_2(\hat{f}\ e_1, \hat{f}\ e_2) \\ \hat{f}(\lambda a.e) = f_3(\langle a \rangle(\hat{f}\ e)) \quad \text{if } a \mathbin{\#} (f_1, f_2, f_3) \end{cases}$

The enriched functor $[\mathbb{A}](-) : \mathbf{Nom} \to \mathbf{Nom}$ sends $f \in X \to_{\mathrm{fs}} Y$ to $[\mathbb{A}]f \in [\mathbb{A}]X \to_{\mathrm{fs}} [\mathbb{A}]Y$ where

$$[\mathbb{A}]f\ (\langle a \rangle x) = \langle a \rangle (f\ x) \quad \text{if } a \mathbin{\#} f$$

Recall: Name abstraction

Theorem. $f \in (\mathbb{A} \times X) \to_{\mathrm{fs}} Y$ factors through the subquotient $\mathbb{A} \times X \supseteq \{(a, x) \mid a \# f\} \twoheadrightarrow [\mathbb{A}]X$ to give a unique element of $\overline{f} \in ([\mathbb{A}]X) \to_{\mathrm{fs}} Y$ satisfying

$$\text{И} a, \forall x,\ \overline{f}(\langle a \rangle x) = f(a, x)$$

iff f satisfies: $\text{И} a, \forall x,\ a \# f(a, x)$.

α-Structural recursion

For λ-terms:

Theorem.
Given any $X \in \mathbf{Nom}$ and $\begin{cases} f_1 & \in & \mathbb{A} \to_{\mathrm{fs}} X \\ f_2 & \in & X \times X \to_{\mathrm{fs}} X \\ f_3 & \in & \mathbb{A} \times X \to_{\mathrm{fs}} X \end{cases}$ s.t.

$$\text{И} a, \forall x,\ a \# f_3(a, x) \qquad \text{(FCB)}$$

$\exists !\ \hat{f} \in \Lambda \to_{\mathrm{fs}} X$ s.t. $\begin{cases} \hat{f}\, a = f_1\, a \\ \hat{f}\,(e_1\, e_2) = f_2(\hat{f}\, e_1, \hat{f}\, e_2) \\ \hat{f}(\lambda a.e) = f_3(a, \hat{f}\, e) \quad \text{if } a \# (f_1, f_2, f_3) \end{cases}$

α-Structural recursion

For λ-terms:

Theorem.
Given any $X \in \mathbf{Nom}$ and $\begin{cases} f_1 & \in & \mathbb{A} \to_{\mathrm{fs}} X \\ f_2 & \in & X \times X \to_{\mathrm{fs}} X \\ f_3 & \in & \mathbb{A} \times X \to_{\mathrm{fs}} X \end{cases}$ s.t.

$$\text{И} a, \forall x,\ a \mathrel{\#} f_3(a, x) \qquad \text{(FCB)}$$

$\exists!\ \hat{f} \in \Lambda \to_{\mathrm{fs}} X$ s.t. $\begin{cases} \hat{f}\, a = f_1\, a \\ \hat{f}\,(e_1\, e_2) = f_2(\hat{f}\, e_1, \hat{f}\, e_2) \\ \hat{f}(\lambda a.e) = f_3(a, \hat{f}\, e) \quad \text{if } a \mathrel{\#} (f_1, f_2, f_3) \end{cases}$

E.g. capture-avoiding substitution $(-)[e'/a'] : \Lambda \to \Lambda$ is the $\hat{f}$ for

$$\begin{array}{rcl} f_1\, a & \triangleq & \text{if } a = a' \text{ then } e' \text{ else } a \\ f_2(e_1, e_2) & \triangleq & e_1\, e_2 \\ f_3(a, e) & \triangleq & \lambda a.e \end{array}$$

for which (FCB) holds, since $a \mathrel{\#} \lambda a.e$

α-Structural recursion

For λ-terms:

Theorem.
Given any $X \in \mathbf{Nom}$ and $\begin{cases} f_1 & \in & \mathbb{A} \to_{\mathrm{fs}} X \\ f_2 & \in & X \times X \to_{\mathrm{fs}} X \\ f_3 & \in & \mathbb{A} \times X \to_{\mathrm{fs}} X \end{cases}$ s.t.

$$\mathsf{N}a, \forall x,\ \ a \# f_3(a, x) \qquad \text{(FCB)}$$

$$\exists!\ \hat{f} \in \Lambda \to_{\mathrm{fs}} X \text{ s.t. } \begin{cases} \hat{f}\ a = f_1\ a \\ \hat{f}\ (e_1\ e_2) = f_2(\hat{f}\ e_1, \hat{f}\ e_2) \\ \hat{f}(\lambda a.e) = f_3(a, \hat{f}\ e) \quad \text{if } \ a \# (f_1, f_2, f_3) \end{cases}$$

Non-example: trying to list the bound variables of a λ-term

$$\begin{array}{rcl} f_1\ a & \triangleq & \mathsf{nil} \\ f_2(\ell_1, \ell_2) & \triangleq & \ell_1 \mathbin{@} \ell_2 \\ f_3(a, \ell) & \triangleq & a :: \ell \end{array}$$

for which (FCB) does not hold, since $a \in \mathrm{supp}(a :: \ell)$.

α-Structural recursion

For λ-terms:

Theorem.
Given any $X \in \mathbf{Nom}$ and $\begin{cases} f_1 & \in & \mathbb{A} \to_{\mathrm{fs}} X \\ f_2 & \in & X \times X \to_{\mathrm{fs}} X \\ f_3 & \in & \mathbb{A} \times X \to_{\mathrm{fs}} X \end{cases}$ s.t.

$$\text{И} a, \forall x,\ a \mathbin{\#} f_3(a,x) \qquad \text{(FCB)}$$

$\exists!\ \hat{f} \in \Lambda \to_{\mathrm{fs}} X$ s.t. $\begin{cases} \hat{f}\, a = f_1\, a \\ \hat{f}\,(e_1\, e_2) = f_2(\hat{f}\, e_1, \hat{f}\, e_2) \\ \hat{f}(\lambda a.e) = f_3(a, \hat{f}\, e) \quad \text{if } a \mathbin{\#} (f_1, f_2, f_3) \end{cases}$

Similar results hold for any nominal algebraic signature—see J ACM 53(2006)459–506.

Implemented in Urban & Berghofer's Nominal package for Isabelle/HOL (classical higher-order logic).

Seems to capture informal usage well, but (FCB) can be tricky…

Counting occurrences of bound variables

For each $e \in \Lambda$, $\boxed{\mathrm{cbv}\, e \triangleq f\, e\, \rho_0 \in \mathbb{N}}$

where we want $f \in \Lambda \to_{\mathrm{fs}} X$ with $X = (\mathbb{A} \to_{\mathrm{fs}} \mathbb{N}) \to_{\mathrm{fs}} \mathbb{N}$ to satisfy

$$
\begin{array}{rcl}
f\, a\, \rho & = & \rho\, a \\
f\, (e_1\, e_2)\, \rho & = & (f\, e_1 \rho) + (f\, e_2\, \rho) \\
f (\lambda a.e)\, \rho & = & f\, e\, (\rho[a \mapsto 1])
\end{array}
$$

and where $\rho_0 \in \mathbb{A} \to_{\mathrm{fs}} \mathbb{N}$ is $\lambda(a \in \mathbb{A}) \to 0$.

E.g. when $e = (\lambda a.\, \lambda b.\, a)\, b$ (with $a \neq b$), then e has a single occurrence of a bound variable (called a) and $\mathrm{cbv}\, e = 1$.

Counting occurrences of bound variables

For each $e \in \Lambda$, $\boxed{\mathrm{cbv}\, e \triangleq f\, e\, \rho_0 \in \mathbb{N}}$

where we want $f \in \Lambda \to_{\mathrm{fs}} X$ with $X = (\mathbb{A} \to_{\mathrm{fs}} \mathbb{N}) \to_{\mathrm{fs}} \mathbb{N}$ to satisfy

$$
\begin{array}{rcl}
f\, a\, \rho & = & \rho\, a \\
f\, (e_1\, e_2)\, \rho & = & (f\, e_1 \rho) + (f\, e_2\, \rho) \\
f(\lambda a.e)\, \rho & = & f\, e\, (\rho[a \mapsto 1])
\end{array}
$$

and where $\rho_0 \in \mathbb{A} \to_{\mathrm{fs}} \mathbb{N}$ is $\lambda(a \in \mathbb{A}) \to 0$.

Looks like we should take $f_3(a, x) = \lambda(\rho \in \mathbb{A} \to_{\mathrm{fs}} \mathbb{N}) \to x(\rho[a \mapsto 1])$, *but this does not satisfy* (FCB). Solution: take X to be a certain nominal subset of $(\mathbb{A} \to_{\mathrm{fs}} \mathbb{N}) \to_{\mathrm{fs}} \mathbb{N}$. [See Nominal Sets book, Example 8.20]