MFPS–LICS Special Session Honouring Dana Scott

Symmetric Scott

Andrew Pitts

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

80 years of Dana Scott

- automata theory
- set theory
- sheaves & logic
- lambda calculus
- domain theory
- $\cdots$

Symmetric Scottery

- automata theory
- set theory
- sheaves & logic
- lambda calculus
- domain theory
- $\cdots$

Mathematics of group actions allows MFPS & LICS access to two related, interesting and useful notions of finiteness:

finite support and orbit-finiteness

Symmetric Scottery

- automata theory ← for languages over infinite alphabets
- set theory ← Fraenkel-Mostowski sets
- sheaves & logic ← atomic toposes (e.g. nominal sets)
- lambda calculus ← calculi for name abstraction & locally scoped names
- domain theory ← nominal Scott domains
- ...

Mathematics of group actions allows MFPS & LICS access to two related, interesting and useful notions of finiteness:

finite support and orbit-finiteness

Orbit-finiteness example

Infinitely many names $a \in \mathbb{A}$.

For properties that are that are equivariant

$$\varphi(a_1, \ldots, a_n) \Rightarrow \varphi(\pi\, a_1, \ldots, \pi\, a_n)$$

with respect to permutations $\pi : \mathbb{A} \cong \mathbb{A}$,

Orbit-finiteness example

Infinitely many names $a \in \mathbb{A}$.

For properties that are that are equivariant

$$\varphi(a_1, \ldots, a_n) \Rightarrow \varphi(\pi\, a_1, \ldots, \pi\, a_n)$$

with respect to permutations $\pi : \mathbb{A} \cong \mathbb{A}$,

the validity of $\exists a_1, \ldots, a_n.\, \varphi(a_1, \ldots, a_n)$

is equivalent to the validity of a finite disjunction of instances of φ,

because $\mathbb{A}^n$ has only finitely many orbits.

e.g. orbits of $\mathbb{A}^2$ are $\{(a, a) \mid a \in \mathbb{A}\}$ and $\{(a, b) \mid a \neq b \in \mathbb{A}\}$

Orbit-finiteness example

Infinitely many names $a \in \mathbb{A}$.

For properties that are that are equivariant

$$\varphi(a_1, \dots, a_n) \Rightarrow \varphi(\pi\, a_1, \dots, \pi\, a_n)$$

with respect to permutations $\pi : \mathbb{A} \cong \mathbb{A}$,

the validity of $\exists a_1, \dots, a_n.\, \varphi(a_1, \dots, a_n)$

is equivalent to the validity of a <u>finite</u> disjunction of instances of φ.

Orbit-finiteness $\rightsquigarrow$ π-calculus model-checking with HD-automata [Montanari & Pistori, MFCS 2000]

automata theory for infinite alphabets [Bojańczyk *et al*, LICS 2011]

Symmetric Scottery

- automata theory
- set theory
- sheaves & logic
- lambda calculus
- domain theory
- $\cdots$

Mathematics of group actions allows MFPS & LICS access to two related, interesting and useful notions of finiteness:

finite support and orbit-finiteness

Finite support example

- Infinitely many names $\mathbb{A} = \{a_0, a_1, \ldots\}$.
  Booleans $\mathbf{2} = \{\mathbf{0}, \mathbf{1}\}$.
- Flat domains $\mathbb{A}_\perp$, $\mathbf{2}_\perp$.
- Existential quantifier
  $f \in (\mathbb{A}_\perp \to \mathbf{2}_\perp) \mapsto$

$$\mathbf{exists}\, f \triangleq \begin{cases} \mathbf{1} & \text{if } (\exists a \in \mathbb{A})\ f\ a = \mathbf{1} \\ \mathbf{0} & \text{if } (\forall a \in \mathbb{A})\ f\ a = \mathbf{0} \\ \perp & \text{otherwise} \end{cases}$$

Finite support example

- Infinitely many names $\mathbb{A} = \{a_0, a_1, \ldots\}$.
  Booleans $\mathbf{2} = \{0, 1\}$.
- Flat domains $\mathbb{A}_\bot$, $\mathbf{2}_\bot$.
- Existential quantifier
  $f \in (\mathbb{A}_\bot \to \mathbf{2}_\bot) \mapsto$

$$\mathbf{exists}\, f \triangleq \begin{cases} 1 & \text{if } (\exists a \in \mathbb{A})\ f\, a = 1 \\ 0 & \text{if } (\forall a \in \mathbb{A})\ f\, a = 0 \\ \bot & \text{otherwise} \end{cases}$$

does not give a continuous function $\mathbf{exists} : (\mathbb{A}_\bot \to \mathbf{2}_\bot) \to \mathbf{2}_\bot$

e.g. consider limit of $f_n : a_i \mapsto \begin{cases} 0 & i < n \\ \bot & i \geq n \end{cases}$

Finite support example

- Infinitely many names $\mathbb{A} = \{a_0, a_1, \ldots\}$.
  Booleans $\mathbf{2} = \{\mathbf{0}, \mathbf{1}\}$.
- Flat domains $\mathbb{A}_\perp$, $\mathbf{2}_\perp$.
- Existential quantifier
  $f \in (\mathbb{A}_\perp \to \mathbf{2}_\perp) \mapsto$
  $$\mathbf{exists}\, f \triangleq \begin{cases} \mathbf{1} & \text{if } (\exists a \in \mathbb{A})\ f\, a = \mathbf{1} \\ \mathbf{0} & \text{if } (\forall a \in \mathbb{A})\ f\, a = \mathbf{0} \\ \perp & \text{otherwise} \end{cases}$$

  does <u>not</u> give a continuous function $\mathbf{exists} : (\mathbb{A}_\perp \to \mathbf{2}_\perp) \to \mathbf{2}_\perp$

  but it does when restricted to finitely supported functions...

Finite support

- Infinitely many names $a \in \mathbb{A}$.
- Sets equipped with an action $\pi, x \mapsto \pi \cdot x$

$$\mathbf{id} \cdot x = x$$
$$\pi' \cdot (\pi \cdot x) = (\pi' \circ \pi) \cdot x$$

of (finite) permutations $\pi : \mathbb{A} \cong \mathbb{A}$

Finite support

- Infinitely many names $a \in \mathbb{A}$.
- <u>Nominal set</u> = set D equipped with a permutation action for which each $x \in D$ possess a finite support $A \subseteq_{\mathbf{fin}} \mathbb{A}$:

$$(\forall \pi)\ ((\forall a \in A)\ \pi\, a = a)\ \Rightarrow\ \pi \cdot x = x$$

if there is such an A, there's a least one, written $\mathbf{supp}\, x$

Finite support

- Infinitely many names $a \in \mathbb{A}$.
- Nominal set = set D equipped with a permutation action for which each $x \in D$ possess a finite support
- Category of nominal sets & action-preserving functions is a well-known (2-valued) topos.

Exponentials: $D \to_{\mathrm{fs}} E \triangleq$ all functions $f : D \to E$ that are finitely supported w.r.t. action $\pi \cdot f : x \mapsto \pi \cdot (f(\pi^{-1} \cdot x))$

group inverses make exponentials much simpler than for more general monoid/category actions

Finite support example

- Infinitely many names $\mathbb{A} = \{a_0, a_1, \ldots\}$.
  Booleans $\mathbf{2} = \{0, 1\}$.
- Flat domains $\mathbb{A}_\perp$, $\mathbf{2}_\perp$.
- Existential quantifier

$$\mathbf{exists}\, f \triangleq \begin{cases} 1 & \text{if } (\exists a \in \mathbb{A})\ f\ a = 1 \\ 0 & \text{if } (\forall a \in \mathbb{A})\ f\ a = 0 \\ \perp & \text{otherwise} \end{cases}$$

is a continuous function $\mathbf{exists} : (\mathbb{A}_\perp \to_{\mathbf{fs}} \mathbf{2}_\perp) \to \mathbf{2}_\perp$,
because $(\forall a \in \mathbb{A})\ f\ a = 0$ if $(\forall a \in A)\ f\ a = 0$, where $A = \mathbf{supp}\, f \uplus \{a\}$ is finite.

Nominal Scott Domains

[Turner & Winskel, CSL 2009] [Lösch & Pitts, POPL 2013]

- Posets in topos of nominal sets.
- Require limits/continuity only for directed sets whose elements have a common finite support ('uniform-directed' subsets).

  Associated notion of compactness is more liberal than classical one – replace 'finite' by 'orbit-finite'

e.g. compact functions are joins of orbit-finite consistent sets of step functions

Nominal Scott Domains

[Turner & Winskel, CSL 2009] [Lösch & Pitts, POPL 2013]

- Posets in topos of nominal sets.
- Require limits/continuity only for directed sets whose elements have a common finite support ('uniform-directed' subsets).
- Category of NSDs is cartesian closed and has fixpoint recursion for both morphisms and objects.

  But it also models name abstraction and locally scoped names.

Nominal Scott Domains

[Turner & Winskel, CSL 2009] [Lösch & Pitts, POPL 2013]

- Posets in topos of nominal sets.
- Require limits/continuity only for directed sets whose elements have a common finite support ('uniform-directed' subsets).
- Category of NSDs is cartesian closed and has fixpoint recursion for both morphisms and objects.

  But it also models name abstraction and locally scoped names.

NSD $[\mathbb{A}]D = \{\langle a\rangle d \mid a \in \mathbb{A} \wedge d \in D\}$
where $\langle a\rangle d = \langle a'\rangle d'$ iff
$(a\ b) \cdot d = (a'\ b) \cdot d'$ for some/any $b \notin \mathbf{supp}(a, d, a', d')$

Nominal Scott Domains

[Turner & Winskel, CSL 2009] [Lösch & Pitts, POPL 2013]

- Posets in topos of nominal sets.
- Require limits/continuity only for directed sets whose elements have a common finite support ('uniform-directed' subsets).
- Category of NSDs is cartesian closed and has fixpoint recursion for both morphisms and objects.

  But it also models name abstraction and locally scoped names.

using morphisms $\nu : [\mathbb{A}]D \to D$ satisfying
$\nu(\langle a\rangle d) = d$ if $a \notin \mathbf{supp}\, d$
$\nu\langle a\rangle(\nu(\langle b\rangle d)) = \nu(\langle b\rangle(\nu\langle a\rangle d))$

Higher-order computable functions with local names

Plotkin's **PCF**

Programming language for **C**omputable **F**unctions:
simply typed λ-calculus over ground types `bool` & `nat`,
with arithmetic and boolean operations and fixpoint recursion.
[Plotkin, *LCF Considered as a Programming Language*, TCS 5(1977)223–255]

Higher-order computable functions with local names

PCFA = Plotkin's **PCF** extended with a type `name`.

Denotational semantics using nominal Scott domains:

$$
\begin{aligned}
[\![\mathtt{name}]\!] &\triangleq \mathbb{A}_\bot \\
[\![\mathtt{bool}]\!] &\triangleq \mathbf{2}_\bot \\
[\![\mathtt{nat}]\!] &\triangleq \mathbb{N}_\bot \\
[\![\tau \to \tau']\!] &\triangleq [\![\tau]\!] \to_{\mathbf{fs}} [\![\tau']\!]
\end{aligned}
$$

supports the interpretation of terms for name-equality test, name-swapping and (Odersky-style) locally scoped names, $\nu a. e$.

Local scoping example

[suggested by Tzevelekos]

$$\mathtt{F_1} \triangleq \lambda q : (\mathtt{name} \to \mathtt{bool}) \to \mathtt{bool}.\ \nu a. q\,\mathtt{eq}_a$$
$$\mathtt{F_2} \triangleq \lambda q : (\mathtt{name} \to \mathtt{bool}) \to \mathtt{bool}.\ q\,\mathtt{k_F}$$

are contextually equivalent **PCFA** terms of type

$$((\mathtt{name} \to \mathtt{bool}) \to \mathtt{bool}) \to \mathtt{bool}$$

where

$$\begin{cases} \mathtt{eq}_a & \triangleq \lambda x : \mathtt{name}.\ \mathtt{if}\ x = a\ \mathtt{then\ T\ else\ F} \\ \mathtt{k_F} & \triangleq \lambda x : \mathtt{name}.\ \mathtt{F} \end{cases}$$

Local scoping example

[suggested by Tzevelekos]

$$F_1 \triangleq \lambda q : (\mathtt{name} \to \mathtt{bool}) \to \mathtt{bool}.\ \nu a. q\, \mathtt{eq}_a$$
$$F_2 \triangleq \lambda q : (\mathtt{name} \to \mathtt{bool}) \to \mathtt{bool}.\ q\, \mathtt{k_F}$$

are contextually equivalent **PCFA** terms of type

$$((\mathtt{name} \to \mathtt{bool}) \to \mathtt{bool}) \to \mathtt{bool}$$

where

$$\begin{cases} \mathtt{eq}_a & \triangleq \lambda x : \mathtt{name}.\ \mathtt{if}\ x = a\ \mathtt{then}\ \mathtt{T}\ \mathtt{else}\ \mathtt{F} \\ \mathtt{k_F} & \triangleq \lambda x : \mathtt{name}.\ \mathtt{F} \end{cases}$$

but $[\![F_1]\!] \neq [\![F_2]\!]$, because $\begin{cases} [\![F_1]\!]\ \textbf{exists} & = \mathbf{1} \\ [\![F_2]\!]\ \textbf{exists} & = \mathbf{0} \end{cases}$

Full abstraction for $\mathbf{PCFA}^{+}$

Plotkin's classic full abstraction result for $\mathbf{PCF} + \texttt{por}$:

contextual preorder (operational)		information order (denotational)
$e \leq_{\mathbf{ctx}} e' : \tau$	$\Leftrightarrow$	$[\![e]\!] \sqsubseteq [\![e']\!]$

Full abstraction for $\mathbf{PCFA}^+$

Plotkin's classic full abstraction result for $\mathbf{PCF} + \mathtt{por}$:

contextual preorder (operational) — information order (denotational)

$$e \leq_{\mathbf{ctx}} e' : \tau \quad \Leftrightarrow \quad [\![e]\!] \sqsubseteq [\![e']\!]$$

Theorem. [Lösch & Pitts, POPL 2013]
The NSD model is fully abstract for $\mathbf{PCFA}^+ = \mathbf{PCFA}$ extended with

parallel or	$\mathtt{por} : \mathtt{bool} \to \mathtt{bool} \to \mathtt{bool}$
exists name	$\mathtt{exists} : (\mathtt{name} \to \mathtt{bool}) \to \mathtt{bool}$
definite name description	$\mathtt{the} : (\mathtt{name} \to \mathtt{bool}) \to \mathtt{name}$

$$\mathbf{the}\, f \triangleq \begin{cases} \text{unique } a \text{ s.t. } f\,a = \mathbf{1}, \text{ if it exists} \\ \bot, \text{ otherwise} \end{cases}$$

Full abstraction for $\mathbf{PCFA}^+$

Plotkin's classic full abstraction result for $\mathbf{PCF} + \mathtt{por}$:

contextual preorder (operational) information order (denotational)

$$e \leq_{\mathrm{ctx}} e' : \tau \quad \Leftrightarrow \quad [\![e]\!] \sqsubseteq [\![e']\!]$$

Theorem. [Lösch & Pitts, POPL 2013]
The NSD model is fully abstract for $\mathbf{PCFA}^+ = \mathbf{PCFA}$ extended with

parallel or	$\mathtt{por} : \mathtt{bool} \to \mathtt{bool} \to \mathtt{bool}$
exists name	$\mathtt{exists} : (\mathtt{name} \to \mathtt{bool}) \to \mathtt{bool}$
definite name description	$\mathtt{the} : (\mathtt{name} \to \mathtt{bool}) \to \mathtt{name}$

Proof has novel aspects (use of retracts – thanks Dana).

Symmetric Scottery

- automata theory ← for languages over infinite alphabets
- set theory ← Fraenkel-Mostowski sets
- sheaves & logic ← atomic toposes (e.g. nominal sets)
- lambda calculus ← calculi for name abstraction & locally scoped names
- domain theory ← nominal Scott domains
- ...

Mathematics of group actions allows MFPS & LICS access to two related, interesting and useful notions of finiteness:

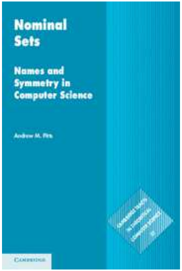
finite support and orbit-finiteness

Finite Support and Orbit-Finiteness

- Can make 'finite support' automatic by working in choice-free classical HOL/set theory.
- We understand 'orbit-finite' category-theoretically (= finitely presentable), but don't really understand its logical status: how to predict where to replace 'finite' by 'orbit-finite' in computation theory?
- Permutations of $\mathbb{A}$ (= name-inequality symmetry) is not the only group of interest – useful to consider automorphisms of various relational structures on $\mathbb{A}$ (linear orders, undirected graphs, . . .).

  See recent work of Bojańczyk *et al*.
- For much more on nominal sets, see. . .

Commercial break



Nominal Sets

Names and Symmetry in Computer Science

Cambridge Tracts in Theoretical Computer Science, Vol. 57 (CUP, 2013)