Free-Algebra Models for the π-Calculus

Ian Stark

Laboratory for Foundations of Computer Science
School of Informatics
University of Edinburgh

Summary

The finite π-calculus has an explicit set-theoretic functor-category model that is known to be fully-abstract for strong late bisimulation congruence [Fiore, Moggi, Sangiorgi]

We can characterise this as the initial free algebra for certain operations and equations in the setting of Power and Plotkin's enriched Lawvere theories.

This combines separate theories of nondeterminism, I/O and name creation in a modular fashion. As a bonus, we get a whole category of models, a modal logic and a computational monad. The tricky part is that everything has to happen inside the functor category $\mathcal{S}et^{\mathcal{I}}$.

Overview

- Equational theories for different features of computation.
- Enrichment over the functor category $\mathcal{S}et^{\mathcal{I}}$.
- A theory of π.
- Free-algebra models; full abstraction; modal logic.

Nondeterministic computation

Operations

$$\begin{aligned} \mathsf{choice} : A^2 &\longrightarrow A \\ \mathsf{nil} : \ 1 &\longrightarrow A \end{aligned}$$

Equations

$$\mathsf{choice}(P, Q) = \mathsf{choice}(Q, P)$$

$$\mathsf{choice}(\mathsf{nil}, P) = \mathsf{choice}(P, P) = P$$

$$\mathsf{choice}(P, (\mathsf{choice}(Q, R)) = \mathsf{choice}(\mathsf{choice}(P, Q), R)$$

Algebras for nondeterminism

For any Cartesian category $\mathcal{C}$ we can form the category $\mathcal{ND}(\mathcal{C})$ of models $(A, \mathsf{choice}, \mathsf{nil})$ for the theory. In particular, there is:

$$\mathcal{ND}(\mathcal{S}et)$$

free $\quad F \quad \dashv \quad U \quad$ forgetful

$$\mathcal{S}et$$

In fact $(U \circ F)$ is finite powerset and the adjunction is monadic: $\mathcal{ND}(\mathcal{S}et)$ is isomorphic to the category of $\mathcal{P}_{\mathsf{fin}}$-algebras.

Computational monad for nondeterminism

$$\mathcal{ND}(Set)$$

free $\quad F \quad \dashv \quad U \quad$ forgetful

$$Set$$

The composition $T = (U \circ F) = \mathcal{P}_{fin}$ is the computational monad for finite nondeterminism. Operations $choice$ and nil then induce generic effects in the Kleisli category:

from $\quad choice : A^2 \longrightarrow A^1 \qquad$ we get $\quad arb : 1 \longrightarrow T\,2$

$\quad nil : A^0 \longrightarrow A^1 \qquad deadlock : 1 \longrightarrow T\,0$

[Plotkin, Power: Algebraic Operations and Generic Effects]

I/O computation

Operations

$$\mathrm{in} : A^V \longrightarrow A$$
$$\mathrm{out} : A \longrightarrow A^V$$

Equations

none

From any Cartesian $\mathcal{C}$ we form the category $\mathcal{IO}(\mathcal{C})$ of models $(A, \mathrm{in}, \mathrm{out})$ for I/O computation over $\mathcal{C}$.

I/O adjunction and monad

$$\mathcal{IO}(\mathit{Set})$$

free $\quad F \quad \dashv \quad U \quad$ forgetful

$$\mathit{Set}$$

The adjunction is monadic: $\mathcal{IO}(\mathit{Set}) \cong T\text{-}\mathcal{A}lg$ for the resumptions monad, the computational monad for I/O:

$$T(X) = \mu Y.(X + Y^V + Y \times V) \ .$$

The operations induce suitable effects in its Kleisli category:

from $\quad in : A^V \longrightarrow A^1 \qquad$ we get $\quad read : 1 \longrightarrow T\,V$

$\qquad out : A^1 \longrightarrow A^V \qquad\qquad write : V \longrightarrow T\,1$

Notions of computation determine monads

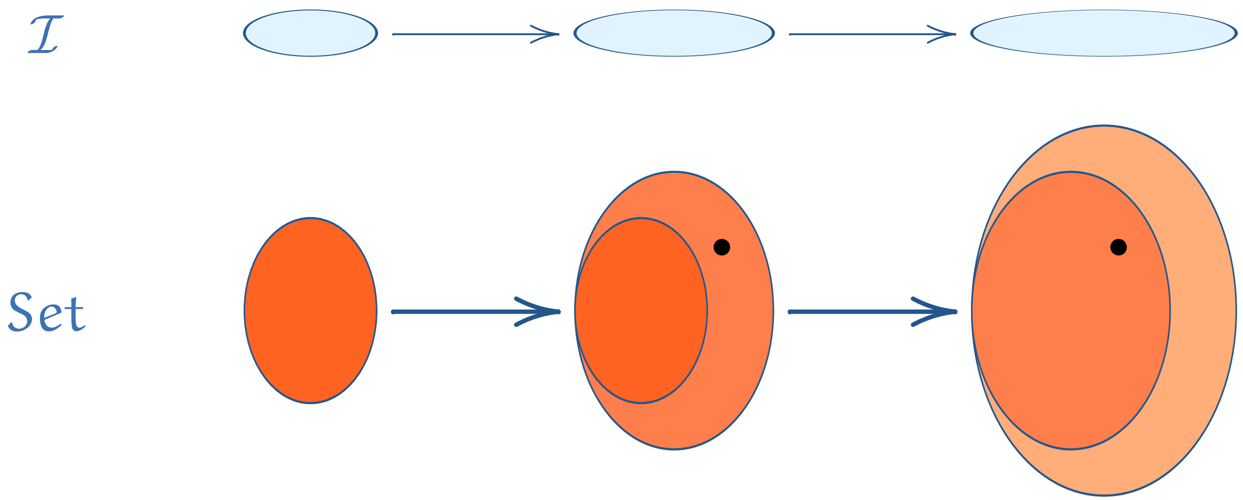
Operations + Equations $\longrightarrow$ Free-algebra models of computational features

$\longrightarrow$ Monads + generic effects

- Characterise known computational monads *and* effects.
- Simple and flexible combination of theories.
- Enriched models and arities: countably infinite, posets, $\omega\mathrm{Cpo}$.

The functor category $\mathcal{S}et^{\mathcal{I}}$

To account for names, we work with structures that vary according to the names available.



An object $\mathrm{B} \in \mathcal{S}et^{\mathcal{I}}$ is a varying set: it specifies for any finite set of names s the set $\mathrm{B}(s)$ of values using names from s, together with information about how these values change with renaming.

Structure within $\mathcal{S}et^{\mathcal{I}}$

We use $\mathcal{S}et^{\mathcal{I}}$ both as the arena for building name-aware algebras and monads, and as the source of arities for operations.

Relevant structure includes:

- Pairs $A \times B$ and function space $A \rightarrow B$;
- Separated pairs $A \otimes B$ and fresh function space $A \multimap B$;
- The object of names N;
- The shift endofunctor $\delta A = A(\_ + 1)$, with $\delta A = N \multimap A$.

In particular, the object N serves as a varying arity.

Theory of π: operations

Nondeterminism

$\mathit{nil} : 1 \longrightarrow A$	inactive process	0
$\mathit{choice} : A^2 \longrightarrow A$	process sum	$P + Q$

I/O

$\mathit{out} : A \longrightarrow A^{N \times N}$	output prefix	$\bar{x}y.P$
$\mathit{in} : A^N \longrightarrow A^N$	input prefix	$x(y).P$
$\mathit{tau} : A \longrightarrow A$	silent prefix	$\tau.P$

Dynamic name creation

$\mathit{new} : \delta A \longrightarrow A$	restriction	$\nu x.P$

Theory of π: interlude

Each operation induces a corresponding effect:

$$\begin{aligned} \mathsf{send} &: N \times N \longrightarrow T\,1 & \mathsf{deadlock} &: 1 \longrightarrow T\,0 \\ \mathsf{receive} &: N \longrightarrow T\,N & \mathsf{arb} &: 1 \longrightarrow T2 \\ \mathsf{skip} &: 1 \longrightarrow T\,1 & \mathsf{fresh} &: 1 \longrightarrow T\,N \end{aligned}$$

Other possible operations:

- par is not algebraic (because $(P \mid Q); R \neq (P; R) \mid (Q; R)$)
- $\mathsf{eq}, \mathsf{neq} : A \longrightarrow A^{N \times N}$ definable from $N \times N \cong N \otimes N + N$
- $\mathsf{bout} : \delta A \longrightarrow A^{N}$ can be defined from new and out

Theory of π: operations

Nondeterminism

$nil : 1 \longrightarrow A$	inactive process	0
$choice : A^2 \longrightarrow A$	process sum	$P + Q$

I/O

$out : A \longrightarrow A^{N \times N}$	output prefix	$\bar{x}y.P$
$in : A^N \longrightarrow A^N$	input prefix	$x(y).P$
$tau : A \longrightarrow A$	silent prefix	$\tau.P$

Dynamic name creation

$new : \delta A \longrightarrow A$	restriction	$\nu x.P$

Theory of π: component equations

Nondeterminism

$choice$ is associative, commutative and idempotent, with identity nil.

I/O

None.

Dynamic name creation

$$new(x.p) = p$$

$$new(x.new(y.p)) = new(y.new(x.p))$$

Theory of π: combining equations

Commuting

$$new(x.choice(p, q)) = choice(new(x.p), new(x.q))$$

$$new(z.out_{x,y}(p)) = out_{x,y}(new(z.p)) \qquad z \notin \{x, y\}$$

$$new(z.in_x(p_y)) = in_x(new(z.p_y)) \qquad z \notin \{x, y\}$$

$$new(z.tau(p)) = tau(new(z.p))$$

Interaction

$$new(x.out_{x,y}(p)) = nil$$

$$new(x.in_x(p_y)) = nil$$

Models of the theory of π

The category $\mathcal{PI}(\mathcal{S}et^{\mathcal{I}})$ of π-algebras has objects of the form $(A \in \mathcal{S}et^{\mathcal{I}}; in, out, \ldots, new)$ satisfying the equations given.

In any π-algebra A, each finite π-calculus process P has interpretation $[\![P]\!]_A$ defined by induction over the structure of P, using the operations of the theory (and the expansion law for parallel composition).

Thm: Every such π-algebra interpretation respects strong late bisimulation congruence:

$$P \approx Q \quad \Longrightarrow \quad [\![P]\!]_A = [\![Q]\!]_A \,.$$

Of course, this doesn't yet give us any actual π-algebras to work with.

Models of the theory of π

The category of π-algebras has a forgetful functor to $\mathit{Set}^{\mathcal{I}}$, taking each algebra to its underlying (varying) set:

$$\mathcal{PI}(\mathit{Set}^{\mathcal{I}}) \xrightarrow{\;U\;} \mathit{Set}^{\mathcal{I}} \qquad \text{forgetful}$$

Naturally, we now look for a free functor left adjoint to U, and its accompanying monad.

As it happens, using both closed structures at the same time means that general results engaged earlier don't immediately apply :-(

Free models for π

Each component theory has a standard monad:

Nondeterminism $\mathcal{P}_{fin}(X)$

I/O $\mu Y.(X + N \times N \times Y + N \times Y^N + Y)$

Name creation $\mathrm{Dyn}(X) = \int^k X(\_ + k)$

Weaving these together as monad transformers gives

$$\mu Y.\mathcal{P}_{fin}(\mathrm{Dyn}(X + N \times N \times Y + N \times Y^N + Y))\ldots$$

Free models for π

Each component theory has a standard monad:

Nondeterminism	$\mathcal{P}_{\mathrm{fin}}(X)$
I/O	$\mu Y.(X + N \times N \times Y + N \times Y^N + Y)$
Name creation	$\mathrm{Dyn}(X) = \int^k X(\_ + k)$

Weaving these together as monad transformers gives

$$\mu Y.\mathcal{P}_{\mathrm{fin}}(\mathrm{Dyn}(X + N \times N \times Y + N \times Y^N + Y))$$

...but the algebras for this do not satisfy the interaction equations between new and in/out.

Free models for π

Each component theory has a standard monad:

Nondeterminism $\mathcal{P}_{fin}(X)$

I/O $\mu Y.(X + N \times N \times Y + N \times Y^N + Y)$

Name creation $Dyn(X) = \int^k X(\_ + k)$

The correct monad for the combined theory is

$$Pi(X) = \mu Y.\mathcal{P}_{fin}(Dyn(X) + N \times N \times Y + N \times \delta Y + N \times Y^N + Y)$$

which adds bound output but otherwise does little with name creation.

Results

Thm: There is an adjunction making the category of π-algebras monadic over $\mathit{Set}^{\mathcal{I}}$.

$$\mathcal{PI}(\mathrm{Set}^{\mathcal{I}})$$

free $\quad \mathrm{Pi} \uparrow \;\dashv\; \downarrow \mathrm{U} \quad$ forgetful

$$\mathrm{Set}^{\mathcal{I}}$$

The composition $\mathrm{T}_\pi = (\mathrm{U} \circ \mathrm{Pi})$ is a computational monad for concurrent name-passing programs, with effects send, $\mathit{receive}$, arb, $\mathit{deadlock}$, skip and fresh.

Results

We have the following:

- A category $\mathcal{PI}(\mathcal{S}et^{\mathcal{I}})$ of π-algebras, all sound models of π-calculus bisimulation.

$$P \approx Q \quad\Longrightarrow\quad [\![P]\!]_A = [\![Q]\!]_A$$

- An explicit free-algebra construction $\mathsf{Pi} : \mathcal{S}et^{\mathcal{I}} \to \mathcal{PI}(\mathcal{S}et^{\mathcal{I}})$ such that all $\mathsf{Pi}(X)$ are fully-abstract models of π.

$$P \approx Q \quad\Longleftrightarrow\quad [\![P]\!]_{\mathsf{Pi}(X)} = [\![Q]\!]_{\mathsf{Pi}(X)}$$

- The inital free algebra $\mathsf{Pi}(0)$ is in fact the previously known fully-abstract model.

Parallel composition

Parallel composition of π-calculus processes is not algebraic, but we can nevertheless handle it in the following ways:

- All π-algebras can support $(P \mid Q)$ externally by expansion.
- All free π-algebras have an internally-defined map

$$\mathit{par}_{X,Y} : Pi(X) \times Pi(Y) \longrightarrow Pi(X \times Y) \ .$$

- Any multiplication $\mu : X \times X \to X$ then gives us

$$\mathit{par}_{\mu} : Pi(X) \times Pi(X) \longrightarrow Pi(X) \ .$$

- For $X = 0$, this is standard parallel composition; for $X = 1$ we get the same with an extra success process $\checkmark$.

Modal logic

Any theory gives rise to a modal logic over its algebras, with possibility and necessity modalities for every operation.

$$P \models \Diamond \mathrm{out}_{x,y}(\phi) \iff \exists Q.\ P \sim \bar{x}y.Q \ \wedge\ Q \models \phi$$

$$P \models \Box \mathrm{out}_{x,y}(\phi) \iff \forall Q.\ P \sim \bar{x}y.Q \ \Rightarrow\ Q \models \phi$$

$$P \models \Diamond \mathrm{choice}(\phi, \psi) \iff \exists Q, R.\ P \sim Q + R \ \wedge\ Q \models \phi \ \wedge\ R \models \psi$$

HML is definable:

$$\langle \bar{x}y \rangle \phi \ = \ \Diamond \mathrm{choice}(\Diamond \mathrm{out}_{x,y}(\phi), \mathrm{true})$$

We could also take other algebraic operations and define modalities. However, in no case is there a $(\phi \mid \psi)$ modality.

Review

Operations and equations with enriched arities can give algebraic models for features of computation.

Taking $\mathcal{S}et^{\mathcal{I}}$ for both arities and algebras, we can give a modular theory for the π-calculus:

$$\pi = (\text{Nondeterminism} + \text{I/O} + \text{Name creation}) \;/\; \mathit{new} \leftrightarrow \mathrm{i/o}$$

We have an explicit formulation of free algebras for this theory; all of these are fully abstract for bisimulation congruence.

The induced computational monad is almost, but not quite, the combination of its three components.

What next?

- Use $\mathcal{C}\mathrm{po}^{\mathcal{I}}$ for the full π-calculus. (OK, FM-$\mathcal{C}\mathrm{po}$)
- Partial order arities for testing equivalences. [Hennessy]
- Modify equations for early/open/weak bisimulation.
- Try $\mathrm{Pi}(X)$ for applied π.
- Investigate algebraic par. (with effect $\mathrm{fork} : 1 \longrightarrow T2$?)

- Build a proper theory of arities over two closed structures.

OR

- Exhibit $\mathcal{S}\mathrm{et}^{\mathcal{I}}$ as the category of algebras for a theory of equality testing in $\mathcal{S}\mathrm{et}^{\mathcal{F}}$, and then redo everything in the single Cartesian closed structure of $\mathcal{S}\mathrm{et}^{\mathcal{F}}$.

Constructions in $\mathcal{S}et^{\mathcal{I}}$

Cartesian closed

$$(A \times B)(k) = A(k) \times B(k)$$

$$B^A(k) = [A(k + \_), B(k + \_)]$$

Monoidal closed

$$(A \otimes B)(k) = \int^{k' + k'' \hookrightarrow k} A(k') \times B(k'')$$

$$(A \multimap B)(k) = [A(\_), B(k + \_)]$$

More constructions in $\mathcal{S}et^{\mathcal{I}}$

Object of names, shift operator

$$N(k) = k$$
$$\delta A(k) = A(k+1)$$

Connections

$$A \otimes B \longrightarrow A \times B \qquad \delta A \cong N \multimap A$$
$$(A \to B) \longrightarrow (A \multimap B) \qquad \delta N \cong N + 1$$

When A and B are pullback-preserving,
these are injective and surjective respectively.