Revisiting: Algebraic laws for nondeterminism and concurrency

Matthew Hennessy

Milner-Symposium, Edinburgh April 2012

TRINITY COLLEGE DUBLIN
COLÁISTE NA TRÍONÓIDE, BAILE ÁTHA CLIATH

sfi
science foundation ireland
fondúireacht eolaíochta éireann

History of a paper

Algebraic laws for nondeterminism and concurrency, JACM 1985
Matthew Hennessy and Robin Milner

- Research in late 1979 33 years ago
- Results presented at ICALP 1980 32 years ago
  (On Observing Nondeterminism and Concurrency)
- Rejected for publication 1982
- Rejected for publication 1983
- Published in JACM 1985

Edinburgh 1979 **33 years ago**

- No Labelled Transition Systems
- No CCS No CSP No ACP No . . .
- No street lightening
- What happened to the sun ?
- Lots of mushrooms
- No Bisimulations
- When does the summer arrive?
- Walks on Arthurs seat
- Lots of parking near George Square
- . . . . . .
- . . . . . .

Edinburgh 1979: Lots of denotational semantics

$D \cong [D \to D]$	functions	Scott, 1969
$P \cong V \to (V \times P)$	transformers	Milner 1971
$R \cong \mathcal{P}(S_\perp + (\mathcal{P}(S_\perp) \otimes R_\perp))^S$	resumptions	Plotkin 1976
$P_L \cong \mathcal{P}(\sum_{\beta \in L}(U_\beta \times (V_\beta \to P_L)))$	processes	Milne&Milner 1979

Edinburgh 1979: Lots of algebraic semantics

The Auld Alliance

- Jean-Marie Cadiou (1972): Recursive Definitions of Partial Functions and their Computations
- Jean Vuillemin (1973): Proof Techniques for Recursive Programs
- Bruno Courcelle, Maurice Nivat (1978): The Algebraic Semantics of Recursive Programme Schemes
- Irene Guessarian (1981): Algebraic Semantics

Edinburgh 1979: Lots of algebraic semantics

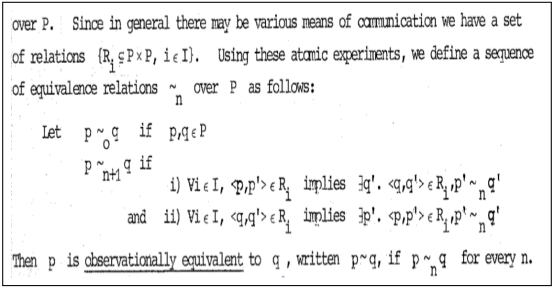
The Auld Alliance

- Jean-Marie Cadiou (1972): Recursive Definitions of Partial Functions and their Computations
- Jean Vuillemin (1973): Proof Techniques for Recursive Programs
- Bruno Courcelle, Maurice Nivat (1978): The Algebraic Semantics of Recursive Programme Schemes
- Irene Guessarian (1981): Algebraic Semantics

- Magmas: ordered sets with operators
- Ideal completions: adding limit points
- Initial algebra semantics

A behavioural equivalence

ICALP 1980:



over P. Since in general there may be various means of communication we have a set of relations $\{R_i \subseteq P \times P, i \in I\}$. Using these atomic experiments, we define a sequence of equivalence relations $\sim_n$ over P as follows:

Let $p \sim_0 q$ if $p,q \in P$

$p \sim_{n+1} q$ if

i) $\forall i \in I, \langle p,p'\rangle \in R_i$ implies $\exists q'. \langle q,q'\rangle \in R_i, p' \sim_n q'$

and ii) $\forall i \in I, \langle q,q'\rangle \in R_i$ implies $\exists p'. \langle p,p'\rangle \in R_i, p' \sim_n q'$

Then p is observationally equivalent to q, written $p \sim q$, if $p \sim_n q$ for every n.

Observatonal equivalence

1979

- Reduction semantics: $P \longrightarrow Q$ well-known

Observatonal equivalence

1979

- Reduction semantics: $P \longrightarrow Q$ well-known
- Observational semantics: $P \xrightarrow{\mu} Q$ new to me

Observatonal equivalence

1979

- Reduction semantics: $P \longrightarrow Q$ well-known
- Observational semantics: $P \xrightarrow{\mu} Q$ new to me

Observing processes:

- $p \sim_o q$ for all p, q zero observations
- $p \sim_{n+1} q$ if for every μ $(n+1)$ observations

  (i) $p \xrightarrow{\mu} p'$ implies $q \xrightarrow{\mu} q'$ such that $p' \sim_n q'$

  (ii) $q \xrightarrow{\mu} q'$ implies $p \xrightarrow{\mu} p'$ such that $p' \sim_n q'$

Transfer properties

Observatonal equivalence

1979

- Reduction semantics: $P \longrightarrow Q$ well-known
- Observational semantics: $P \xrightarrow{\mu} Q$ new to me

Observing processes:

- $p \sim_o q$ for all p, q zero observations
- $p \sim_{n+1} q$ if for every μ $(n+1)$ observations

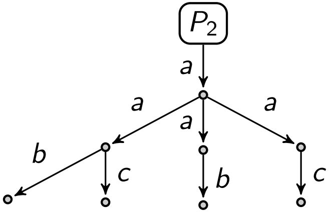
(i) $p \xrightarrow{\mu} p'$ implies $q \xrightarrow{\mu} q'$ such that $p' \sim_n q'$
(ii) $q \xrightarrow{\mu} q'$ implies $p \xrightarrow{\mu} p'$ such that $p' \sim_n q'$

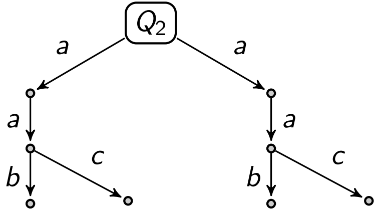
Transfer properties

Observational equivalence:

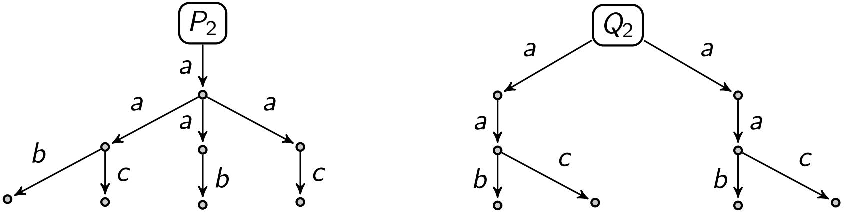
$$p \sim q \text{ if } \qquad p \; \boxed{\left(\bigcap_{n \geq 0} \sim_n\right)} \; q$$

Observing processes



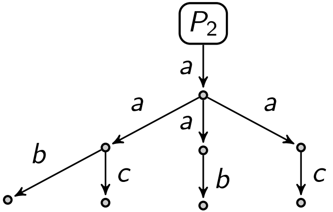


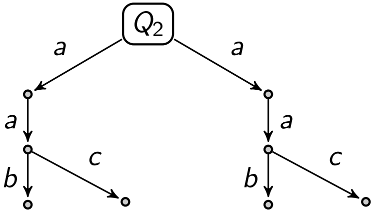
Observing processes



$P_2 \sim_o Q_2$ $P_2 \sim_1 Q_2$ $P_2 \sim_2 Q_2$ $P_2 \not\sim_3 Q_2$

Observing processes





$P_2 \sim_o Q_2$ $P_2 \sim_1 Q_2$ $P_2 \sim_2 Q_2$ $P_2 \not\sim_3 Q_2$

Life could get much more complicated:

$P_n \sim_n Q_n$ $P_n \not\sim_{(n+1)} Q_n$

Observational equivalence: Where from?

A Denotational Model

Milne&Milner 1979

$$P_L \;\cong\; \mathcal{P}(\sum_{\beta\in L}(U_\beta \times (V_\beta \to P_L)))$$

- L: set of *ports*
- U_β: output values on port β
- V_β: input values on port β

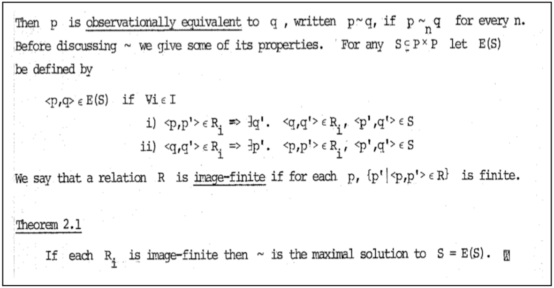
A simplification $U_\beta = V_\beta = 1$:

$$P_L \quad \cong \quad \mathcal{P}(\sum_{\mu\in L} P_L)$$

How would you compare two elements p, q from P_L?

Observational equivalence: a theorem

ICALP 1980:



Then p is observationally equivalent to q, written p ~ q, if $p \sim_n q$ for every n. Before discussing ~ we give some of its properties. For any $S \subseteq P \times P$ let $E(S)$ be defined by

$\langle p,q\rangle \in E(S)$ if $\forall i \in I$

i) $\langle p,p'\rangle \in R_i \Rightarrow \exists q'.\ \langle q,q'\rangle \in R_i,\ \langle p',q'\rangle \in S$

ii) $\langle q,q'\rangle \in R_i \Rightarrow \exists p'.\ \langle p,p'\rangle \in R_i,\ \langle p',q'\rangle \in S$

We say that a relation R is image-finite if for each p, $\{p' \mid \langle p,p'\rangle \in R\}$ is finite.

Theorem 2.1

If each R_i is image-finite then ~ is the maximal solution to $S = E(S)$. ☒

First research experiment

Process language:

finite non-deterministic machines

$$p \in W_{\Sigma_1} ::= \mathbf{0} \mid p + p \mid \mu.p$$

First research experiment

Process language:

finite non-deterministic machines

$$p \in W_{\Sigma_1} ::= \mathbf{0} \mid p + p \mid \mu . p$$

Result:

- $\boxed{\bigcap_{n \geq 0}(\sim_n)}$ is a Σ_1- congruence
- $p \; \boxed{\bigcap_{n \geq 0}(\sim_n)} \; q \qquad$ iff $p =_A q$

Axioms (A): $x + (y + z) = (x + y) + z \qquad x + y = y + x$

$x + x = x \qquad x + \mathbf{0} = x$

First research experiment

Process language:

finite non-deterministic machines

$$p \in W_{\Sigma_1} ::= \mathbf{0} \mid p + p \mid \mu.p$$

Result:

- $\boxed{\bigcap_{n\geq 0}(\sim_n)}$ is a Σ_1- congruence
- $p \ \boxed{\bigcap_{n\geq 0}(\sim_n)} \ q \qquad \text{iff } p =_A q$

Axioms (A): $x + (y + z) = (x + y) + z$ $\quad x + y = y + x$

$x + x = x$ $\quad x + \mathbf{0} = x$

Denotational semantics:

$$p \ \boxed{\bigcap_{n\geq 0}(\sim_n)} \ q \qquad \text{iff } [\![p]\!]_{(W_{\Sigma_1}\backslash A)} = [\![q]\!]_{(W_{\Sigma_1}\backslash A)}$$

$(W_{\Sigma_1}\backslash A)$: Initial algebra over W_{Σ_1} generated by axioms A

Robin had a lot of background

- 1973: Processes: A Mathematical model . . .
- 1978: Algebras for Communicating Systems
- 1978: Synthesis of Communicating Behaviour
- 1978: Flowgraphs and Flow Algebras
- 1979: An Algebraic Theory for Synchronisation
- 1979: Concurrent Processes and Their Syntax

Robin had a lot of background

- 1973: Processes: A Mathematical model . . .
- 1978: Algebras for Communicating Systems
- 1978: Synthesis of Communicating Behaviour
- 1978: Flowgraphs and Flow Algebras
- 1979: An Algebraic Theory for Synchronisation
- 1979: Concurrent Processes and Their Syntax

Combinators and their Laws proposed:

Robin had a lot of background

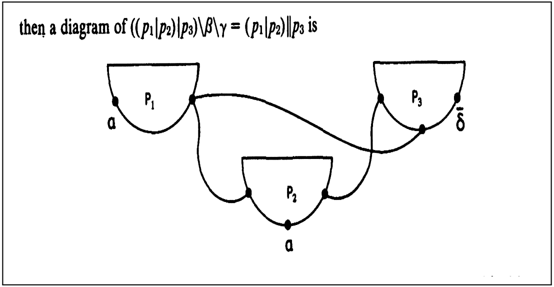
- 1973: Processes: A Mathematical model . . .
- 1978: Algebras for Communicating Systems
- 1978: Synthesis of Communicating Behaviour
- 1978: Flowgraphs and Flow Algebras
- 1979: An Algebraic Theory for Synchronisation
- 1979: Concurrent Processes and Their Syntax

Combinators and their Laws proposed:

- Flowgraphs and flow algebras for static structure
- Synchronisation trees for dynamics

Justifying equations

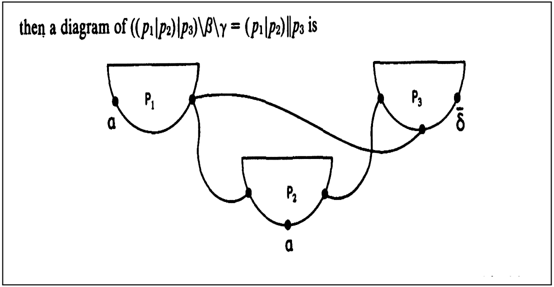
Flowgraphs:



then a diagram of $((p_1|p_2)|p_3)\backslash\beta\backslash\gamma = (p_1|p_2)\|p_3$ is

Justifying equations

Flowgraphs:



then a diagram of $((p_1|p_2)|p_3)\backslash\beta\backslash\gamma = (p_1|p_2)\|p_3$ is

Synchronisation trees:

Let $p = \sum_i \lambda_i.p_i,\ \ q = \sum_j \mu_j.q_j$. Then

$$p|q = \sum_i \lambda_i.(p_i|q) + \sum_j \mu_j.(p|q_j) + \sum_{\mu_j=\lambda_i} \tau.(p_i|q_j)$$

Theorems for free

$\Sigma_2 = \Sigma_1$ plus

- Parallelism: $|$
- Restriction: $\backslash\lambda$
- Renaming: $[S]$ S a function over names

Result:

- $\boxed{(\bigcap_{n\geq 0} \sim_n)}$ is a Σ_2- congruence
- $p \ \boxed{(\bigcap_{n\geq 0} \sim_n)} \ q \qquad \text{iff } p =_{A2} q$

Theorems for free

$\Sigma_2 = \Sigma_1$ plus

- Parallelism: $|$
- Restriction: $\backslash\lambda$
- Renaming: $[S]$ $\quad$ S a function over names

Result:

- $\boxed{(\bigcap_{n\geq 0} \sim_n)}$ is a Σ_2- congruence
- p $\boxed{(\bigcap_{n\geq 0} \sim_n)}$ q $\quad$ iff $p =_{A2} q$

A2 $=$ A1 $+$ existing axioms for $|, \ \backslash\lambda, \ [S]$

Weak case: abstracting from internal activity τ

- Weak observational semantics:

$$P \overset{\mu}{\Longrightarrow} Q \text{ meaning } P \xrightarrow{\tau}{}^{*} \xrightarrow{\mu} \xrightarrow{\tau}{}^{*} Q$$

External observations:

- $p \approx_o q$ for all p, q — zero observations
- $p \approx_{n+1} q$ if for every $\mu \in \mathsf{Act}_\tau$ — $(n+1)$ observations

(i) $p \overset{\mu}{\Longrightarrow} p'$ implies $q \overset{\mu}{\Longrightarrow} q'$ such that $p' \approx_n q'$

(ii) $q \overset{\mu}{\Longrightarrow} q'$ implies $p \overset{\mu}{\Longrightarrow} p'$ such that $p' \approx_n q'$

Weak transfer properties
look: no hats

Weak case: abstracting from internal activity τ

- Weak observational semantics:

$$P \overset{\mu}{\Longrightarrow} Q \text{ meaning } P \xrightarrow{\tau}{}^{*} \xrightarrow{\mu} \xrightarrow{\tau}{}^{*} Q$$

External observations:

- $p \approx_o q$ for all p, q — zero observations
- $p \approx_{n+1} q$ if for every $\mu \in \mathsf{Act}_\tau$ — $(n+1)$ observations

(i) $p \overset{\mu}{\Longrightarrow} p'$ implies $q \overset{\mu}{\Longrightarrow} q'$ such that $p' \approx_n q'$

(ii) $q \overset{\mu}{\Longrightarrow} q'$ implies $p \overset{\mu}{\Longrightarrow} p'$ such that $p' \approx_n q'$

Weak transfer properties
look: no hats

Weak observational equivalence:

$$p \approx q \text{ if } p \ \boxed{(\bigcap_{n\geq 0} \approx_n)} \ q$$

Equational characterisation

- Problem: $\boxed{(\bigcap_{n\geq 0} \approx_n)}$ is NOT preserved by operators $+$ or $|$

Equational characterisation

- Problem: $\boxed{(\bigcap_{n\geq 0} \approx_n)}$ is NOT preserved by operators $+$ or $|$
- Result: In Σ_1, $p \boxed{(\bigcap_{n\geq 0} \approx_n)_c} q$ iff $p =_{WA1} q$

Axioms WA1: add to A1 the τ-axioms:

$$x + \tau.x = \tau.x$$

$$\cancel{\mu.(x + \tau.y) = \mu.(x + y) + \mu.y} \quad \mu.\tau.y = \mu.y$$

$$\mu.(x + \tau.y) = \mu.(x + \tau.y) + \mu.y$$

Equational characterisation

- Problem: $\boxed{(\bigcap_{n\geq 0} \approx_n)}$ is NOT preserved by operators $+$ or $|$
- Result: In Σ_1, $p \boxed{(\bigcap_{n\geq 0} \approx_n)_c} q$ iff $p =_{WA1} q$

Axioms WA1: add to A1 the τ-axioms:

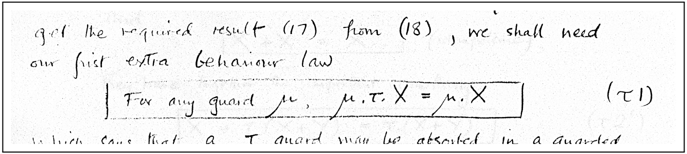
$$x + \tau.x = \tau.x$$

$$\cancel{\mu.(x + \tau.y) = \mu.(x + y) + \mu.y} \quad \mu.\tau.y = \mu.y$$

$$\mu.(x + \tau.y) = \mu.(x + \tau.y) + \mu.y$$

Where did these come from?

An exercise in Behaviour Algebra notes by Robin on modelling queues

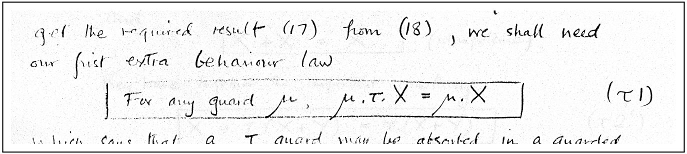


get the required result (17) from (18), we shall need our first extra behaviour law

$$\boxed{\text{For any guard } \mu, \quad \mu.\tau.X = \mu.X} \qquad (\tau 1)$$

which says that a τ guard may be absorbed in a guarded

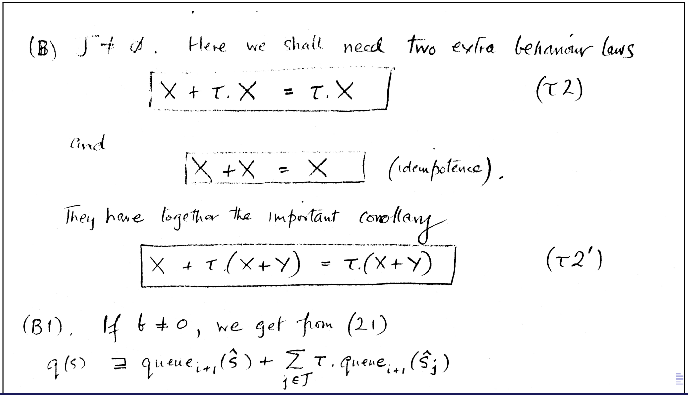
An exercise in Behaviour Algebra notes by Robin on modelling queues



get the required result (17) from (18), we shall need our first extra behaviour law

$$\boxed{\text{For any guard } \mu,\quad \mu.\tau.X = \mu.X} \qquad (\tau 1)$$

which says that a τ guard may be absorbed in a guarded



(B) $J \neq \emptyset$. Here we shall need two extra behaviour laws

$$\boxed{X + \tau.X = \tau.X} \qquad (\tau 2)$$

and

$$\boxed{X + X = X} \quad \text{(idempotence)}.$$

They have together the important corollary

$$\boxed{X + \tau.(X+Y) = \tau.(X+Y)} \qquad (\tau 2')$$

(B1). If $b \neq 0$, we get from (21)

$$q(s) \sqsupseteq queue_{i+1}(\hat{s}) + \sum_{j \in J} \tau . queue_{i+1}(\hat{s}_j)$$

Hennessy Milner Logic where did this come from?

Observational equivalence $p \; \boxed{(\bigcap_{n \geq 0} \sim_n)} \; q$

- *Inspired by* identity in domain $P_L \;\cong\; \mathcal{P}(\,\sum_{\mu \in L} P_L\,)$

Hennessy Milner Logic where did this come from?

Observational equivalence $p \; \boxed{(\bigcap_{n\geq 0} \sim_n)} \; q$

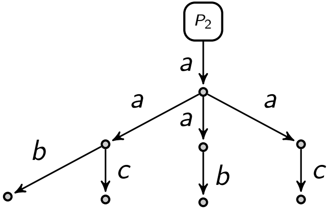
- *Inspired by* identity in domain $P_L \;\cong\; \mathcal{P}(\,\sum_{\mu\in L} P_L\,)$
- *Requires* independent justification

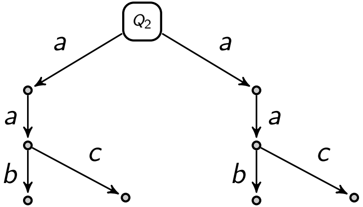
Hennessy Milner Logic where did this come from?

Observational equivalence $p \; \boxed{\left(\bigcap_{n \geq 0} \sim_n\right)} \; q$

- *Inspired by* identity in domain $P_L \;\cong\; \mathcal{P}(\sum_{\mu \in L} P_L\,)$
- *Requires* independent justification

Why are these behaviourally different:

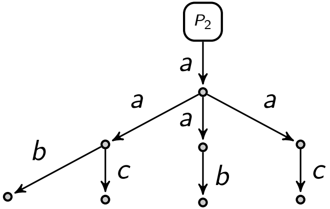


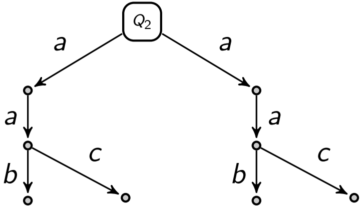


Discover difference using *interaction games*:

- can do action x
- can not do action x

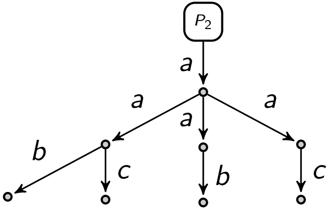
Discovering differences

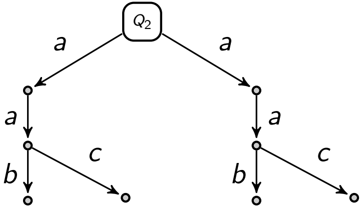




Q_2 can perform a so that
every time a is subsequently performed
both b and c can be performed

Discovering differences





Q_2 can perform *a* so that

every time *a* is subsequently performed

both *b* and *c* can be performed

$$Q_2 \models \langle a\rangle[a](\langle b\rangle\mathrm{tt} \wedge \langle c\rangle\mathrm{tt})$$

$$P_2 \not\models \ldots$$

Hennessy Milner Logic

$$A, B \in \mathcal{L} ::= \texttt{tt} \mid A \wedge B \mid \neg A \mid \langle \mu \rangle A$$

- $p \models \langle \mu \rangle A$ if $p \xrightarrow{\mu} p'$ such that $p' \vdash A$
- $p \models A \wedge B$ if ......

Result:

- $p \boxed{(\bigcap_{n \geq 0} \sim_n)} q$ iff $\mathcal{L}(p) = \mathcal{L}(q)$ requires image-finiteness
- $p \cancel{\boxed{(\bigcap_{n \geq 0} \sim_n)}} q$ iff $p \models A$ and $q \not\models A$, for some $A \in \mathcal{L}$.

A is an explanation of why p, q are different

Enter . . . **David Park** 1935 - 1990

Enter . . . **David Park** 1935 - 1990

Fixpoint induction:

1970 machine intelligence

If $F(H) \leq H$ then $\mathbf{min} X.F(X) \leq H$

requires monotonicity

Enter . . . **David Park** 1935 - 1990

Fixpoint induction:

1970 machine intelligence

If $F(H) \leq H$ then $\mathbf{min}X.F(X) \leq H$

requires monotonicity

Fair merge:

1979

$$\text{fairmerge} = \mathbf{max}X.\mathbf{min}\,Y.(\text{Fm}(\mathbf{min}Z.\text{Fm}(Z,X),Y)$$

$$\begin{array}{rcl} \text{where Fm}(X,Y) & = & \{(\epsilon,x,x)|x \in \Sigma^\infty\} \ \cup \ \{(x,\epsilon,x)|x \in \Sigma^\infty\} \\ & = & \{(ax,y,az)|a \in \Sigma,\ (x,y,z) \in X\} \\ & = & \{(x,ay,az)|a \in \Sigma,\ (x,y,z) \in Y\} \end{array}$$

Using Maximal Fixpoints

Icalp 1980: Hennessy & Milner

Extensive use in meta-theory of processes:

- *Theorem 2.1 If each R_i is image-finite then $\sim$ is the maximal solution to $S = E(S)$*
- ALNC, page 157: *Now let $\approx'$ be the maximal solution to the equation $S = E'(S)$*

Using Maximal Fixpoints

Icalp 1980: Hennessy & Milner

Extensive use in meta-theory of processes:

- *Theorem 2.1 If each R_i is image-finite then $\sim$ is the maximal solution to $S = E(S)$*
- ALNC, page 157: *Now let $\approx'$ be the maximal solution to the equation $S = E'(S)$*

David Park:

Use maximal fixpoints in object-theory of processes

Replace $\boxed{(\bigcap_{n\geq 0} \sim_n)}$ with a maximal fixpoint $\sim_{bis}$

Co-induction à la David Park

Transfer property:

For $R \subseteq P \times P$, define $\mathcal{B}(R) \subseteq P \times P$ by
$p\,\mathcal{B}(R)\,q$ whenever

(i) $p \xrightarrow{\mu} p'$ implies $q \xrightarrow{\mu} q'$ such that $p\,R\,q$

(ii) $q \xrightarrow{\mu} q'$ implies $p \xrightarrow{\mu} p'$ such that $p\,R\,q$

Bisimulations:

- $R \subseteq P \times P$ is a bisimulation if $\mathcal{B}(R) \subseteq R$
- $p \sim_{bis} q$ if $p\,R\,q$ for some bisimulation R

Elegant proof for establishing $p \sim_{bis} q$

Co-induction à la David Park

Robin Milner: A Calculus of Communicating Systems, LNCS **1980**

Co-induction à la David Park

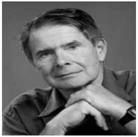
Robin Milner: A Calculus of Communicating Systems, LNCS **1980**

$\Downarrow$

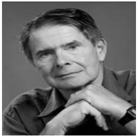
Robin Milner: Communication and Concurrency, Prentice-Hall, **1984**

- elegant theory
- lots of worked examples
- detailed proofs

Jim Morris and his style of equivalences



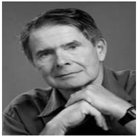
Jim Morris and his style of equivalences



James H Morris, PhD Thesis: *Lambda Calculus Models of Programming Languages*, 1968.

- Proposed Theorem:
  In Lambda, if $FA \sqsubseteq A$ then $\mathbf{Y}F \sqsubseteq A$

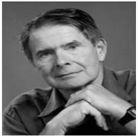
Jim Morris and his style of equivalences



James H Morris, PhD Thesis: *Lambda Calculus Models of Programming Languages*, 1968.

- Proposed Theorem:
  In Lambda, if $FA \sqsubseteq A$ then $\mathbf{Y}F \sqsubseteq A$
- Question: What is $\sqsubseteq$?

Jim Morris and his style of equivalences



James H Morris, PhD Thesis: *Lambda Calculus Models of Programming Languages*, 1968.

- Proposed Theorem:
  In Lambda, if $FA \sqsubseteq A$ then $\mathbf{Y}F \sqsubseteq A$
- Question: What is $\sqsubseteq$?

Morris Preorder:

$A \sqsubseteq_{morris} B$ if for every context $C[\,]$

$C[A]$ *has a normal form* *implies* $C[B]$ *has a normal form*

Morris - style of equivalences

Ingredients:

- A reduction semantics: $P \;\rightarrow\; Q$
- Results: $P \Downarrow v$ barbs
- Language syntax for contexts $C[\,]$

Contextual equivalence:

$P \cong_{cxt} Q$ if for every context, for every barb,

$$C[P] \rightarrow^* P' \Downarrow v \quad \text{iff} \quad C[Q] \rightarrow^* Q' \Downarrow v$$

Morris - style of equivalences

Ingredients:

- A reduction semantics: $P \;\rightarrow\; Q$
- Results: $P \Downarrow v$ barbs
- Language syntax for contexts $C[\,]$

Contextual equivalence:

$P \approxeq_{cxt} Q$ if for every context, for every barb,

$$C[P] \rightarrow^{*} P' \Downarrow v \quad \text{iff} \quad C[Q] \rightarrow^{*} Q' \Downarrow v$$

Where are the quantifiers?

Justifying Bisimulation Equivalence

Barbed congruence:

Milner, Sangiorgi 1992

For image-finite CCS processes,

$$P \approx_{bism} Q \quad \text{iff} \quad P \cong_{barb} Q$$

Justifying Bisimulation Equivalence

Barbed congruence: Milner, Sangiorgi 1992

For image-finite CCS processes,

$$P \approx_{bism} Q \quad \text{iff} \quad P \cong_{barb} Q$$

Reduction barbed congruence: Honda, Yoshida 1993

For arbitrary CCS processes,

$$P \approx_{bism} Q \quad \text{iff} \quad P \cong_{rbc} Q$$

Justifying Bisimulation Equivalence

Barbed congruence:

Milner, Sangiorgi 1992

For image-finite CCS processes,

$$P \approx_{bism} Q \quad \text{iff} \quad P \approxeq_{barb} Q$$

Reduction barbed congruence:

Honda, Yoshida 1993

For arbitrary CCS processes,

$$P \approx_{bism} Q \quad \text{iff} \quad P \approxeq_{rbc} Q$$

Both contextual equivalences are reduction closed:

- $P \rightarrow^* P'$ implies $Q \rightarrow^* Q'$ s.t. $P' \approxeq_{.} Q'$
- $Q \rightarrow^* Q'$ implies $P \rightarrow^* Q'$ s.t. $P' \approxeq_{.} Q'$

Bisimulations in the Modern World

Pick your favourite process language

Bisimulations in the Modern World

Pick your favourite process language

- Bisimulations do not provide a behavioural theory of processes *per se*
- Bisimulations provide a proof methodology for demonstrating processes to be equivalent
- HML provide a methodology for explaining why processes are not equivalent

Bisimulations in the Modern World

Pick your favourite process language

- Bisimulations do not provide a behavioural theory of processes *per se*
- Bisimulations provide a proof methodology for demonstrating processes to be equivalent
- HML provide a methodology for explaining why processes are not equivalent
- Bisimulations are very often sound w.r.t. the natural contextual equivalence $\cong_{cxt}$
- Bisimulations are sometimes complete w.r.t. the natural contextual equivalence $\cong_{cxt}$
- Formulating complete bisimulations very often sheds light process behaviour

Examples a very small sample

- Asynchronous Picalculus: Honda, Tokoro 1991, Amadio Castellani Sangiorgi 1998
- Mobile Ambients: Merro, Zappa Nardelli 1985
- Existential and recursive types in lambda-calculus: Sumii, Pierce 2007
- Higher-order processes: environmental bisimulations Sangiorgi, Kobayahsi, Sumii 2007
- Aspects in a functional language: open bisimulations Jagadeesan, Pitcher, Riely 2007
- Concurrent Probabilistic processes: Deng, Hennessy 2011

Examples a very small sample

- Asynchronous Picalculus: Honda, Tokoro 1991, Amadio Castellani Sangiorgi 1998
- Mobile Ambients: Merro, Zappa Nardelli 1985
- Existential and recursive types in lambda-calculus: Sumii, Pierce 2007
- Higher-order processes: environmental bisimulations Sangiorgi, Kobayahsi, Sumii 2007
- Aspects in a functional language: open bisimulations Jagadeesan, Pitcher, Riely 2007
- Concurrent Probabilistic processes: Deng, Hennessy 2011
- Bigraphs: Robin and co-workers
  - Bigraphs: all encompassing descriptive language
  - Recovery of LTS from reduction semantics
  - ensuring soundness of bisimulations