

Explicit Fusions

Lucian Wischik (Cambridge)

Bologna, October 2000

*joint work with Philippa Gardner
paper in proceedings of MFCS2000*

what is an “explicit fusion”?

```
void P (int &x, int &y)
{
  x = x+y;
  y = x-y;
  x = x-y;
}
```

```
fun P x y =
(
  x := !x + !y;
  y := !x - !y;
  x := !x - !y;
);
```

How does P behave?

How does P behave in a context where $x=y$?

We write:

$$\begin{aligned} & \langle x=y \rangle \mid P \\ \equiv & \langle x=y \rangle \mid P\{y/x\} \\ \equiv & \langle x=y \rangle \mid P\{x/y\} \end{aligned}$$

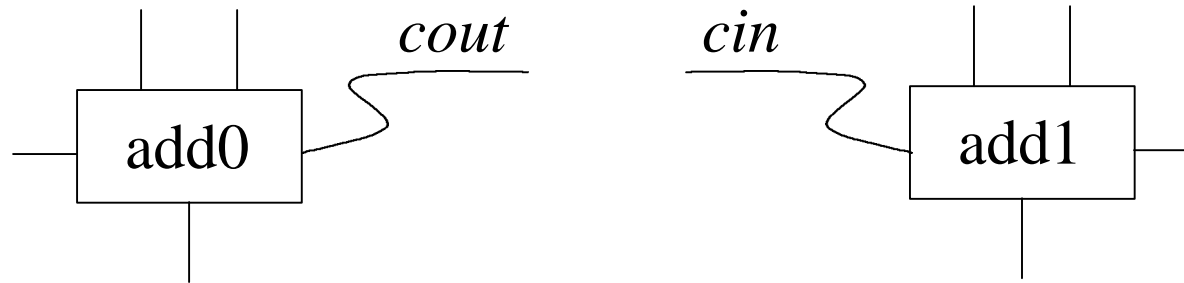
why explicit fusions?

- they occur in real world
 - help implementation
 - momentum of related work
 - simplify syntax
-
- can add explicit fusions to different calculi. For example:

pi-F calculus

- like pi but with explicit fusions
- LTS, bisimulation
- embeds pi, fusion calculi
- full abstraction for fusion calc.

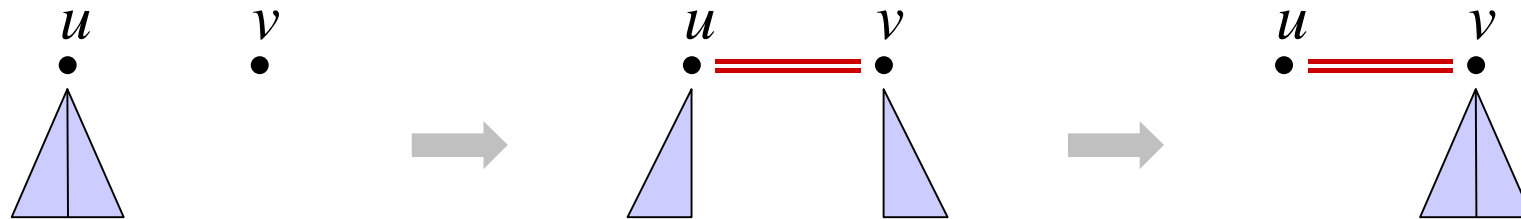
fusions in hardware



$$(\langle \text{cout} \rangle | A_0) @ (\langle \text{cin} \rangle | A_1) \equiv \langle \text{cout} = \text{cin} \rangle | A_0 | A_1$$

e.g. SHRM [G00]

fusions in internet



Fusion allows for *asynchronous* migration of code: migration can happen gradually, and is non-blocking.

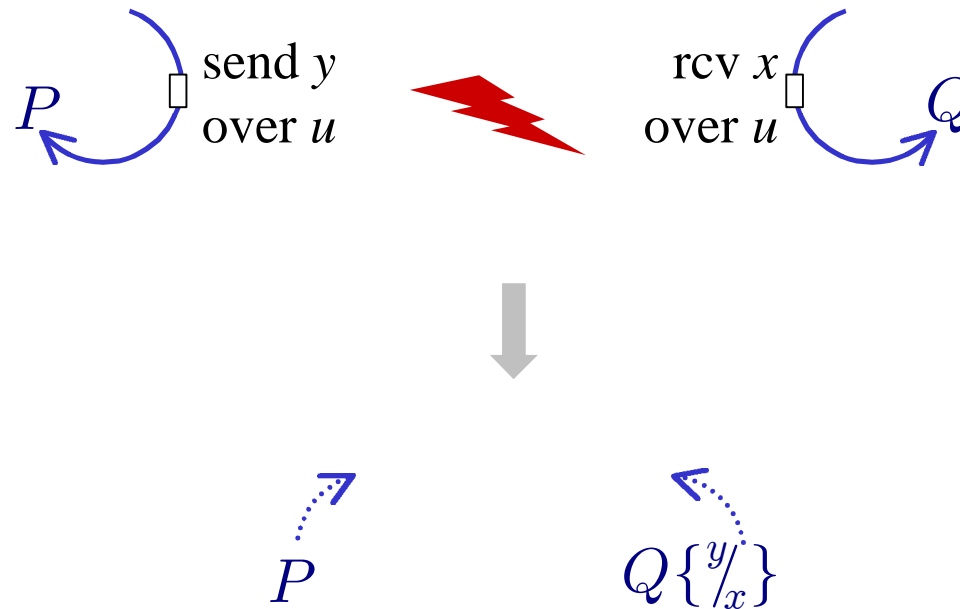
eg. ambients [FLS00], pi-calculus machine [in progress]

“Explicitly record and delay the effect of the fusion”
cf. *Explicit Substitutions* [ACCL 91]

pi-calculus

- communication between output & input

$$\bar{u}.<y>P \mid u.(x)Q \quad \searrow \quad P \mid Q\{y/x\}$$



- dynamic creation of names/channels
- mobile scope of names/channels $(\nu y)P \mid Q \equiv (\nu y)(P \mid Q)$

related work 1

pi. [MPW89]

$\bar{u}.<y>P$ output

$u.(x)Q$ input

$(\nu z)R$ name-hiding

$\bar{u}.<y>P \mid u.(x)Q \searrow P \mid Q\{y/x\}$

explicit fusions. [GW00]

fusion calculus. [VP98]

chi calculus. [FU97]

$\bar{u}.<y>P, \quad u.<x>Q, \quad (\nu z)R$

$\bar{u}.<y>P \mid u.<x>Q \searrow ???$

pi-I. [Sang96]

$\bar{u}.(y)P, \quad u.(x)Q, \quad (\nu z)R$

$\bar{u}.(y)P \mid u.(x)Q \searrow (\nu z)(P\{z/y\} \mid Q\{z/x\})$
 z fresh

related work 2

Categorical Framework

Graphs

Fusions

Explicit

Symmetric ACs [GW99]

Expl. Subs

Fusion Systems [G00]

π -F [GW00]

Equators [H95]

Permutations [H]

Process Graphs [Y94]

Term graph rewriting [GM]

pi-F calculus

- grammar
- reaction relation
- structural congruence
(e.g. alpha conversion)

labels, bisimulation

- use CCS labels \bar{u} u τ
- “open” bisimulation
- result: is a congruence

embedding pi, fusion

- how to encode bound inp.
- result: reaction preserved. Full abstraction for fusion but not pi

pi-F calculus

$P ::=$	nil	empty process
	$x.P$	input
	$\bar{x}.P$	output
	$P P$	parallel composition
	$(\nu x)P$	name-hiding
	$\langle x \rangle$	datum
	$\langle x=y \rangle$	explicit fusion

datums

- are like concretions
- not commutative
- polyadic, through parallel composition

$\langle x \rangle | \langle y \rangle | P \dots \langle xy \rangle P$

explicit fusion

- denotes an equivalence relation
- finite basis
- has substitutive effect

pi-F reaction

$$\bar{u}.P \mid u.Q \searrow P @ Q$$

effect of @ is to fuse datums

interface

$$\begin{aligned} \bar{u}.(\langle y \rangle \mid P) \mid u.(\nu x)(\langle x \rangle \mid Q) &\searrow (\langle y \rangle \mid P) @ (\nu x)(\langle x \rangle \mid Q) \\ &\equiv (\nu x)(\langle x=y \rangle \mid P \mid Q) \\ &\equiv P\{y/x\} \mid Q\{y/x\} \end{aligned}$$

encode bound input $u.(x)Q$

encode lazy input $(\nu x)(u.\langle x \rangle \mid Q)$

struct.cong. rules for fusions

$$\begin{aligned}
 & (\nu x)(\bar{x}.\mathbf{nil}) \\
 \equiv & (\nu x)(\nu y)(\langle x=y \rangle | \bar{x}.\mathbf{nil}) && \text{create fresh bound name } y \text{ as alias for } x \\
 \equiv & (\nu x)(\nu y)(\langle x=y \rangle | \bar{y}.\mathbf{nil}) && \text{substitute } y \text{ for } x \\
 \equiv & (\nu y)(\bar{y}.\mathbf{nil}) && \text{remove the now-unused bound name } x
 \end{aligned}$$

$$\begin{aligned}
 \langle x=y \rangle | \langle y=z \rangle & \equiv \langle x=y \rangle | \langle x=z \rangle && \text{transitivity} \\
 \langle x=y \rangle & \equiv \langle y=x \rangle && \text{symmetry} \\
 \langle x=x \rangle & \equiv \mathbf{nil} && \text{reflexivity}
 \end{aligned}$$



$$\begin{aligned}
 \langle x=y \rangle | \bar{z}.P & \equiv \langle x=y \rangle | \bar{z}.\langle x=y \rangle | P \\
 \langle x=y \rangle | \bar{x}.P & \equiv \langle x=y \rangle | \bar{y}.P && \text{and other small-step substitutions}
 \end{aligned}$$



$$(\nu x)(\langle x=y \rangle) \equiv \mathbf{nil} \quad \text{restriction delimits scope of fusion}$$

pi-F labels, bisimulation

$$\bar{u}.P \xrightarrow{\bar{u}} P \qquad u.P \xrightarrow{u} P \qquad \bar{u}.P | u.Q \xrightarrow{\tau} P @ Q$$

closed w.r.t. contexts and \equiv

Bisimulation is standard, augmented for fusion contexts:

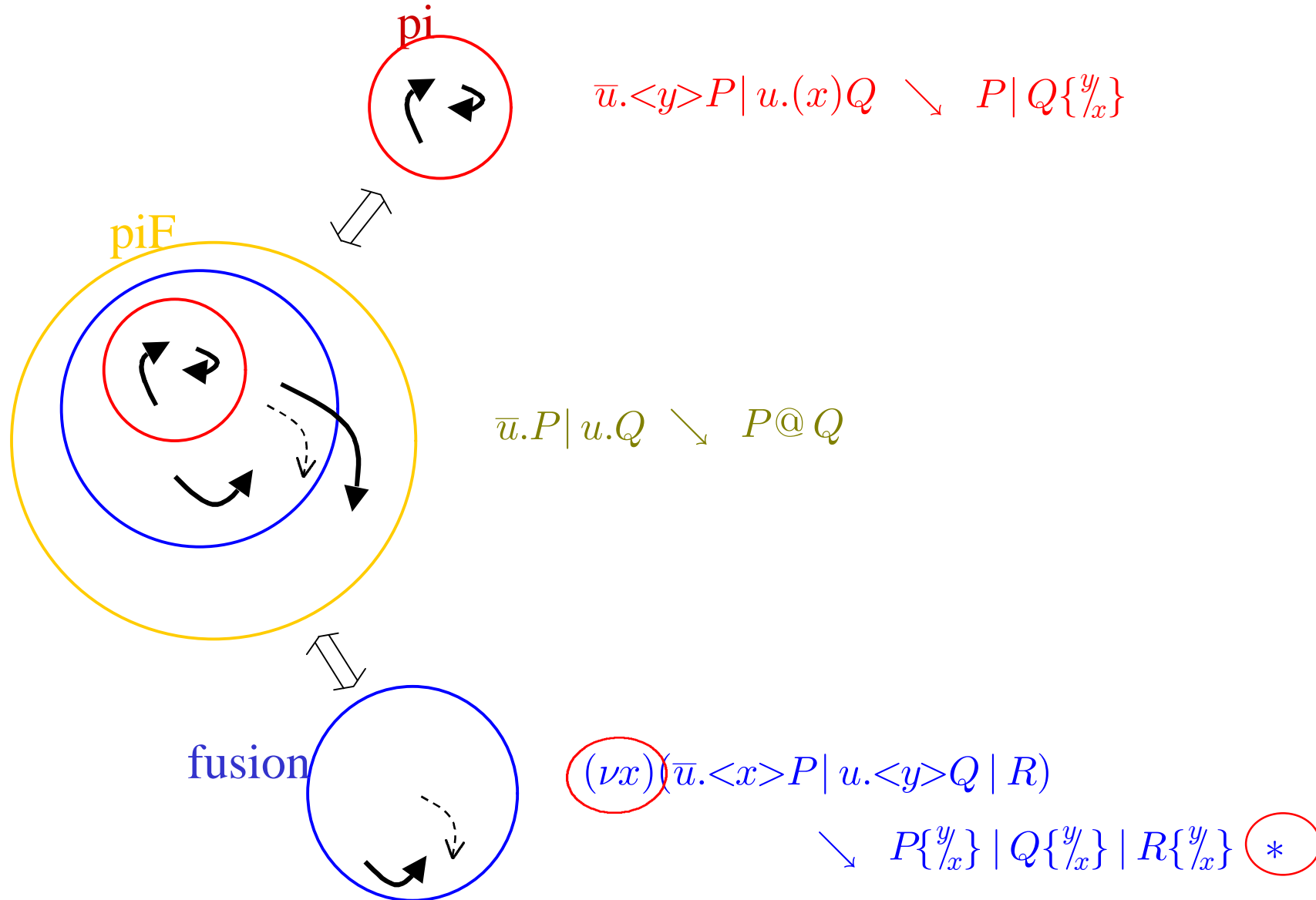
a relation S is an *open bisimulation* iff whenever $P S Q$ then

- P, Q have same **interface** (same datums, fusions, restrictions)
 - for all x, y , if $\langle x=y \rangle | P \xrightarrow{\alpha} P_1$ then $\langle x=y \rangle | Q \xrightarrow{\alpha} Q_1$ and $P_1 S Q_1$
 - and vice versa

To avoid the quantification, we add a *fusion transition*

$$\bar{u}.P | v.Q \xrightarrow{?u=v} P @ Q$$

embedding results 1



embedding results 1b

(1) Structural congruence for **pi** and **fusion embeddings**

$$P \equiv Q \quad \text{iff} \quad P^* \equiv Q^*$$

(2) Reaction relation for **pi embedding**

$$P \searrow Q \quad \text{iff} \quad P^* \searrow Q^*$$

(3) Reaction relation for **fusion embedding**. Need constraint *

$$P \searrow Q \quad \text{implies} \quad P^* \searrow Q^*$$

$$P^* \searrow Q \quad \text{implies} \quad \exists R, \vec{x}. (\nu \vec{x})P \searrow R \quad \text{and} \quad R^* \equiv (\nu \vec{x})Q$$

fusion calculus

$$\bar{u}.<y>P \quad u.<x>Q \quad (\nu x)R$$

$$\bar{u}x.P \xrightarrow{\bar{u}x} P \quad \text{standard output}$$

$$(\nu x)\bar{u}x.P \xrightarrow{(x)\bar{u}x} P \quad \text{standard bound output}$$

$$\bar{u}x.P \mid uy.Q \xrightarrow{!x=y} P \mid Q \quad \text{interaction results in a fusion}$$

$$\bar{u}x.P \mid uy.Q \mid R \xrightarrow{!x=y} P \mid Q \mid R \quad \text{fusion has potentially global effect}$$

$$(\nu x)(\bar{u}x.P \mid uy.Q \mid R) \xrightarrow{\mathbf{1}} P\{y/x\} \mid Q\{y/x\} \mid R\{y/x\} \quad \text{delimit extent of fusion}$$

embedding results 2

$$\begin{array}{lcl}
 \bar{u}x.P & \xrightarrow{\bar{u}x} & P & \dots\dots\dots & \bar{u}.(\langle x \rangle | P^*) & \xrightarrow{\bar{u}} & \langle x \rangle | P^* \\
 (\nu x)\bar{u}x.P & \xrightarrow{(x)\bar{u}x} & P & \dots\dots\dots & (\nu x)\bar{u}.(\langle x \rangle | P^*) & \xrightarrow{\bar{u}} & (\nu x)(\langle x \rangle | P^*) \\
 \bar{u}x.P | uy.Q & \xrightarrow{!x=y} & P | Q & \dots\dots\dots & \bar{u}.(\langle x \rangle | P^*) | u.(\langle y \rangle | Q^*) & \xrightarrow{\tau} & \langle x=y \rangle | P^* | Q^*
 \end{array}$$

embedding results 2b

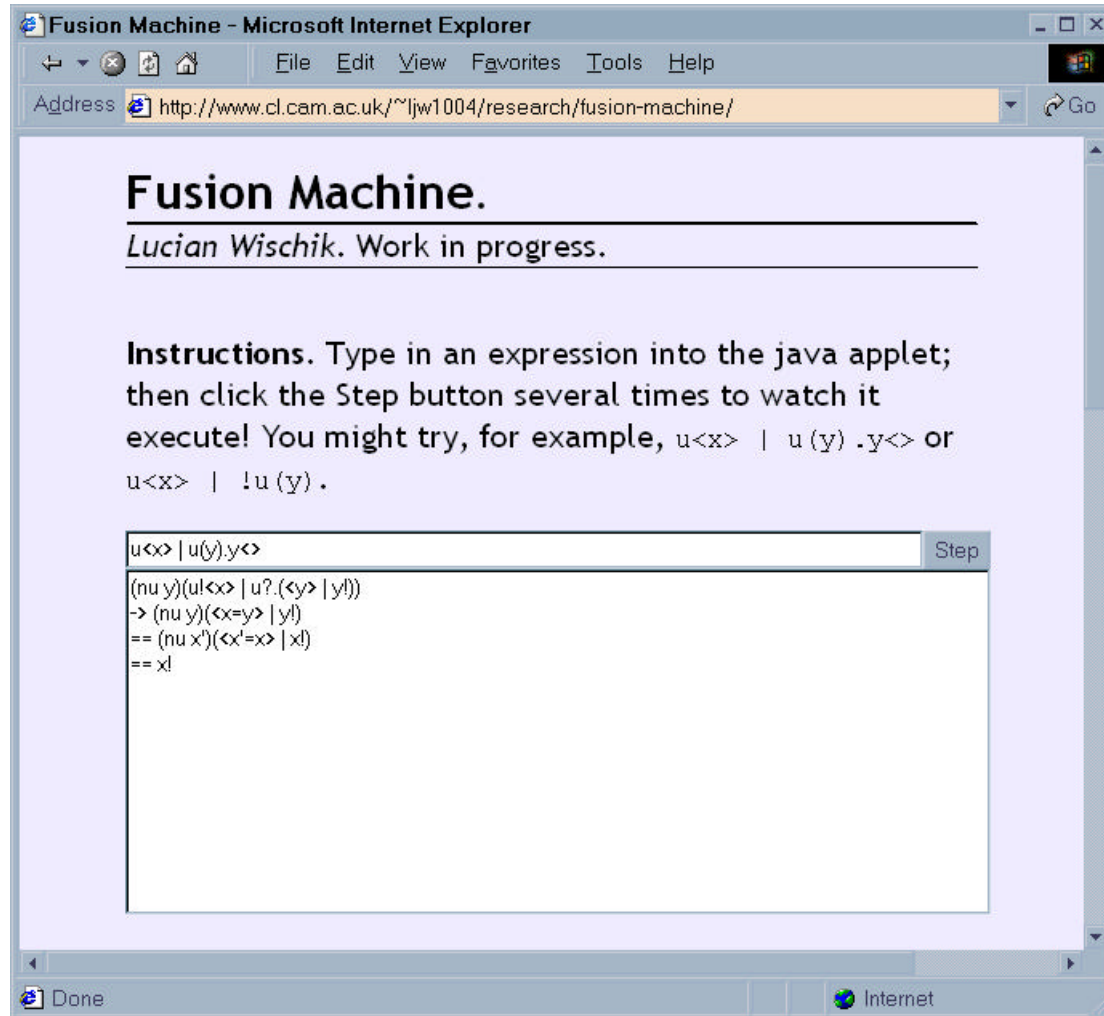
(1) fusion-calculus hyperequivalence coincides with pi-F bisimulation on its embedded image.

- proof: take hyperequivalence, close it under all interfaces, and the result is a pi-F bisimulation. And *vice versa*.

(2) pi-calculus open bisimulation is strictly weaker than pi-F bisimulation.

- “open” bisimulation [S96] : one that’s closed w.r.t. substitutions — or fusions
- proof [VP98]: in following expression, no pi-context can make $x=y$; but pi-F can: $(\nu x y) (\bar{u}. \langle x y \rangle \mid P)$

ongoing work



- (replication)
- implementation based on explicit fusions
- lambda calculus encoding, using datums to represent lambda variables
- use explicit fusions in other areas (“fusion systems”)