A categorical model of the Fusion Calculus

Marino Miculan
University of Udine

IoC
Tallinn, 11 February 2010

Adding the missing notion of state to behaviour functors obtained from nice labelled transition systems, in order to get a working categorical semantics

Marino Miculan
University of Udine

IoC
Tallinn, 11 February 2010

How to uglify your semantics – but only a bit, and for good reasons

Marino Miculan
University of Udine

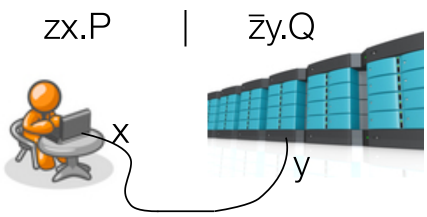
IoC
Tallinn, 11 February 2010

Handshakes, negotiations

- Distributed programming often involve *negotiations, handshakes…*
    - “Connect port 127.0.0.1:26451 to port 163.10.72.61:25”   (TCP)
    - “let us choose a cypher supported by both” (SSL, IPSec)
    - “we will use the following encoding”
- Basic concept: **communication via unification**
- Quite different from usual one-way communication (like CCS and π-calculus)
- In this talk: a categorical model for the Fusion calculus
- Methodology is more general

4

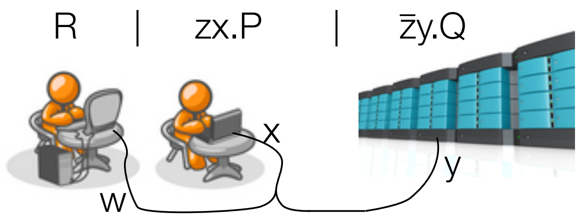
Communication via unification



- **Symmetric**: unification affects *both* agents (not only one side)
- P and Q keep using x and y for the same “wire” (i.e., substitution is not (always) performed)
- Action is **locally performed** by two agents

5

Communication via unification



- **Symmetric**: unification affects *both* agents (not only one side)
- P and Q keep using x and y for the same “wire” (i.e., substitution is not (always) performed)
- Action is **locally performed** by two agents, but has **global effect**

6

The Fusion Calculus (Parrow and Victor, 1998)

- **Syntax** (up-to alpha-equivalence):

x is bound

$$P, Q ::= \mathbf{0} \mid zx.P \mid \bar{z}x.P \mid P|Q \mid (x)P$$

- **Semantics**: labelled transitions of the form

fusion: "from now on, consider x and y equal"

$$\alpha ::= xy \mid \bar{x}y \mid x(z) \mid \bar{x}(z) \mid \mathbf{1} \mid x{=}y$$

- transitions **do not** keep track of equivalences between names: fusions caused by communications are *exposed* to the environment as labels; e.g.

$$Com\ \frac{P \xrightarrow{ux} P',\ Q \xrightarrow{\bar{u}y} Q'}{P|Q \xrightarrow{\{x=y\}} P'|Q'}$$

The Fusion Calculus – Labelled transition system

$$Pref\ \frac{-}{\alpha.P \xrightarrow{\alpha} P} \qquad Com\ \frac{P \xrightarrow{ux} P',\ Q \xrightarrow{\bar{u}y} Q'}{P|Q \xrightarrow{\{x=y\}} P'|Q'} \qquad Par\ \frac{P \xrightarrow{\alpha} P'}{P|Q \xrightarrow{\alpha} P'|Q}$$

$$Open\ \frac{P \xrightarrow{ux} P',\ u \notin \{x, \bar{x}\}}{(x)P \xrightarrow{u(x)} P'} \qquad Pass\ \frac{P \xrightarrow{\alpha} P',\ x \notin n(\alpha)}{(x)P \xrightarrow{\alpha} (x)P'} \qquad Scope\ \frac{P \xrightarrow{\{x=y\}} P'}{(x)P \xrightarrow{\mathbf{1}} P'\{y/x\}}$$

$$Congr\ \frac{P \equiv P'\ P' \xrightarrow{\alpha} Q'\ Q' \equiv Q}{P \xrightarrow{\alpha} Q} \qquad (P|Q)|R \equiv P|(Q|R) \quad P|Q \equiv Q|P \quad P|\mathbf{0} \equiv P$$

$$(x)\mathbf{0} \equiv \mathbf{0} \quad (x)(y)P \equiv (y)(x)P \qquad P|(x)P \equiv (x)(P|Q) \text{ if } x \notin fn(P)$$

Notice the Congruence rule - terms are taken up to congruence

The Fusion Calculus – Bisimulation

A fusion bisimulation *is a symmetric relation* $\mathcal{S}$ *between processes such that whenever* $(P, Q) \in \mathcal{S}$, *if* $P \xrightarrow{\alpha} P'$ *with* $bn(\alpha) \cap fn(Q) = \emptyset$, *then* $Q \xrightarrow{\alpha} Q'$ *and*

- *if* α *is a communication action:* $(P', Q') \in \mathcal{S}$;
- *if* α *is a fusion:* $(P'\sigma, Q'\sigma) \in \mathcal{S}$, *for some* σ *agreeing with* α.

P *and* Q *are* fusion bisimilar *if* $(P, Q) \in \mathcal{S}$ *for some fusion bisimulation* $\mathcal{S}$.

A hyperbisimulation *is a substitution-closed fusion bisimulation, i.e., an* $\mathcal{S}$ *such that* $(P, Q) \in \mathcal{S}$ *implies* $(P\sigma, Q\sigma) \in \mathcal{S}$ *for any substitution* σ. P *and* Q *are* hyper-equivalent, *written* $P \sim Q$, *if they are related by a hyperbisimulation.*

- Ad hoc treatment for fusions: substitution must be performed at each step. Does not fall in the usual "categorical" bisimilarity definition

- Bisimilarity is **not** a congruence; closure under substitutions is needed

In this talk:
A categorical presentation of Fusion

- We want to give a categorical presentation of Fusion calculus:
- Find a category *C* with two endofunctors Σ,*B* : *C* → *C* such that
  - object of processes (up-to alpha) is the initial Σ-algebra *Proc*
  - transition relations of processes are B-coalgebras
  - B has final coalgebra ν B (and is weak pullback preserving)
- We obtain:
  - alternative (more explicit) labelled operational semantics
  - characterization of hyperequivalence as (categorical) B-bisimulation
  - the unique map ⟦ ⟧ : *Proc* → ν B is a fully abstract semantics (by [TP97])
- Also: a methodology for making explicit in the categorical semantics the "hidden states" of operational semantics

10

"Home is where the syntax lives"

- Fusion processes have "lambda"-like binders => standard approach in category of presheaves [FPT99]:
  - $Set^{\mathbb{F}}$ = category of presheaves over $\mathbb{F}$, (skeleton) category of finite sets and functions

$$\Sigma_F : Set^{\mathbb{F}} \to Set^{\mathbb{F}}$$
$$\Sigma_F(A) = 1 + N \times N \times A + N \times N \times A + A \times A + \delta A$$
$$(\Sigma_F(A))_n = \overbrace{1}^{\mathbf{0}} + \overbrace{n \times n \times A_n}^{xy.a} + \overbrace{n \times n \times A_n}^{\bar{x}y.a} + \overbrace{A_n \times A_n}^{a|b} + \overbrace{A_{n+1}}^{(x)a}$$

  - Proc is the initial Σ_F-algebra, that is $Proc_m = \{P \mid fn(P) \subseteq m\}$
- Thus we use $Set^{\mathbb{F}}$ as ambient category

11

What about structural congruence?

- The presheaf approach allows only for *free* algebras (with binders), and does not deal with equations (like e.g. structural congruences)
- (But neither the theory of universal semantics à la Plotkin-Turi does)
- **"Solution"**: define a different but equivalent presentation of the semantics, without congruence
- Does not work always – we need a more general theory for universal semantics, covering also equations and coequations - future work…

12

The Fusion Calculus – Labelled transition system without congruence rule

$$Pref\ \frac{-}{\alpha.P \xrightarrow{\alpha} P}$$

$$Com\ \frac{P \xrightarrow{ux} P',\ Q \xrightarrow{\bar{u}y} Q'}{P|Q \xrightarrow{\{x=y\}} P'|Q'} \qquad Par_l\ \frac{P \xrightarrow{\alpha} P'}{P|Q \xrightarrow{\alpha} P'|Q} \qquad Par_r\ \frac{P \xrightarrow{\alpha} P'}{Q|P \xrightarrow{\alpha} Q|P'}$$

$$Open\ \frac{P \xrightarrow{ux} P',\ u \notin \{x,\bar{x}\}}{(x)P \xrightarrow{u(x)} P'} \qquad Pass\ \frac{P \xrightarrow{\alpha} P',\ x \notin n(\alpha)}{(x)P \xrightarrow{\alpha} (x)P'} \qquad Scope\ \frac{P \xrightarrow{\{x=y\}} P'}{(x)P \xrightarrow{1} P'\{y/x\}}$$

$$Close_l\ \frac{P \xrightarrow{u(x)} P',\ Q \xrightarrow{\bar{u}y} Q'}{P|Q \xrightarrow{1} P'\{y/x\}|Q'} \qquad Close_r\ \frac{P \xrightarrow{ux} P',\ Q \xrightarrow{\bar{u}(y)} Q'}{P|Q \xrightarrow{1} P'|Q'\{x/y\}}$$

$$Close\ \frac{P \xrightarrow{u(x)} P',\ Q \xrightarrow{\bar{u}(x)} Q'}{P|Q \xrightarrow{1} (x)(P'|Q')}$$

The new *Close* rules allow to get rid of structural equivalence

Behaviour functor

- Usually, of the form $BX = \wp_f(Act \times X)$ where *Act* is fixed
- *B*-coalgebras correspond to LTSs: given (A, α:A→*B*(A))

$$P \xrightarrow{a} Q \iff (a, Q) \in \alpha(P)$$

- Each *B* induces *coalgebraic B-bisimulation:* two coalgebras A_1, A_2 are B-bisimilar if there exist R and f_1,f_2 jointly monic such that

$$\begin{array}{ccccc} A_1 & \xleftarrow{f_1} & R & \xrightarrow{f_2} & A_2 \\ \downarrow \alpha_1 & & \exists \vdots \gamma & & \downarrow \alpha_2 \\ BA_1 & \longleftarrow & BR & \longrightarrow & BA_2 \end{array}$$

First attempt

- In our case we could define

$$LX \triangleq \overbrace{N \times N \times X}^{xy} + \overbrace{N \times N \times X}^{\bar{x}y} + \overbrace{N \times \delta X}^{x(y)} + \overbrace{N \times \delta X}^{\bar{x}(y)} + \overbrace{X}^{1} + \overbrace{N \times N \times X}^{x=y}$$
$$BX \triangleq \tilde{K}(LX)$$

where $\tilde{K}$ is Freyd's "finite powerobject" endofunctor (preserves weak pullbacks, needed for universal semantics construction – usual pointwise finite powerset does not preserve weak pullbacks)

- Does **not** work: *B*-bisimilarity ≠ hyperequivalence
- Problem is: *B*-bisimilarity says nothing about substitutions after fusions.
- We have to "correct" *B*.

The approach – in a nutshell

- When a local action has a global effect, a *global state* is hidden somewhere
  - semantics (intentionally) does not mention state => simpler rules, labels
  - at the price of a more complex bisimulation
  - labels do not induce right behavior functor
- Solution:

  1. define the object of states
  2. build it into the naive behaviour functor, using the *state monad*
  3. unravel the coalgebras into "stateful" LTS, to get a simple notion of bisimulation
  4. (same happens for categorical rules)

1. Finding the State

- In Fusion: *state* = *equivalence between names*
- Presheaf of states is *E*

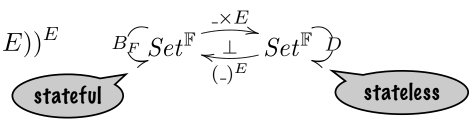
$$E_n \triangleq \{coeq(f,g) \mid k \in \mathbb{F}, f, g : k \rightrightarrows n\}$$

- $E_n \ni e = \{x_1=y_1, \ldots, x_k=y_k\}$
- But also: (class of) surjective substitution $e: n \twoheadrightarrow m$
- Has nice properties (e.g. finitary), and operations (e.g. union is pushout, restriction is composition)

2. Adding state to behaviour functor

- In general, state can be added by lifting along the *(global) state monad*

$$B_F X \triangleq (D(X \times E))^E$$



- In our case:

$$LX \triangleq \overbrace{N \times N \times X}^{xy} + \overbrace{N \times N \times X}^{\bar{x}y} + \overbrace{N \times \delta X}^{x(y)} + \overbrace{N \times \delta X}^{\bar{x}(y)} + \overbrace{X}^{\mathbf{1}} + \overbrace{N \times N \times X}^{x=y}$$
$$DX \triangleq \tilde{K}(LX)$$
$$B_F X = (\tilde{K}(L(X \times E)))^E$$

- B_F seems cumbersome (exponents in presheaf categories)…

3. Simplifying coalgebras

- **Proposition**: B_F-coalgebras (A, α:A→B_F(A)) correspond to (A, β:A×E→*D*(A×E)) such that β is a *D*-coalgebra structure

$$\frac{A \longrightarrow B_F(A) = D(A \times E)^E}{A \times E \longrightarrow D(A \times E)}$$

- A "coalgebra" (A, β:A×E→*D*(A×E)) correspond to "indexed LTS" (A,⟶) where
  - nodes: elements of ∫A×E = {(n,P,e) | (P,e) ∈ A_n×E_n}
  - labels: elements of ∫*Act* = {(n,a) | a ∈ Act_n}
  - subject to some conditions (e.g. closure under substitutions)
- We can denote transitions as

$$(P, e) \xrightarrow{\alpha}_n (Q, d)$$

a "stateful" LTS

From original LTS to indexed LTS

- Original LTS does not mention state => it is not an indexed LTS
- Define $\mathcal{L}_F$ = (Proc, ⟶) as

$$\text{for } \alpha = xy, \bar{x}y, 1: \quad (P, e) \xrightarrow{\alpha}_n (Q, e) \iff P[e] \xrightarrow{\alpha[e]} Q[e]$$

$$\text{for } \alpha = x, \bar{x}: \quad (P, e) \xrightarrow{\alpha}_n (Q, e+1) \iff P[e] \xrightarrow{\alpha[e](z)} Q[e+1];$$

$$(P, e) \xrightarrow{x=y}_n (Q, e \cup x{=}y) \iff P[e] \xrightarrow{(x=y)[e]} Q[e]$$

state is updated

- **Proposition**: $\mathcal{L}_F$ is an indexed LTS and

$$P \xrightarrow{\alpha} Q \iff (P, \emptyset) \xrightarrow{\alpha}_n (Q, e)$$

3 bis. Indexed bisimulation

- A B_F-bisimulation can be presented as a family of symmetric relations

$$\{R_n \subseteq (A_n \times E_n) \times (A_n \times E_n)\}_{n \in N}$$

such that, if $(P, e) R_n (Q, d)$ then

$$(P, e) \xrightarrow{\alpha}_n (P', e') \Rightarrow \exists (Q', d') \in A_{n'} \times E_{n'} s.t.\ (Q, d) \xrightarrow{\alpha}_n (Q', d') \text{ and } (P', e') R_{n'} (Q', d')$$

and vice versa, and closed under substitution:

$$\text{for all } \sigma : n \to m : (P[\sigma], e[\sigma]) R_m (Q[\sigma], d[\sigma])$$

- **Theorem**: P, Q ∈ *Proc*$_n$ are hyperequivalent iff (P,∅) and (Q,∅) are B_F-bisimilar

  Proof hint: substitution-closed bisimulations correspond to indexed families of relations above.

4. Categorical rules for stateful behaviour functors

- Following [TP97,FT01], we have to define

$$\mathcal{S}_X : \Sigma(X \times BX) \longrightarrow BTX \qquad \text{in } \mathit{Set}^{\mathbb{F}}$$

(where *T* is the monad of free Σ-algebras).

- Unfolding *B* and by adjunction, this is equivalent to:

constructor's arguments

their stateful behaviour

$$\mathcal{S}_X : \Sigma(X \times BX) \times E \longrightarrow D(TX \times E)$$

current state

result: labelled transitions with new state

Categorical rules for the Fusion calculus

- expanding Σ, this is the product of the interpretation of each constructor

$$\begin{aligned}
\mathcal{S}_X^{\mathbf{0}} :&\qquad E \longrightarrow D(TX \times E)\\
\mathcal{S}_X^{\mathrm{in}} :&\quad N \times N \times X \times BX \times E \longrightarrow D(TX \times E)\\
\mathcal{S}_X^{\mathrm{out}} :&\quad N \times N \times X \times BX \times E \longrightarrow D(TX \times E)\\
\mathcal{S}_X^{\mathrm{par}} :&X \times BX \times X \times BX \times E \longrightarrow D(TX \times E)\\
\mathcal{S}_X^{\mathrm{res}} :&\quad N \times \delta X \times \delta BX \times E \longrightarrow D(TX \times E)
\end{aligned}$$

each defined by collecting relevant rules, taking care of the state; e.g. for |

$$\rho^{Par}, \rho^{Com}, \rho^{Close_l}, \rho^{Close_r}, \rho^{Close} : X \times BX \times X \times BX \times E \longrightarrow \tilde{K}(\mathit{Act} \times TX \times E)$$

$$\begin{aligned}
\rho_n^{Com}(P, \beta, Q, \gamma, e) \;=\; & \{(x{=}y, P'|Q', e_1 \cup e_2 \cup \{x{=}y\}) \mid \text{for some } z, w \in n, e(z) = e(w) :\\
& ((zx, P', e_1) \in \beta_n(id_n, e) \wedge (\bar{w}y, Q', e_2) \in \gamma_n(id_n, e)) \vee\\
& ((\bar{z}x, P', e_1) \in \beta_n(id_n, e) \wedge (wy, Q', e_2) \in \gamma_n(id_n, e))\}
\end{aligned}$$

Categorical rules – in rule format

- Categorical rules can be written as "stateful LTS"

$$\frac{(P, e) \xrightarrow{ux}_n (P', e_1) \quad (Q, e) \xrightarrow{\bar{w}y}_n (Q', e_2)}{(P|Q, e) \xrightarrow{x=y}_n (P'|Q', e_1 \cup e_2 \cup \{x = y\})} e \vdash u = w$$

- (actually $e_1 = e_2 = e$)

Categorical rules for the Fusion calculus

- **Theorem**: there exists a (necessarily unique)

$$[\![\ ]\!] : Proc \to \nu B_F$$

which is both compositional and fully abstract, that is, for all P,Q ∈ $Proc_n$

$[\![P]\!]_n = [\![Q]\!]$ iff P and Q are B_F-bisimilar (iff are hyperequivalent)

Proof sketch: follows from general results [TP97], by proving that *Proc* can be given a B_F-coalgebra structure, νB_F a Σ-algebra structure, and that B_F is finitary and preserves weak pullbacks.

Conclusions

- Often calculi for distributed systems hide some notion of global state
  - omitted in the LTSs (“stateless”: simpler relation, actions, rules)
  - comes out in bisimulation, and yields unsatisfactory behaviour functor
- State can be added systematically using the state monad
  - Resulting “stateful” coalgebras (LTSs) can be quite simplified
  - For Fusion: state = equivalence of names; yields correct behaviour functor
- **To Do**:
  - other examples (Explicit Fusion, Mobile Ambients, in particular)
  - fit into some theory of *modular mathematical operational semantics*

Thanks!