# Interaction and observation, categorically.

Vincenzo Ciancia

Institute for Logic, Language and Computation
University of Amsterdam

Reykjavik - June 9, 2011

# Summary

We investigate a categorical semantic model for interactive programming languages.

In bisimilarity, we assume an observer who can see the actions performed by the system

Idea: allow the observer to interact with the system while observing it.

# Labelled transition systems

The semantics of interactive systems is often given by a labelled transition system, that is, a set of states $X$ and a transition function

$$f : X \to \mathcal{P}_{fin}(L \times X)$$

Labelled transition systems are equipped with bisimilarity, used as a basic but fundamental equivalence relation for processes.

# Bisimilarity in LTSs

Def: Bisimilarity, on an LTS $(X, f : X \to \mathcal{P}_{fin}(L \times X))$ is the greatest

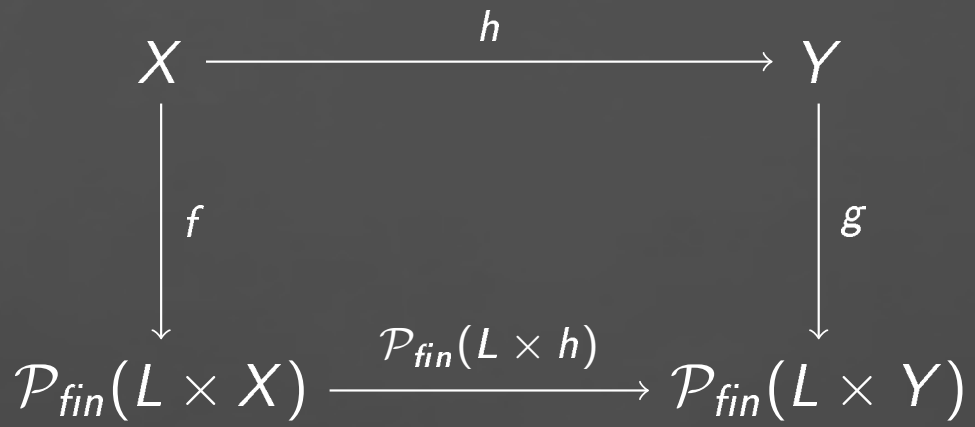
$$- \sim - \subseteq X \times X$$

such that, for all $x \sim y$:

- $x \xrightarrow{\alpha} x' \implies \exists y'.y \xrightarrow{\alpha} y' \wedge x' \sim y'$;
- $y \xrightarrow{\alpha} y' \implies \exists x'.x \xrightarrow{\alpha} x' \wedge x' \sim y'$.

# Bisimilarity from kernels of homomorphisms

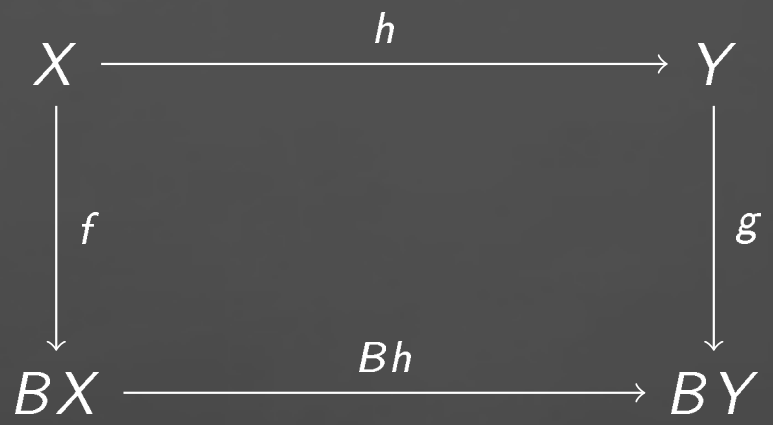
Thm: $x \simeq y$ if and only if there is an LTS $(Y, g : Y \to \mathcal{P}_{fin}(L \times Y))$ and a function $h : X \to Y$ such that $h(x) = x(y)$ and:



# Coalgebras

We fix a category $C$ and assume it is $Set$ for clarity.

For each endofunctor $B : Set \to Set$, we define a category of $B$-coalgebras, having objects $(X, f : X \to BX)$ and arrows commuting in:



# Coalgebraic bisimilarity

The equivalence relation induced by kernels of morphisms is called coalgebraic bisimilarity.

Def: Coalgebraic bisimilarity on a coalgebra $(X, f)$ is the relation $\approx \subseteq X \times X$ induced by the kernel of homomorphisms:

$$x \approx y \iff \exists (Y, g).\exists h : (X, f) \to (Y, g).h(x) = h(y)$$

# Various kinds of coalgebras in the category *Set*

| Functor $F(X)$ | Coalgebras | Equivalence induced by kernels |
|---|---|---|
| $L \times X$ | Deterministic systems | same stream of observations |
| $2 \times (X^L)$ | Deterministic automata | language equivalence |
| $\mathcal{P}_{fin}(L \times X)$ | LTSs | bisimilarity |
| $(\mathcal{D}(X) + 1)^L$ | probabilistic LTSs | probabilistic bisimilarity |

See also coalgebras in $Set^C$ for some index category $C$: resource-allocating bisimilarity (e.g. the $\pi$-calculus).

Coalgebras describe the possible observations on elements of the carrier $X$.

# Algebras

Coalgebras are often contrasted with algebras.

An $F$-algebra is a set $X$ equipped with a function $f : F(X) \to X$.

E.g. $F(X) = X \times X + 1$ - signature for monoids:

$$f : F(X) \to X \qquad f = [f_1 : X \times X \to X; f_2 : 1 \to X]$$

Given elements $x, y$, one can build the element $f_1(x, y)$. That is, $(X, f)$ is a model of the signature of monoids.

Algebras describe how to build new elements from existing elements.

# Mealy machines

A Mealy Machine is a function $f : I \times X \to O \times X$ for a set of states $X$, a set of possible inputs $I$ and of possible outputs $X$.

At each step, the machine accepts an input, and returns an output and a next state.

# Dialgebras

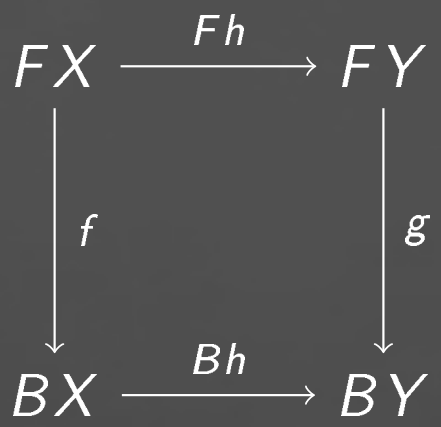
Def: An $(F, B)$-dialgebra is a pair $(X, f)$ where $X$ is an object, and

$$f : FX \to BX$$

for two endo-functors $F$ and $B$.

# Category of dialgebras

$(F, B)$-dialgebras form a category whose morphisms $h : (X, f) \to (Y, g)$ are those arrows $h : X \to Y$ that commute in:



# Dialgebraic bisimilarity

Def: Dialgebraic bisimilarity on a dialgebra $(X, f)$ is the relation $\approx \subseteq X \times X$ induced by the kernel of homomorphisms:

$$x \approx y \iff \exists (Y, g).\exists h : (X, f) \to (Y, g).h(x) = h(y)$$

Rem: $B$-coalgebras are $(F, B)$-dialgebras with $F = Id$, the identity functor. Coalgebraic bisimilarity is dialgebraic bisimilarity.

# Examples?

First examples:

| $F(X)$ | $B(X)$ | Dialgebras |
|---|---|---|
| $F$ | $Id$ | Algebras for the functor $F$ |
| $Id$ | $B$ | Coalgebras for the functor $B$ |
| $I \times X$ | $O \times X$ | Deterministic Mealy machines |
| $I \times X$ | $\mathcal{P}_{fin}(O \times X)$ | Non-deterministic Mealy machines |
| $I \times X$ | $\mathcal{D}(O \times X)$ | Probabilistic Mealy machines |

# "Effectful" algebraic operations

Compare dialgebras to algebras and to coalgebras. Algebras describe operations, coalgebras describe observations or side effects.

A $(F, B)$-dialgebra is the implementation of operations that yield observations - or side effects - and future states. E. g. one may define a binary "algebraic" operation

$$f : X \times X \to L \times X$$

Given two elements $x, y$, one can build an element $\pi_2(f(x))$ observing the effect $\pi_1(f(x))$.

Dialgebras specify side-effecting operations.

# The asynchronous CCS

Let $c$ range over a countable set $C$ of channel names.

Syntax of processes:

| | | | |
|---|---|---|---|
| $P$ | $::=$ | $\emptyset$ | inactive process |
| | $\mid$ | $\tau.P$ | internal action |
| | $\mid$ | $c.P$ | wait for signal on $c$ |
| | $\mid$ | $\bar{c}$ | send signal on $c$ |
| | $\mid$ | $P \parallel P$ | parallel composition |
| | $\mid$ | $P + Q$ | choice |
| | $\mid$ | $!P$ | replication |

# Operational semantics

$$c.P \xrightarrow{c} P\,(in) \qquad \tau.P \xrightarrow{\tau} P\,(tau) \qquad \bar{c} \xrightarrow{\bar{c}} \emptyset\,(out)$$

$$\frac{P \xrightarrow{\alpha} P'}{P \parallel Q \xrightarrow{\alpha} P' \parallel Q}\,(par) \qquad \frac{Q \xrightarrow{\alpha} Q'}{P \parallel Q \xrightarrow{\alpha} P \parallel Q'}\,(par')$$

$$\frac{P \xrightarrow{c} P' \quad Q \xrightarrow{\bar{c}} P'}{P \parallel Q \xrightarrow{\tau} Q' \parallel Q'}\,(syn) \qquad \frac{P \xrightarrow{\alpha} P'}{!P \xrightarrow{\alpha} P' \parallel !P}\,(rep)$$

$$\frac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'}\,(sum) \qquad \frac{Q \xrightarrow{\alpha} Q'}{P + Q \xrightarrow{\alpha} Q'}\,(sum')$$

# Asynchronous Bisimilarity

Def: Asynchronous bisimilarity is the greatest relation $R \subseteq X \times X$ such that, whenever $(x, y) \in R$, and $x \xrightarrow{\alpha} x'$, there is $y'$ such that:

- if $\alpha = \tau$ or $\alpha = \bar{c}$ for some $c$, then $y \xrightarrow{\alpha} y'$ and $(x', y') \in R$;
- if $\alpha = c$ for some $c$, then $\bar{c} \parallel y \xrightarrow{\tau} y'$ and $(x', y') \in R$
  or, equivalently
  if $\alpha = c$ for some $c$, then there is $(x', y') \in \mathcal{R}$ such that either $y \xrightarrow{c} y'$ or $y \xrightarrow{\tau} y''$ with $y' = \bar{c} \parallel y''$.

(+ the symmetric case)

# Asynchronous Bisimilarity

Example: $a.\bar{a} + \tau$ and $\tau$

$a.\bar{a}$ can read a value, making it immediately available to the other parallel components for reading. From the outside, it's the same as seeing just an internal computation.

Input actions are not observable by themselves, but only depending on their continuation (in this case, $\bar{a}$).

# Interaction functor

Syntax of the experiments of the observer, described by the functor

$$FX = X + (\bar{C} \times X)$$

(where $\bar{C} = \{\bar{c} \mid c \in C\}$).

$P \in FX$: the observer can let the system run and see what happens

$(\bar{c}, P) \in FX$: the observer can send a signal on channel $c$ to process $P$.

# Observation functor

Syntax of the observations, described by the functor

$$BX = \mathcal{P}_{fin}\left((\{\tau\} + \bar{C}) \times X\right)$$

$(\tau, P) \in p \in BX$: the observer sees an internal computation step.

$(\bar{c}, P) \in p \in BX$: the observer sees the output of a signal on channel $c$.

# Theorem

Let $F$ and $B$ be the functors we defined above.

Thm: Dialgebraic bisimilarity over an $(F, B)$-dialgebra $(X, f)$ is the greatest relation $\mathcal{R} \subseteq X \times X$ such that, for all $(x, y) \in \mathcal{R}$ and $c \in C$:

1. whenever $x \xrightarrow{\alpha}_f x'$, there is $y'$ such that such that $y \xrightarrow{\alpha}_f y'$ and $(x', y') \in \mathcal{R}$;
2. whenever $(\bar{c}, x) \xrightarrow{\tau}_f x'$, there is $y'$ such that $(\bar{c}, y) \xrightarrow{\tau}_f y'$ and $(x', y') \in \mathcal{R}$.

(+ symmetric condition)

# Dialgebra for the asynchronous CCS

We define a dialgebra $f : FX \to BX$, where $X$ is the set of CCS processes; we use the LTS of the operational semantics in premises of rules. An inductive definition that does not use an auxiliary LTS is possible too.

$$\frac{x \xrightarrow{\alpha} x' \quad \alpha = \tau \vee \alpha = \overline{c}}{x \xrightarrow{\alpha}_f x'} \, (run)$$

$$\frac{x \xrightarrow{c} x'}{(c, x) \xrightarrow{\tau}_f x'} \, (in)$$

$$\frac{x \xrightarrow{\tau} x'}{(c, x) \xrightarrow{\tau}_f \overline{c} \parallel x'} \, (store)$$

# Theorem

Thm: Dialgebraic bisimilarity coincides with asynchronous bisimilarity.

# More complex interactions

The example is aimed to explain how dialgebras can be used to model "input as input" in a natural way.

Rules express precisely what one can observe when the system reads data from the environment.

More complex notions of "experiments" are obtained by considering richer interaction functors.

# Example: "chemical" reactions

Consider the finite multi-set functor

$$\mathcal{M}(X) = \{m : X \to \mathbb{N} | \{x \mid m(x) \neq 0\} \text{is finite}\}$$

Think of $X$ as a set of elements that take part in reactions in variable quantities. A dialgebra

$$f : \mathcal{M}(X) \to \mathcal{M}(X)$$

specifies how a given reaction evolves.

# Example: complex systems

Consider dialgebras of the form

$$f : \mathcal{P}_{fin}(X) \to L \times \mathcal{P}_{fin}(X)$$

At each step in time, from a set $p \in \mathcal{P}_{fin}(X)$, a side effect in $L$ is observed, and a new set of elements $p'$ is obtained.

They represent systems where the semantics depends on a number of entities that collaborate. The behaviour of the system is more than the sum of its parts.

# Final dialgebras?

Consider categories of dialgebras where $F$ makes use of some $k$-ary product with $k > 1$.

Example: $f : X \times X \to L \times X$ modelling some set equipped with a binary side-effecting operator with effects in $L$.

The category of $(- \times -, L \times -)$-dialgebras has no final object.
Consequence: lack of a unique denotational domain for all the possible dialgebras.

# Bisimilarity quotients.

A unique denotational model has the power to compare all systems. However we are typically interested in defining a relation on some systems, e.g. the CCS processes.

Adopting this point of view, one may resort to bisimilarity quotients instead of on a final object.

Q: categorical algorithm (similar to iteration along the terminal sequence for coalgebras) that computes the bisimilarity quotient of a system?

# Future work

Logics. One can reason on elements of algebras by using equational logic, and on elements of coalgebras by using modal logic.
Q: what is a suitable logic for reasoning on dialgebras?

Dialgebras defined by induction. In bialgebras the semantic equivalence is always a congruence with respect to certain algebraic operators. Can we prove similar results for dialgebras defined by induction? (e.g. on the syntax of CCS processes).

Examples not from process algebras. See the examples of "chemical" reactions and chaotic behaviour.