


Intro. to Categorical Algebra / Coalgebra in Comp. Sci.



Ichiro Hasuo 
Dept. CS, U. Tokyo
www-mmm.is.s.u-tokyo.
ac.jp

Part I.

Introduction:

Induction and Coinduction.

References:

- Bart Jacobs, Introduction to Coalgebra (a draft  of a book, available on his website)
- Dirk Pattinson, NASSLLI 2003 Lecture Notes  (available on his website)

Induction and Coinduction

1.

§1. Intro.

You all are familiar with

(def. / proof.) by induction ...

Ex. 1.1. (Inductive def. of terms)

$\Sigma = (\Sigma_n)_{n \in \mathbb{N}}$. Σ_n : a set. of n-ary operation symbols

The set $T_{\Sigma X}$ of Σ -terms over variables from X is defined by

$$T_{\Sigma X} \ni t ::= x \mid f(t_1, \dots, t_n) \\ (x \in X) \quad (f \in \Sigma_n)$$

That is,

$$\frac{}{x \in T_{\Sigma X}} \quad (x \in X)$$

$$\frac{t_1 \in T_{\Sigma X} \quad \dots \quad t_n \in T_{\Sigma X}}{f(t_1, \dots, t_n) \in T_{\Sigma X}} \quad (f \in \Sigma_n)$$

That is equivalently:



Ex. 1.2 (Inductive def. of a function)

2.

Let A be a Σ -algebra with a X -valuation

$J: X \rightarrow A$. That is,

- For each $n \in \mathbb{N}$ and $f \in \Sigma_n$,

$$\llbracket f \rrbracket: A^n \rightarrow A$$

- $J: X \rightarrow A$ is a function.

We define the map

$$\llbracket - \rrbracket_J: T_\Sigma X \rightarrow A$$

by:

- $\llbracket x \rrbracket_J := J(x) \quad (x \in \text{Var})$

- $\llbracket f(t_1, \dots, t_n) \rrbracket_J := \frac{\llbracket f \rrbracket}{f} (\llbracket t_1 \rrbracket_J, \dots, \llbracket t_n \rrbracket_J)$

The interpretation
of $f \in \Sigma_n$ in A

That is,

$$\llbracket \begin{array}{c} \boxed{f} \\ \swarrow \quad \searrow \\ \triangle t_1 \quad \dots \quad \triangle t_n \end{array} \rrbracket_J := \llbracket f \rrbracket (\llbracket \triangle t_1 \rrbracket_J, \dots, \llbracket \triangle t_n \rrbracket_J)$$

Ex. (Proof by Induction)

$t \in T_{\Sigma} X$ with $FO(t) \subseteq X$

$J, J': X \rightarrow A,$

valuations

with an obvious inductive def.

free var's

$$J(x) = J'(x), \quad \forall x \in FO(t)$$

Then $\llbracket t \rrbracket_J = \llbracket t \rrbracket_{J'}$

Proof. \hookrightarrow syntactic eq.

- If $t \equiv x \in \text{Var}$, $FO(t) = \{x\}$ and

$$\llbracket t \rrbracket_J = J(x) \underset{\text{asmp.}}{=} J'(x) = \llbracket t \rrbracket_{J'}$$

- If $t \equiv f(t_1, \dots, t_n),$

$FO(t_i) \subseteq FO(t)$, thus by ind. hyp.

$$\llbracket t_i \rrbracket_J = \llbracket t_i \rrbracket_{J'}, \quad \forall i \in [1, n]$$

Use this in:

$$\llbracket t \rrbracket_J = \llbracket f \rrbracket (\llbracket t_1 \rrbracket_J, \dots, \llbracket t_n \rrbracket_J)$$

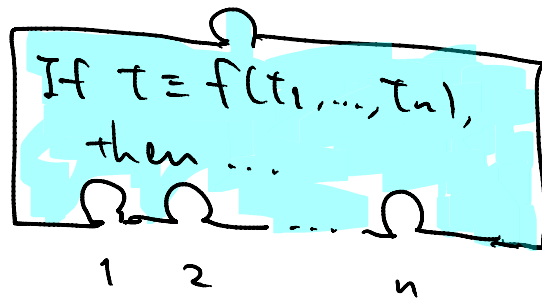
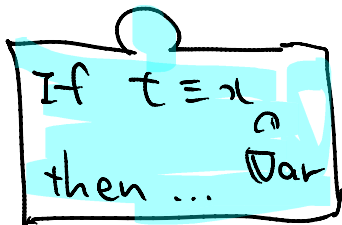
$$\stackrel{\text{above}}{=} \llbracket f \rrbracket (\llbracket t_1 \rrbracket_{J'}, \dots, \llbracket t_n \rrbracket_{J'})$$

$$= \llbracket t \rrbracket_{J'}$$

□

Inductive proof, schematically

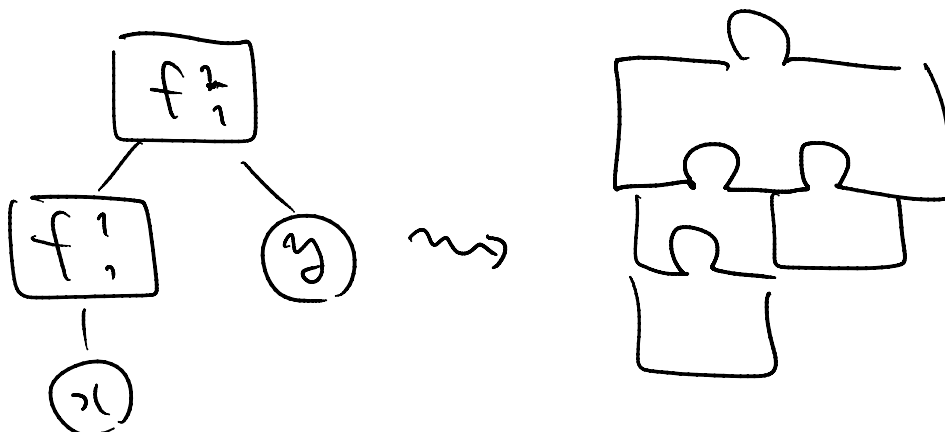
1. I provide the modules for proofs, namely



2. and I leave the use of these to somebody else

“You have a term $t \in T_{\Sigma X}$;
for which you can use the p/f
modules I give you. This'll do.
Good luck, I'm now going out
for beer...”

For example:



We now exhibit a dual notion of Coinduction.

- What is similar:
 - * data structure
 - * Defs / P'fs by "modules"
 - What is not similar:
 - * Well-founded vs. non well-fdd
 - * Possible infinity vs. real infinity
-

Let us now turn to data that are "truly infinite" ...

Def. 1.4 A stream over L is

$a_0 a_1 a_2 \dots$ with $a_i \in L$
that is, a function

$$a: \begin{array}{c} \mathbb{N} \\ \parallel \\ \omega \end{array} \rightarrow L$$

The set of streams over L is denoted by L^ω .

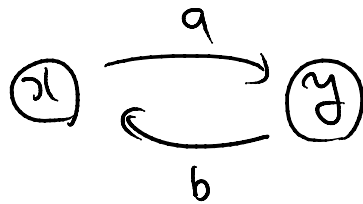
Ex. 1.5 (Def. by coinduction)

- The stream $abab \dots$ ($a, b \in L$)

What is its "precise" / ("finitary") /
"useful"

Try this: definition?

* A "state machine" / "automaton"



* $abab \dots$ as its "behavior,"

i.e. the stream it produces
(eventually)

- The stream $aa \dots$ is by

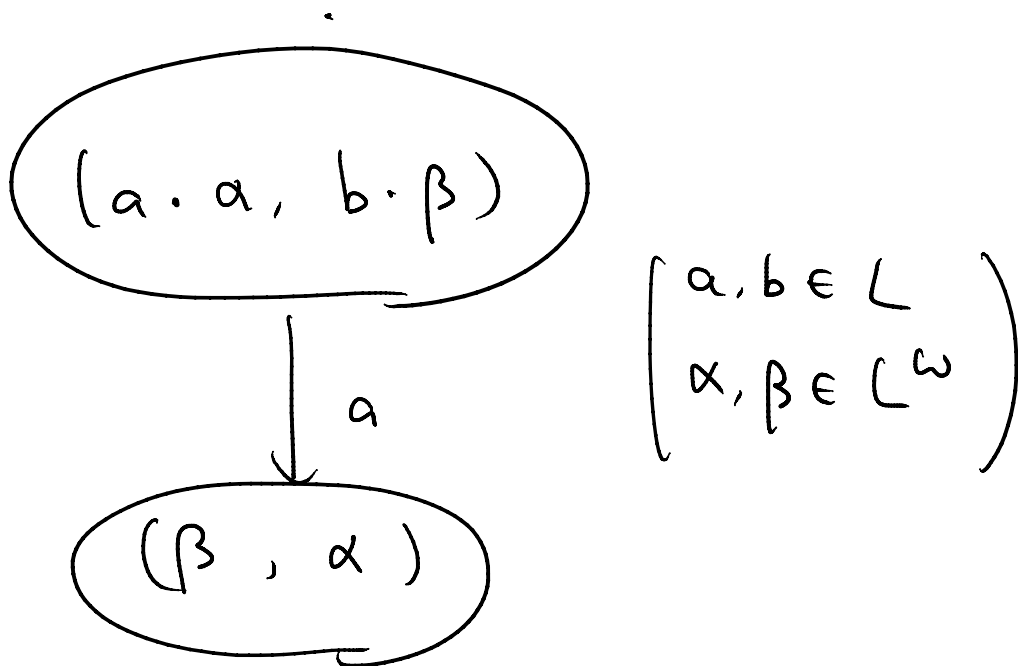


(Ex. 1.5 cont'd)

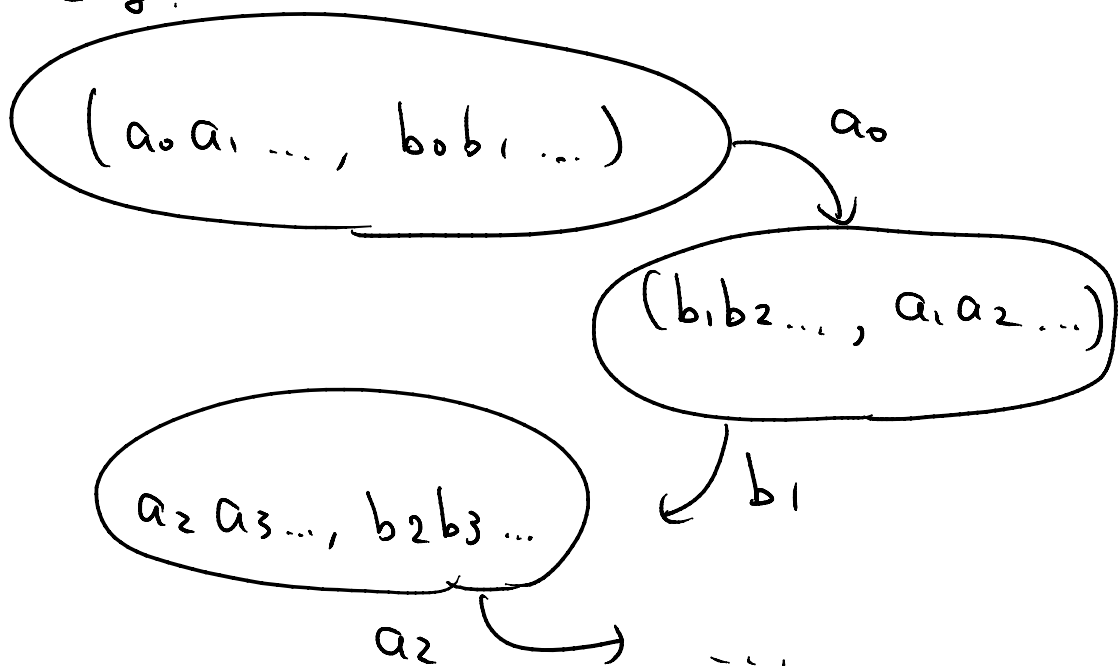
- alt: $L^{\omega} \times L^{\omega} \longrightarrow L^{\omega}$

$(a_0 a_1 \dots, b_0 b_1 \dots) \mapsto a_0 b_1 a_2 b_3 \dots$

is defined, or "implemented", by the "state machine"



E.g.



Point

Represent a stream by means
of a certain kind of an automaton

Def. (Stream automaton)

An L-stream automaton is

$$\left(\begin{array}{c} X \\ \uparrow \\ \text{a set} \end{array}, \begin{array}{c} C: X \rightarrow L \times X \\ \uparrow \\ \text{"transition function"} \\ \underline{C(x) = (a, x')} \\ \underline{\underline{x \xrightarrow{a} x'}} \end{array} \right)$$

Ex. (proof by coinduction)

Claim. $\text{alt}(\alpha, \alpha) = \alpha$

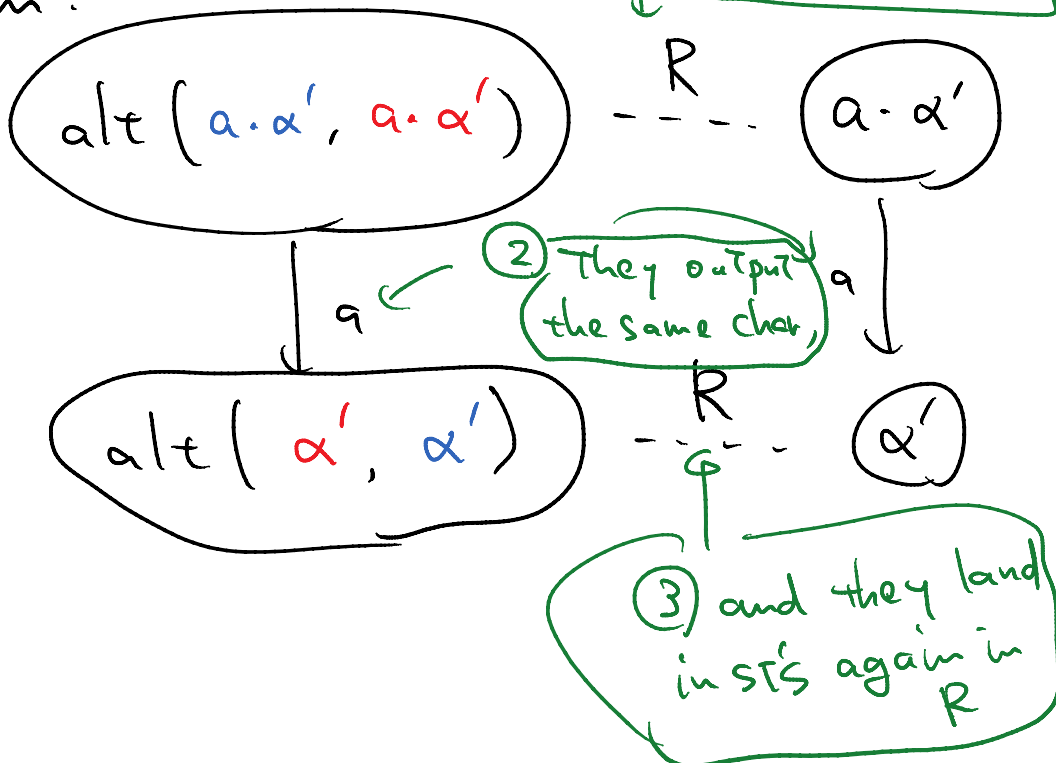
Intuitively obvious:
 $\alpha = a_0 a_1 a_2 \dots$
 $\text{alt}(\alpha, \alpha) = \overline{a_0} \underline{a_1} \overline{a_2} \underline{a_3} \dots$
But: "precise proof"?
(generalizes)

We define

$$\mathcal{R} = \left\{ (\text{alt}(\alpha, \alpha), \alpha) \mid \alpha \in L^w \right\}$$

$\mathcal{R} \subseteq L^w \times L^w$

Then:



The p'f says:

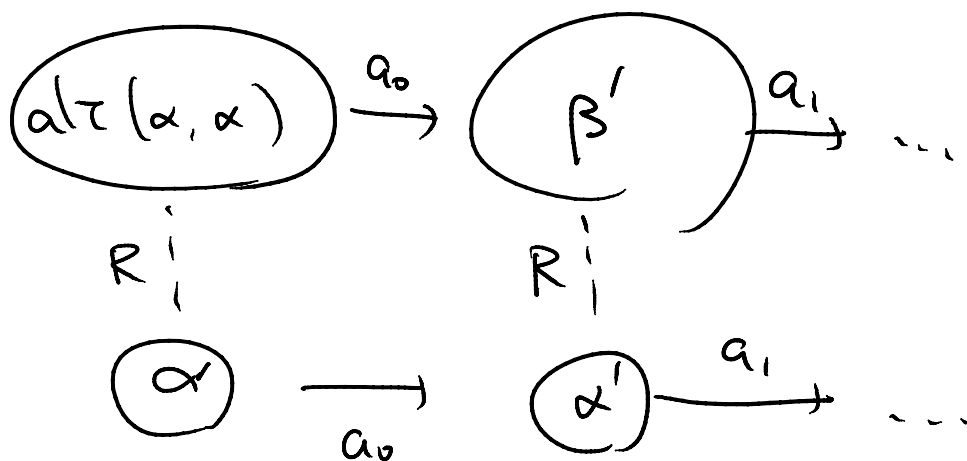
66

Here is

$R \subseteq L^w \times L^w$ s.t.

- $\alpha R \beta \Rightarrow$ the first char's of α, β are the same
- $\alpha R \beta$
 $\alpha = a \cdot \alpha'$
 $\beta = b \cdot \beta'$ $\Rightarrow \alpha' R \beta'$
- $(\text{alt}(\alpha, \alpha), \alpha) \in R$

You have time, so you can try



I'm mortal and I prefer going out for beer."

So this R is like a proof

module , or a witness .

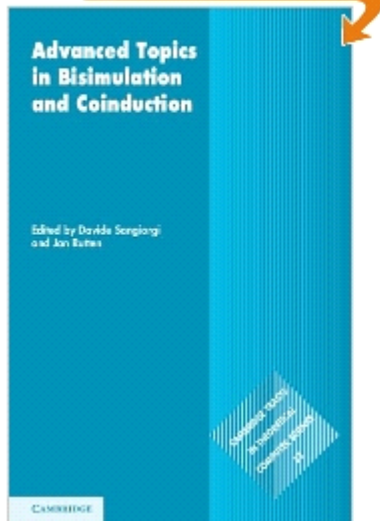
bisimulation

画面の領域の取り込み日時: 2012/07/17 10:49

This argument with a relation R as an invariant — “initially R , and R is preserved under a transition” — is one of the several different things referred to as coinduction.

They include: the argument using largest fixed pt.
We'll see that they are all instances of the categorical principle of coinduction

クリック なか見! 検索



Advanced Topics in Bisimulation and Coinduction (Cambridge Tracts in Theoretical Computer Science) [ハードカバー]

[Davide Sangiorgi](#) (編集), [Jan Rutten](#) (編集)

[この商品の最初のレビューを書き込んでください。](#)

いいね (0)

価格: ¥ **7,795** 通常配送無料 [詳細](#)

在庫あり。 [在庫状況](#)について

この商品は、[Amazon.co.jp](#) が販売、発送します。ギフトラッピングを利用できません。

4点在庫あり。ご注文はお早めに。

7/18 水曜日 にお届けします。関東へのお届けの場合、今から**16時間と40分**以内に「[お急ぎ便](#)」または「[当日お急ぎ便](#)」を選択して注文を確定してください(有料オプション。[Amazonプライム会員](#)は無料)。関東以外の場合は、注文確定画面でお届け予定日をご確認ください。

Summary

- Induction / Coinduction are two principal methods for dealing with infinity.
- * But with different flavours ...
 - ① induction
 - ⇒ "possible infinity"
 - well-fdd
 - ② coinduction
 - ⇒ "real infinity"
 - non-well-fdd
- * CS indeed involve the latter aspect (cf. while loops, μ , ...)
- We now present a categorical formalization of these