Algebraically Enriched Coalgebras

Filippo Bonchi[4] Marcello Bonsangue[1,2] Jan Rutten[1,3]
Alexandra Silva[1]

[1]Centrum Wiskunde en Informatica
[2]LIACS - Leiden University
[3]Radboud Universiteit Nijmegen
[4]INRIA Saclay - LIX, École Polytechnique

CMCS, March 2010

Motivation

- One of the nice things about (modelling systems as) coalgebras:

  The type of the system determines a canonical notion of equivalence.

  e.g bisimilarity for LTS's

- One of the not so nice things about coalgebras:

  The canonical notion of equivalence is not what one wants.

  e.g language equivalence for LTS's

  Goal of this talk: Show a way of uniformly deriving a new set of canonical equivalences from the type of the system.

Motivation

- One of the nice things about (modelling systems as) coalgebras:

  The type of the system determines a canonical notion of equivalence.

  e.g bisimilarity for LTS's

- One of the not so nice things about coalgebras:

  The canonical notion of equivalence is not what one wants.

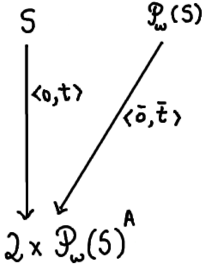
  e.g language equivalence for LTS's

  Goal of this talk: Show a way of uniformly deriving a new set of canonical equivalences from the type of the system.

Example I: Determinizing (coalgebraically)

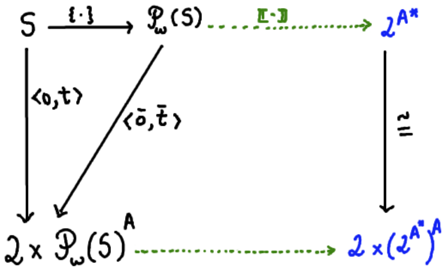
$$S \xrightarrow{\langle o,t \rangle} 2 \times \mathcal{P}_\omega(S)^A$$

Example I: Determinizing (coalgebraically)



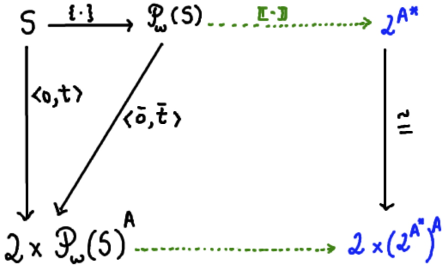
$$\overline{o}(Q) = \begin{cases} 1 & \exists_{q \in Q} o(q) = 1 \\ 0 & \text{otherwise} \end{cases} \qquad \overline{t}(Q)(a) = \bigcup_{q \in Q} t(q)(a)$$

Example I: Determinizing (coalgebraically)



$$\overline{o}(Q) = \begin{cases} 1 & \exists_{q \in Q} o(q) = 1 \\ 0 & \text{otherwise} \end{cases} \qquad \overline{t}(Q)(a) = \bigcup_{q \in Q} t(q)(a)$$

Example I: Determinizing (coalgebraically)

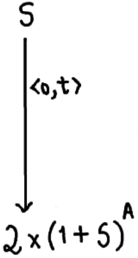


$$\overline{o}(Q) = \begin{cases} 1 & \exists_{q \in Q} o(q) = 1 \\ 0 & \text{otherwise} \end{cases} \qquad \overline{t}(Q)(a) = \bigcup_{q \in Q} t(q)(a)$$

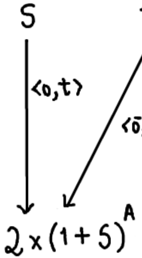
How do we study NDA wrt language equivalence?

$$L_s = [\![\{s\}]\!]$$

Example II: Totalizing

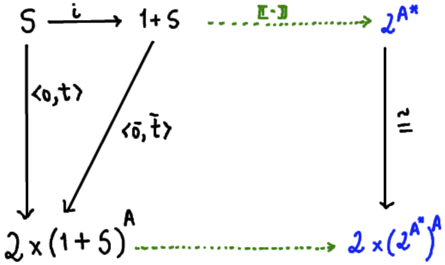


Example II: Totalizing



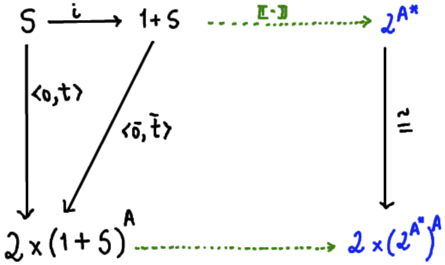
$$\begin{cases} \overline{o}(*) = 0 \\ \overline{o}(s) = o(s) \end{cases} \qquad \begin{cases} \overline{t}(*)(a) = * \\ \overline{t}(s)(a) = t(s)(a) \end{cases}$$

Example II: Totalizing



$$\begin{cases} \overline{o}(*) = 0 \\ \overline{o}(s) = o(s) \end{cases} \qquad \begin{cases} \overline{t}(*)(a) = * \\ \overline{t}(s)(a) = t(s)(a) \end{cases}$$

Example II: Totalizing



$$\begin{cases}\overline{o}(*) = 0\\ \overline{o}(s) = o(s)\end{cases} \qquad \begin{cases}\overline{t}(*)(a) = *\\ \overline{t}(s)(a) = t(s)(a)\end{cases}$$

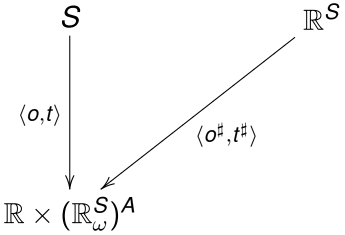
How do we study PA wrt language equivalence?

$$L_s = [\![\, i(s)\,]\!]$$

Example III: Linearization

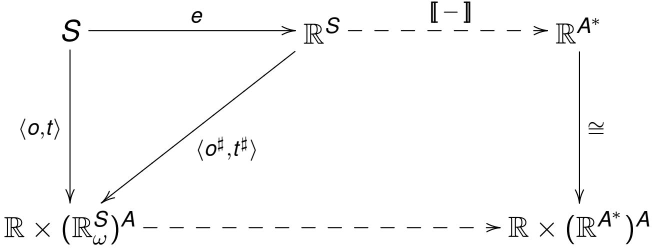
$$\begin{array}{c} S \\ \Big\downarrow \langle o,t \rangle \\ \mathbb{R} \times (\mathbb{R}_\omega^S)^A \end{array}$$

Example III: Linearization



$$o^\sharp\left(\begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix}\right) = \sum v_i \times o(s_i) \qquad t^\sharp\left(\begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix}\right)(a)(s_j) = \sum v_i \times t(s_i)(a)(s_j)$$

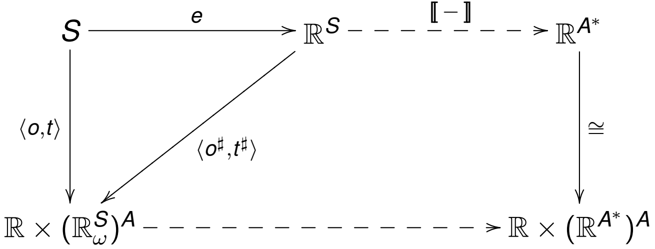
Example III: Linearization



$$e(s_i) = \begin{matrix} s_1 \\ \vdots \\ s_i \\ \vdots \\ s_n \end{matrix} \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix}$$

$$o^\sharp(\begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix}) = \sum v_i \times o(s_i) \qquad t^\sharp(\begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix})(a)(s_j) = \sum v_i \times t(s_i)(a)(s_j)$$

Example III: Linearization



$$e(s_i) = \begin{matrix} s_1 \\ \vdots \\ s_i \\ \vdots \\ s_n \end{matrix} \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix}$$

$$o^\sharp(\begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix}) = \sum v_i \times o(s_i) \qquad t^\sharp(\begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix})(a)(s_j) = \sum v_i \times t(s_i)(a)(s_j)$$

How do we study WA wrt weighted languages (linear bisimilarity)?

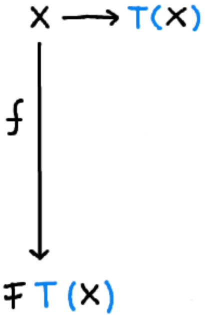
$$L_s = [\![\, e(s) \,]\!]$$

Chasing the pattern. . .

How do we capture all the examples (and more) in the same framework?

Chasing the pattern. . .

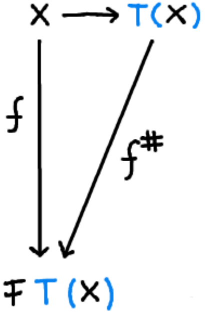
How do we capture all the examples (and more) in the same framework?



The state space was *enriched* : T monad ($\mathcal{P}$, $1+$, . . .).

Chasing the pattern...

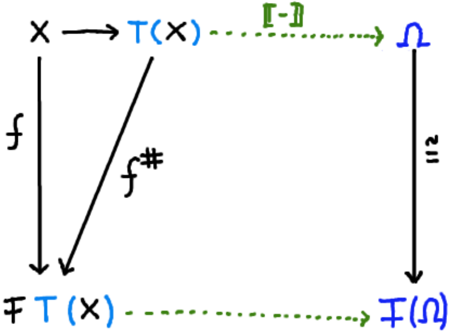
How do we capture all the examples (and more) in the same framework?



The state space was *enriched* : T monad ($\mathcal{P}$, $1+$, ...).
Transform an FT-coalgebra (X,f) into an F-coalgebra ($T(X), f^\sharp$).

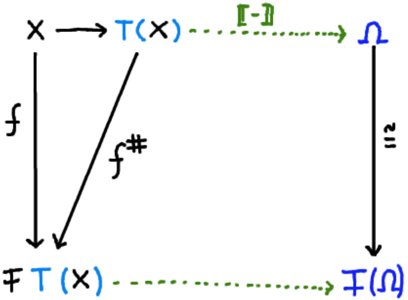
Chasing the pattern...

How do we capture all the examples (and more) in the same framework?



The state space was *enriched* : T monad ($\mathcal{P}$, $1+$, ...).
Transform an FT-coalgebra (X,f) into an F-coalgebra $(T(X), f^\sharp)$.
If F has final coalgebra: $x_1 \approx_F^T x_2 \Leftrightarrow [\![\eta_X(x_1)]\!] = [\![\eta_X(x_2)]\!]$.

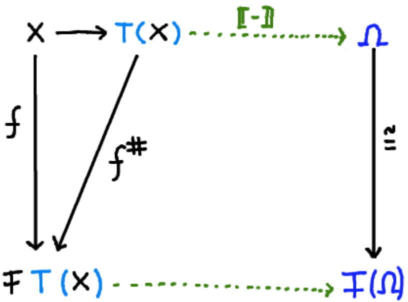
In a nutshell. . .



Ingredients:

- A monad T;
- A final coalgebra for F (for instance, take F to be bounded);
- An extension $f^\sharp$ of f;

In a nutshell. . .



Ingredients:

- A monad T;
- A final coalgebra for F (for instance, take F to be bounded);
- An extension $f^\sharp$ of f; We can require $FT(X)$ to be a T-algebra: $(FT(X), h\colon T(FT(X)) \to FT(X))$

$$f^\sharp\colon T(X) \xrightarrow{T(f)} T(F(T(X))) \xrightarrow{h} F(T(X))$$

Bisimilarity implies T-enriched bisimilarity

Theorem

$$\sim_{FT} \quad \Rightarrow \quad \approx_F^T$$

Bisimilarity implies T-enriched bisimilarity

Theorem

$$\sim_{FT} \quad \Rightarrow \quad \approx_F^T$$

The above theorem instantiates to well known facts:

- for NDA ($F(X) = 2 \times X^A$, $T = \mathcal{P}$) that bisimilarity implies language equivalence;
- for PA ($F(X) = 2 \times X^A$, $T = 1 + -$) that equivalences of pair of languages, consisting of defined paths and accepted words, implies equivalence of accepted words;
- for weighted automata ($F(X) = \mathbb{R} \times X^A$, $T = \mathbb{R}_\omega^-$) that weighted bisimilarity implies weighted language equivalence.

Examples, Examples, Examples,. . .

- **Partial Mealy machines** $S \to (B \times (1+S))^A$;
- **Automata with exceptions** $S \to 2 \times (E+S)^A$;
- **Automata with side effects** $S \to E^E \times ((E \times S)^E)^A$;
- **Total subsequential transducers** $S \to O^* \times (O^* \times S)^A$;
- **Probabilistic automata** $S \to [0, 1] \times (\mathcal{D}_\omega(X))^A$;
- . . .

Conclusions

- Lifted *powerset construction* to the more general framework of FT-coalgebras;
- Uniform treatment of several types of automata, recovery of known constructions/results;
- Opens the door to the study of T-*enriched equivalences* for many types of automata.

Thanks!!

Conclusions

- Lifted *powerset construction* to the more general framework of FT-coalgebras;
- Uniform treatment of several types of automata, recovery of known constructions/results;
- Opens the door to the study of *T-enriched equivalences* for many types of automata.

Thanks!!

The relation with [HJS]

1. Some examples do not fit their framework (e.g., interactive output monad is not commutative, side-effect monad has no $\bot$,...); some of our examples might not fit our framework (?);
2. If $FT \cong TG$ (e.g $2 \times \mathcal{P}(-)^A \cong \mathcal{P}(1 + A \times -)$) then:

$$x \sim_{tr} y \iff x \approx_F^T y$$

If $\rho\colon TG \Rightarrow FT$ then:

$$x \sim_{tr} y \Rightarrow x \approx_F^T y$$