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)

\[ S \xrightarrow{\langle 0, t \rangle} \mathbb{2} \times \mathcal{P}_\omega(S)^A \]

How do we study NDA wrt language equivalence?
Example I: Determinizing (coalgebraically)

\[ \overline{o}(Q) = \begin{cases} 
1 & \exists q \in Q \ o(q) = 1 \\
0 & \text{otherwise} 
\end{cases} \]

\[ \overline{t}(Q)(a) = \bigcup_{q \in Q} t(q)(a) \]

How do we study NDA wrt language equivalence?

Alexandra Silva (CWI)
Algebraically Enriched Coalgebras
CMCS, March 2010
Example I: Determinizing (coalgebraically)

\[
\begin{align*}
\bar{o}(Q) &= \begin{cases} 
1 & \exists q \in Q \bar{o}(q) = 1 \\
0 & \text{otherwise}
\end{cases} \\
\bar{t}(Q)(a) &= \bigcup_{q \in Q} t(q)(a)
\end{align*}
\]
Example I: Determinizing (coalgebraically)

\[
\overline{o}(Q) = \begin{cases} 
1 & \exists q \in Q \ o(q) = 1 \\
0 & \text{otherwise}
\end{cases}
\]

\[
\overline{t}(Q)(a) = \bigcup_{q \in Q} t(q)(a)
\]

How do we study NDA wrt language equivalence?

\[
L_s = \llbracket \{s\} \rrbracket
\]
Example II: Totalizing

\[ S \xrightarrow{\langle 0, t \rangle} 2 \times (1 + 5)^A \]

How do we study PA wrt language equivalence?
Example II: Totalizing

\[
\begin{align*}
\bar{o}(\ast) &= 0 \\
\bar{o}(s) &= o(s) \\
\bar{t}(\ast)(a) &= \ast \\
\bar{t}(s)(a) &= t(s)(a)
\end{align*}
\]
Example II: Totalizing

\[ S \xrightarrow{i} 1 + S \xrightarrow{=} \mathcal{P}(\mathcal{P}(A)) \]

\[ \langle 0, t \rangle \]

\[ \langle 0, \bar{t} \rangle \]

\[ 2^\times (1 + S) \xrightarrow{=} 2^\times (\mathcal{P}(A)) \]

\[
\begin{align*}
\bar{o}(\ast) &= 0 \\
\bar{o}(s) &= o(s) \\
\bar{t}(\ast)(a) &= \ast \\
\bar{t}(s)(a) &= t(s)(a)
\end{align*}
\]
Example II: Totalizing

\[ \overline{o}(\ast) = 0 \]
\[ \overline{o}(s) = o(s) \]
\[ \overline{t}(\ast)(a) = \ast \]
\[ \overline{t}(s)(a) = t(s)(a) \]

How do we study PA wrt language equivalence?

\[ L_s = \llbracket i(s) \rrbracket \]
Example III: Linearization

\[ S \xrightarrow{\langle o, t \rangle} \mathbb{R} \times (\mathbb{R}_\omega^S)^A \]

How do we study WA wrt weighted languages (linear bisimilarity)?
Example III: Linearization

\[ S \xrightarrow{\langle o, t \rangle} \mathbb{R}^S \]

\[ \mathbb{R} \times (\mathbb{R}_\omega^S)^A \]

\[
o^\#(\begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix}) = \sum v_i \times o(s_i) \\
t^\#(\begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix})(a)(s_j) = \sum v_i \times t(s_i)(a)(s_j)
\]
Example III: Linearization

\[
\begin{array}{cccccc}
S & \xrightarrow{e} & \mathbb{R}S & \xrightarrow{\mathcal{R}} & \mathbb{R}A^* \\
\langle o, t \rangle & \downarrow & \langle o^\#, t^\# \rangle & & \\
\mathbb{R} \times (\mathbb{R}_\omega S)^A & \xrightarrow{\mathcal{R}} & \mathbb{R} \times (\mathbb{R}A^*)^A \\
\end{array}
\]

\[
o^\#( \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix} ) = \sum v_i \times o(s_i) \\
t^\#( \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix} )(a)(s_j) = \sum v_i \times t(s_i)(a)(s_j)
\]

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

\[
L_s = [ \begin{pmatrix} e(s) \end{pmatrix} ]
\]

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

$$L_s = \left\lceil e(s) \right\rceil$$
Chasing the pattern... 

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

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

Transform an $\mathcal{F}$-coalgebra $(X, f)$ into an $\mathcal{T}$-coalgebra $(\mathcal{T}(X), f^{\#})$.

If $\mathcal{F}$ has final coalgebra: $x_1 \approx \mathcal{T}\mathcal{F}x_2 \iff \left[ \eta_{\mathcal{X}}(x_1) \right] = \left[ \eta_{\mathcal{X}}(x_2) \right]$.
How do we capture all the examples (and more) in the same framework?

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

The diagram shows:

$\xymatrix{X \ar[r] \ar[d]_f & T(X) \\
T T(X) \ar@{.>}[u]}$
Chasing the pattern...

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

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

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

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

Transform an $FT$-coalgebra $(X, f)$ into an $F$-coalgebra $(T(X), f^\#)$.

If $F$ has final coalgebra: $x_1 \approx_T x_2 \iff [\eta_X(x_1)] = [\eta_X(x_2)]$. 

Alexandra Silva (CWI) 
Algebraically Enriched Coalgebras 
CMCS, March 2010
In a nutshell. . .

**Ingredients:**
- A monad $T$;
- A final coalgebra for $F$ (for instance, take $F$ to be bounded);
- An extension $f^#$ of $f$;
In a nutshell...

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

$$f^\#: T(X) \xrightarrow{T(f)} T(F(T(X))) \xrightarrow{h} F(T(X))$$
Bisimilarity implies $T$-enriched bisimilarity

Theorem

$\sim_{FT} \Rightarrow \approx^T_F$
Bisimilarity implies $T$-enriched bisimilarity

Theorem

$$\sim_{FT} \Rightarrow \approx^T_F$$

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 $(F(X) = 2 \times X^A, T = 1 + \text{–})$ 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 $S \rightarrow (B \times (1+S))^A$;
- Automata with exceptions $S \rightarrow 2 \times (E+S)^A$;
- Automata with side effects $S \rightarrow E^E \times ((E \times S)^E)^A$;
- Total subsequential transducers $S \rightarrow O^* \times (O^* \times S)^A$;
- Probabilistic automata $S \rightarrow [0, 1] \times (D_\omega(X))^A$;
- ...
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 ⊥,...); 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:

$$x \sim_{tr} y \iff x \approx^T_F y$$

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

$$x \sim_{tr} y \Rightarrow x \approx^T_F y$$