Modeling Automata with Coalgebras

TOC 2019

ERIC BOND

Motivation & Background

- What?
  - A general mathematical framework for modeling state transition systems of various types
    - Deterministic, Nondeterministic, Probabilistic, Quantum, …
- How?
  - By representing these systems with variations of the same mathematical structure and building a framework around that structure.
- Why?
  - Observe old results in a new light with a stronger toolkit.
  - Is this a useful encoding of systems for proof assistants?

High Level Idea

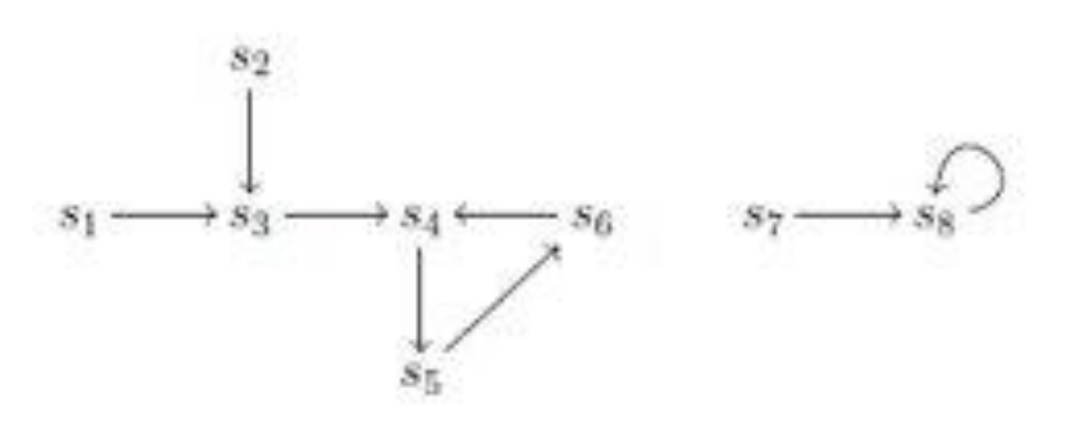
- Components
  - States
  - Observation function
  - Transition function
- Understand properties of systems by mapping(comparing) them to other systems
- Features
  - Mathematical definition of behavioral equivalence
  - Local behavior to global behavior
  - Behavioral equivalence of states in a system
  - Behavioral equivalence of states in different systems
  - Yields a proof technique, Coindution (dual to structural induction)
  - Automata minimization

Coalgebras with "state space" X are maps out of X, of the form:

$$X \xrightarrow[\text{modify/observe}]{} \boxed{X \quad \cdots \quad X}$$

$$\begin{tikzcd} S \arrow[d, "\alpha"] \\ A \end{tikzcd} \qquad \begin{tikzcd} S \arrow[d, "\beta"] \\ S \end{tikzcd} \qquad \begin{tikzcd} S \arrow[d, "\gamma"] \\ S^A \end{tikzcd} \qquad \begin{tikzcd} S \arrow[d, "\delta"] \\ \mathcal{P}(S) \end{tikzcd}$$


S
β
S


s_1
s_2
s_3
s_4
s_5
s_6
s_7
s_8

Definition 38 (homomorphism of dynamical systems).
A *homomorphism* $f\colon (S,\alpha) \to (T,\beta)$ of dynamical systems is a function $f\colon S \to T$ such that

$$\begin{tikzcd} S \arrow[r, "f"] \arrow[d, "\alpha"'] & T \arrow[d, "\beta"] \\ S \arrow[r, "f"'] & T \end{tikzcd}$$

that is, such that $\beta \circ f = f \circ \alpha$. $\square$

Equivalently, f is a homomorphism if and only if

$$\forall\, s \in S: \quad s \longrightarrow s' \implies f(s) \longrightarrow f(s')$$

Thus a homomorphism between two dynamical systems is a *function* between the underlying sets of states that *preserves* transitions. In other words, homomorphisms are (transition) *structure preserving* functions.

Using homomorphisms, one can for instance express the fact that *all* dynamical systems are equivalent, in the following sense. Consider the following dynamical system:

$$(\mathbf{1}, id) \qquad \mathbf{1} = \{*\} \qquad id\colon \mathbf{1} \to \mathbf{1} \qquad id(*) \;=\; *$$

There is the following trivial fact: there exists a unique homomorphism from any dynamical system (S, α) to $(\mathbf{1}, id)$:

$$\begin{tikzcd} S \arrow[r, "\exists!\, f"] \arrow[d, "\alpha"'] & \mathbf{1} \arrow[d, "id"] \\ S \arrow[r, "f"'] & \mathbf{1} \end{tikzcd} \qquad f(s) \;=\; * \qquad (s \in S)$$

$$\begin{tikzcd} S \arrow[d] \\ A \times S \end{tikzcd}$$

$$(S, \langle \mathsf{o}, \mathsf{tr} \rangle) \quad = \quad \begin{tikzcd} s_0|a \arrow[r] & s_1|a \arrow[r] & s_2|b \arrow[r, bend left] & s_3|a \arrow[l, bend left] \end{tikzcd}$$

$$\begin{tikzcd} S \arrow[d] \\ A \times S \end{tikzcd}$$

Definition 54 (homomorphism of stream systems).
A *homomorphism* $f\colon (S, \langle \mathsf{o}_S, \mathsf{tr}_S\rangle) \to (T, \langle \mathsf{o}_T, \mathsf{tr}_T\rangle)$ of stream systems is a function $f\colon S \to T$ such that

$$\begin{tikzcd} S \arrow[r, "f"] \arrow[d, "\langle \mathsf{o}_S, \mathsf{tr}_S\rangle"'] & T \arrow[d, "\langle \mathsf{o}_T, \mathsf{tr}_T\rangle"] \\ A \times S \arrow[r, "1_A \times f"'] & A \times T \end{tikzcd}$$

that is, such that $\mathsf{o}_T \circ f = \mathsf{o}_S$ and $\mathsf{tr}_T \circ f = f \circ \mathsf{tr}_S$. □

Homomorphisms are functions that preserve transitions and outputs:

$$s|a \longrightarrow t|b \quad \Longrightarrow \quad f(s)|a \longrightarrow f(t)|b \tag{7.1}$$

where we are using the following notation:

$$s|a \longrightarrow t|b \quad \Longleftrightarrow \quad \mathsf{o}_S(s) = a \;\text{ and }\; \mathsf{tr}_S(s) = t \;\text{ and }\; \mathsf{o}_S(t) = b$$

Definition 54 (homomorphism of stream systems).
A *homomorphism* $f\colon (S, \langle \mathsf{o}_S, \mathsf{tr}_S\rangle) \to (T, \langle \mathsf{o}_T, \mathsf{tr}_T\rangle)$ of stream systems is a function $f\colon S \to T$ such that

$$\begin{tikzcd} S \arrow[r, "f"] \arrow[d, "\langle \mathsf{o}_S,\mathsf{tr}_S\rangle"'] & T \arrow[d, "\langle \mathsf{o}_T,\mathsf{tr}_T\rangle"] \\ A \times S \arrow[r, "1_A \times f"'] & A \times T \end{tikzcd}$$

that is, such that $\mathsf{o}_T \circ f = \mathsf{o}_S$ and $\mathsf{tr}_T \circ f = f \circ \mathsf{tr}_S$. $\square$

Homomorphisms are functions that preserve transitions and outputs:

$$s|a \longrightarrow t|b \quad \Longrightarrow \quad f(s)|a \longrightarrow f(t)|b \tag{7.1}$$

where we are using the following notation:

$$s|a \longrightarrow t|b \quad \Longleftrightarrow \quad \mathsf{o}_S(s) = a \ \text{ and } \ \mathsf{tr}_S(s) = t \ \text{ and } \ \mathsf{o}_S(t) = b$$

$$\begin{tikzcd} S \arrow[d] \\ A \times S \end{tikzcd}$$

$$A^\omega \;=\; \{\sigma \mid \sigma\colon \mathbb{N} \to A\}$$

Theorem 78 (finality – streams). *The stream system* $(A^\omega, \langle \mathsf{i}_{st}, \mathsf{d}_{st}\rangle)$ *is* final*: For every stream system* $(S, \langle \mathsf{o}_S, \mathsf{tr}_S\rangle)$ *there exists a* unique *homomorphism*

$$\begin{tikzcd} S \arrow[r, dashed, "\exists!\, [\![-]\!]"] \arrow[d, "\forall\, \langle \mathsf{o}_S, \mathsf{tr}_S\rangle"'] & A^\omega \arrow[d, "\langle \mathsf{i}_{st}, \mathsf{d}_{st}\rangle"] \\ A \times S \arrow[r, dashed, "1_A \times [\![-]\!]"'] & A \times A^\omega \end{tikzcd}$$

$$
\begin{tikzcd}
S \arrow[d] \\
A \times S
\end{tikzcd}
$$

$$
\begin{tikzcd}
(S, \langle \mathsf{o}, \mathsf{tr} \rangle) \quad = \quad s_0|a \arrow[r] & s_1|a \arrow[r] & s_2|b \arrow[r, bend left] & s_3|a \arrow[l, bend left]
\end{tikzcd}
$$

then

$$
[\![s_0]\!] \;=\; aa(ba)^\omega \qquad [\![s_1]\!] \;=\; a(ba)^\omega \qquad [\![s_2]\!] \;=\; (ba)^\omega \qquad [\![s_3]\!] \;=\; (ab)^\omega
$$

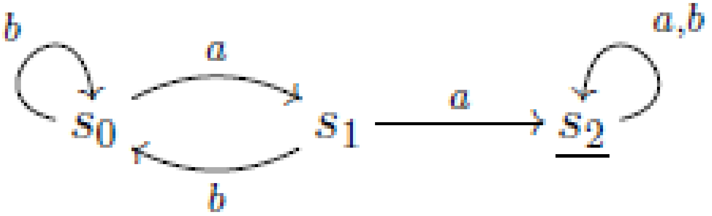
where $(ab)^\omega = (a, b, a, b, a, b, \ldots)$ and $(ba)^\omega = (b, a, b, a, b, a, \ldots)$.

$$\begin{array}{c} S \\ \big\downarrow \\ 2 \times S^A \end{array}$$

$$S^A \;=\; \{\, f \mid f\colon A \to S \,\}$$

Let $A = \{a, b\}$. Here is an example of an automaton:

$$(S, \langle \mathsf{o}, \mathsf{tr} \rangle) \;=\;$$



Theorem 153 (finality – languages). *For every automaton* $(S, \langle \mathsf{o}, \mathsf{tr} \rangle)$ *there exists a unique homomorphism* $[\![-]\!] \colon (S, \langle \mathsf{o}, \mathsf{tr} \rangle) \to (\mathcal{P}(A^*), \langle \mathsf{i}_l, \mathsf{d}_l \rangle)$*:*

$$\begin{tikzcd} S \arrow[r, dashed, "\exists!\, [\![-]\!]"] \arrow[d, "\forall\, \langle \mathsf{o}, \mathsf{tr} \rangle"'] & \mathcal{P}(A^*) \arrow[d, "\langle \mathsf{i}_l, \mathsf{d}_l \rangle"] \\ 2 \times S^A \arrow[r, dashed, "1 \times [\![-]\!]^A"'] & 2 \times \mathcal{P}(A^*)^A \end{tikzcd}$$

Proof: We define

$$[\![-]\!] \colon S \to \mathcal{P}(A^*) \qquad [\![s]\!] \;=\; l(s) \;=\; \{ w \in A^* \mid \mathsf{o}(s_w) = 1 \} \qquad (s \in S,\, w \in A^*)$$

One easily verifies that $[\![-]\!]$ is the only function making the diagram above commute. $\square$

The final homomorphism $[\![-]\!]$ assigns to every state s its *global* behaviour, consisting of the language $l(s)$ of all words accepted by s.

Coalgebras with "state space" X are maps out of X, of the form:

$$X \xrightarrow[\text{modify/observe}]{} \boxed{X \quad \cdots \quad X}$$

Definition 1.2. A *functor*

$$F : \mathbf{C} \to \mathbf{D}$$

between categories $\mathbf{C}$ and $\mathbf{D}$ is a mapping of objects to objects and arrows to arrows, in such a way that:

(a) $F(f : A \to B) = F(f) : F(A) \to F(B)$,

(b) $F(g \circ f) = F(g) \circ F(f)$,

(c) $F(1_A) = 1_{F(A)}$.

$$
\begin{array}{rrcll}
\mathsf{Id}\colon \mathsf{Set} \to \mathsf{Set} & \mathsf{Id}(S) & = & S & \text{(dynamical systems)} \\
\mathsf{Str}\colon \mathsf{Set} \to \mathsf{Set} & \mathsf{Str}(S) & = & A \times S & \text{(stream systems)} \\
\mathsf{dA}\colon \mathsf{Set} \to \mathsf{Set} & \mathsf{dA}(S) & = & 2 \times S^A & \text{(deterministic automata)} \\
\mathsf{pA}\colon \mathsf{Set} \to \mathsf{Set} & \mathsf{pA}(S) & = & 2 \times (1+S)^A & \text{(partial automata)} \\
\mathsf{ndA}\colon \mathsf{Set} \to \mathsf{Set} & \mathsf{ndA}(S) & = & 2 \times \mathcal{P}(S)^A & \text{(non-deterministic automata)} \\
\mathsf{MA}\colon \mathsf{Set} \to \mathsf{Set} & \mathsf{MA}(S) & = & (B \times S)^A & \text{(Mealy automata)} \\
\mathsf{MoA}\colon \mathsf{Set} \to \mathsf{Set} & \mathsf{MoA}(S) & = & B \times S^A & \text{(Moore automata)} \\
\mathsf{wsA}\colon \mathsf{Set} \to \mathsf{Set} & \mathsf{wsA}(S) & = & \mathbb{R} \times \mathbb{R}^S_\omega & \text{(weighted stream automata)}
\end{array}
$$

Coalg_F	F	name for $X \to FX$/reference
MC	$\mathcal{D}$	Markov chains
DLTS	$(\_ + 1)^A$	deterministic automata
LTS	$\mathcal{P}(A \times \_) \cong \mathcal{P}^A$	non-deterministic automata, LTSs
React	$(\mathcal{D} + 1)^A$	reactive systems [55,30]
Gen	$\mathcal{D}(A \times \_) + 1$	generative systems [30]
Str	$\mathcal{D} + (A \times \_) + 1$	stratified systems [30]
Alt	$\mathcal{D} + \mathcal{P}(A \times \_)$	alternating systems [33]
Var	$\mathcal{D}(A \times \_) + \mathcal{P}(A \times \_)$	Vardi systems [77]
SSeg	$\mathcal{P}(A \times \mathcal{D})$	simple Segala systems [67,66]
Seg	$\mathcal{P}\mathcal{D}(A \times \_)$	Segala systems [67,66]
Bun	$\mathcal{D}\mathcal{P}(A \times \_)$	bundle systems [22]
PZ	$\mathcal{P}\mathcal{D}\mathcal{P}(A \times \_)$	Pnueli–Zuck systems [62]
MG	$\mathcal{P}\mathcal{D}\mathcal{P}(A \times \_ + \_)$	most general systems

Sources

- Jacobs, Bart. "Introduction to coalgebra." *Towards mathematics of states and observations* (2005).
- Rutten, Jan. "The Method of Coalgebra: exercises in coinduction." (2019).
- Sokolova, Ana. "Probabilistic systems coalgebraically: A survey." *Theoretical Computer Science* 412.38 (2011): 5095-5110.
- Jacobs, Bart. "Bayesian Networks as Coalgebras." (2014)