Coalgebraic Representation of Büchi Automata

Research Internship Presentation

Jorrit de Boer

17 January 2025

Introduction

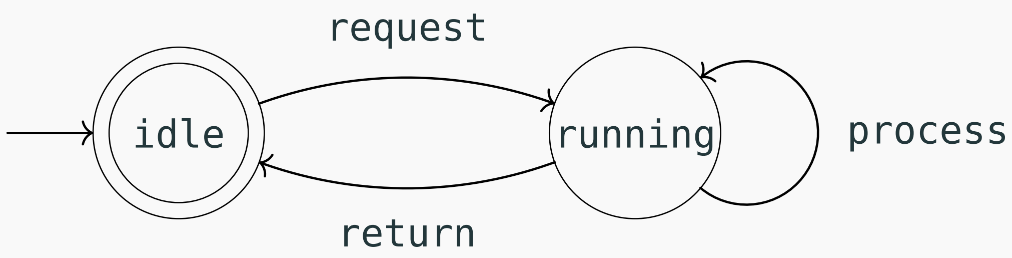
Introduction

Outline:

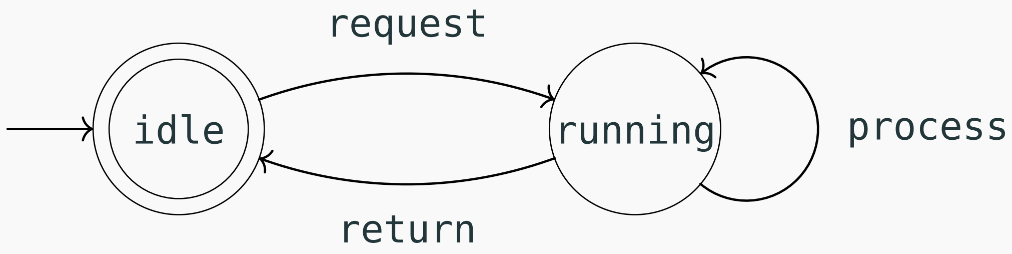
1. Büchi Automata
2. Coalgebra Determinstic Finite Automata
3. Coalgebra Nondeterministic Finite Automata
4. Coalgebra Possibly Infinite Behavior Nondeterministic Finite Automata
5. Coalgebra Büchi Automata
6. Outline Derivation using Game Semantics

Büchi Automata

Büchi Automata



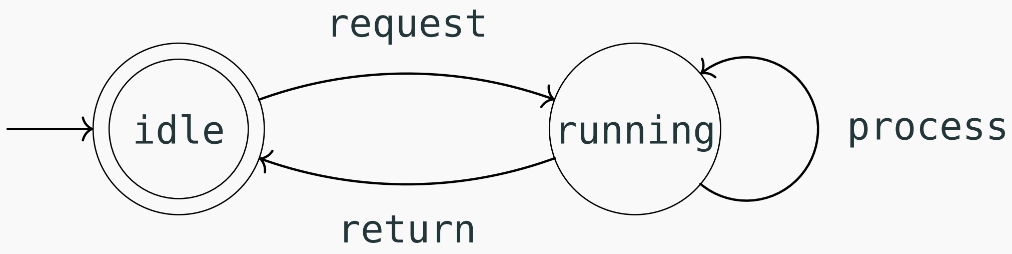
Büchi Automata



Definition: A (nondeterministic) *Büchi Automaton* $A = \langle S, \Sigma, \delta, s_0, F \rangle$, where:

- S: finite set of states
- Σ: alphabet
- $s_0 \in S$: initial state
- $\delta : S \times \Sigma \to \mathcal{P}(S)$: transition function
- $F \subseteq S$: set of *final* (or *accepting*) states.

Büchi Automata

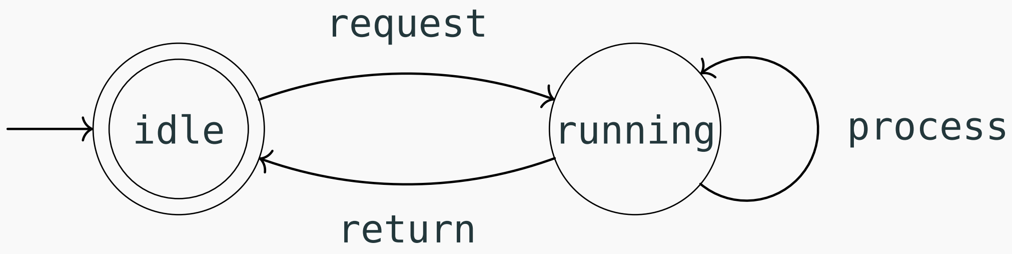


Definition: A (nondeterministic) *Büchi Automaton* $A = \langle S, \Sigma, \delta, s_0, F \rangle$, where:

- S: finite set of states
- Σ: alphabet
- $s_0 \in S$: initial state
- $\delta : S \times \Sigma \to \mathcal{P}(S)$: transition function
- $F \subseteq S$: set of *final* (or *accepting*) states.

A *run* of A on an ω-word $w = \sigma_0 \sigma_1 \ldots \in \Sigma^\omega$ is an infinite sequence of states $s_0, s_1, \ldots \in S^\omega$ such that for all n, $s_{n+1} \in \delta(s_n, \sigma_n)$

A run is *accepted* if it visits F infinitly often.



Definition: A (nondeterministic) *Büchi Automaton* $A = \langle S, \Sigma, \delta, s_0, F \rangle$, where:

- S: finite set of states
- Σ: alphabet
- $s_0 \in S$: initial state
- $\delta : S \times \Sigma \to \mathcal{P}(S)$: transition function
- $F \subseteq S$: set of *final* (or *accepting*) states.

A *run* of A on an ω-word $w = \sigma_0 \sigma_1 \ldots \in \Sigma^\omega$ is an infinite sequence of states $s_0, s_1, \ldots \in S^\omega$ such that for all n, $s_{n+1} \in \delta(s_n, \sigma_n)$

A run is *accepted* if it visits F infinitly often.

Accepted language:
$(\texttt{request} \cdot \texttt{process}^* \cdot \texttt{return})^\omega$

Coalgebra

Final Coalgebra Deterministic Finite Automata

$\langle S, \Sigma, \delta, o \rangle$ with states S, alphabet Σ, transition function $\delta : S \times \Sigma \to S$, $o : S \to 2$ ($2 = \{0, 1\}$). Can be represented by a coalgebra $\langle o, \delta \rangle : S \to 2 \times S^{\Sigma}$ for functor $FS = 2 \times S^{\Sigma}$

Final Coalgebra Deterministic Finite Automata

$\langle S, \Sigma, \delta, o \rangle$ with states S, alphabet Σ, transition function $\delta : S \times \Sigma \to S$, $o : S \to 2$ ($2 = \{0, 1\}$). Can be represented by a coalgebra $\langle o, \delta \rangle : S \to 2 \times S^{\Sigma}$ for functor $FS = 2 \times S^{\Sigma}$

The final coalgebra for F is $\langle e, d \rangle : 2^{\Sigma^*} \to 2 \times \left(2^{\Sigma^*}\right)^{\Sigma}$. Where

- $e(L) = 1$ iff $\varepsilon \in L$
- $d(L)(a) = L_a$ where $L_a(w) = L(aw)$ so $w \in d(L)(a)$ iff $aw \in L$

Final Coalgebra Deterministic Finite Automata

$\langle S, \Sigma, \delta, o \rangle$ with states S, alphabet Σ, transition function $\delta : S \times \Sigma \to S$, $o : S \to 2$ ($2 = \{0, 1\}$). Can be represented by a coalgebra $\langle o, \delta \rangle : S \to 2 \times S^{\Sigma}$ for functor $FS = 2 \times S^{\Sigma}$

The final coalgebra for F is $\langle e, d \rangle : 2^{\Sigma^*} \to 2 \times \left(2^{\Sigma^*}\right)^{\Sigma}$. Where

- $e(L) = 1$ iff $\varepsilon \in L$
- $d(L)(a) = L_a$ where $L_a(w) = L(aw)$ so $w \in d(L)(a)$ iff $aw \in L$

$$\begin{tikzcd}
2 \times S^{\Sigma} \arrow[r, dashed, "\mathrm{id} \times \mathsf{beh}^{\Sigma}"] & 2 \times \left(2^{\Sigma^*}\right)^{\Sigma} \\
S \arrow[u, "\langle o, \delta \rangle"] \arrow[r, dashed, "\mathsf{beh}"'] & 2^{\Sigma^*} \arrow[u, "\langle e, d \rangle"']
\end{tikzcd}$$

Final Coalgebra Deterministic Finite Automata

$\langle S, \Sigma, \delta, o \rangle$ with states S, alphabet Σ, transition function $\delta : S \times \Sigma \to S$, $o : S \to 2$ ($2 = \{0, 1\}$). Can be represented by a coalgebra $\langle o, \delta \rangle : S \to 2 \times S^\Sigma$ for functor $FS = 2 \times S^\Sigma$

The final coalgebra for F is $\langle e, d \rangle : 2^{\Sigma^*} \to 2 \times \left(2^{\Sigma^*}\right)^\Sigma$. Where

- $e(L) = 1$ iff $\varepsilon \in L$
- $d(L)(a) = L_a$ where $L_a(w) = L(aw)$ so $w \in d(L)(a)$ iff $aw \in L$

$$\begin{tikzcd}
2 \times S^\Sigma \arrow[r, dashed, "\mathrm{id} \times \mathsf{beh}^\Sigma"] & 2 \times \left(2^{\Sigma^*}\right)^\Sigma \\
S \arrow[u, "\langle o, \delta \rangle"] \arrow[r, dashed, "\mathsf{beh}"'] & 2^{\Sigma^*} \arrow[u, "\langle e, d \rangle"']
\end{tikzcd}$$

Following the paths through the diagram we obtain:

- $\mathsf{beh}(s)(\varepsilon) = o(s)$, and
- $\mathsf{beh}(s)(\sigma w) = \mathsf{beh}(\delta(s)(\sigma))(w)$,

Final Coalgebra Deterministic Finite Automata

$\langle S, \Sigma, \delta, o \rangle$ with states S, alphabet Σ, transition function $\delta : S \times \Sigma \to S$, $o : S \to 2$ ($2 = \{0, 1\}$). Can be represented by a coalgebra $\langle o, \delta \rangle : S \to 2 \times S^{\Sigma}$ for functor $FS = 2 \times S^{\Sigma}$

The final coalgebra for F is $\langle e, d \rangle : 2^{\Sigma^*} \to 2 \times \left(2^{\Sigma^*}\right)^{\Sigma}$. Where

- $e(L) = 1$ iff $\varepsilon \in L$
- $d(L)(a) = L_a$ where $L_a(w) = L(aw)$ so $w \in d(L)(a)$ iff $aw \in L$

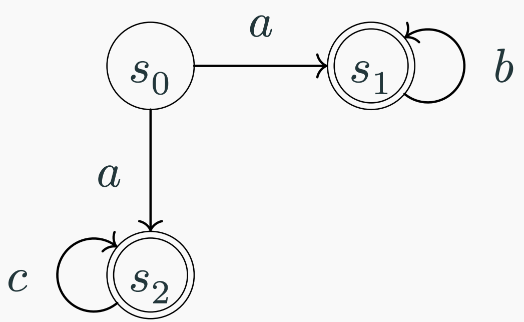
$$\begin{tikzcd}
2 \times S^{\Sigma} \arrow[r, dashed, "\mathrm{id} \times \mathsf{beh}^{\Sigma}"] & 2 \times \left(2^{\Sigma^*}\right)^{\Sigma} \\
S \arrow[u, "\langle o, \delta \rangle"] \arrow[r, dashed, "\mathsf{beh}"'] & 2^{\Sigma^*} \arrow[u, "\langle e, d \rangle"']
\end{tikzcd}$$

Following the paths through the diagram we obtain:

- $\mathsf{beh}(s)(\varepsilon) = o(s)$, and
- $\mathsf{beh}(s)(\sigma w) = \mathsf{beh}(\delta(s)(\sigma))(w)$,

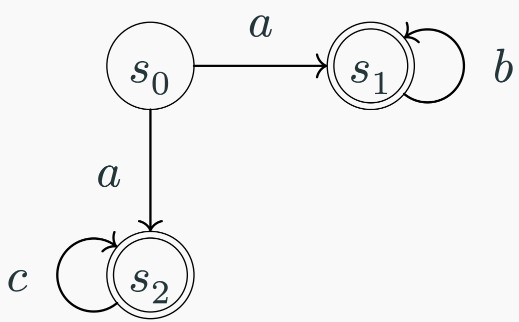
So beh captures exactly the accepted language of the automaton!

Nondeterministic Finite Automata



Might be modeled by coalgebra $c : S \to 2 \times \mathcal{P}(\Sigma \times S)$.

Nondeterministic Finite Automata



Might be modeled by coalgebra $c : S \to 2 \times \mathcal{P}(\Sigma \times S)$.

A final coalgebra $z : Z \to 2 \times \mathcal{P}(\Sigma \times Z)$ cannot exist. Lambek's lemma says z would have to be an isomorphism, which would imply $Z \cong \mathcal{P}(Z)$

Solution by Hasuo, Jacobs, Sokolova 2007

Kleisli Category of the monad $\mathcal{P}$:

A coalgebra $c : S \to \Sigma \times S$ in $\mathcal{K}\ell(\mathcal{P})$ is $c : S \to \mathcal{P}(\Sigma \times S)$ in **Sets**.

Solution by Hasuo, Jacobs, Sokolova 2007

Kleisli Category of the monad $\mathcal{P}$:

A coalgebra $c : S \to \Sigma \times S$ in $\mathcal{K}\ell(\mathcal{P})$ is $c : S \to \mathcal{P}(\Sigma \times S)$ in **Sets**. Concretely:

- $\eta_X : X \to \mathcal{P}(X)$: $\eta_X(x) = \{x\}$
- $\mu_X : \mathcal{P}(\mathcal{P}(X)) \to \mathcal{P}(X)$: $\mu_X(A) = \bigcup_{a \in A} a$.

For $f : X \to Y$, $\mathcal{P}(f) : \mathcal{P}(X) \to \mathcal{P}(Y)$ by $\mathcal{P}(f)(A) = \{f(a) \mid a \in A\}$

- **objects**: the same as for **Sets**, sets
- **morphisms**: $f : X \to Y$ in $\mathcal{K}\ell(\mathcal{P})$ is $f : X \to \mathcal{P}(Y)$ in **Sets**.
  For morphisms $f : X \to Y$ and $g : Y \to Z$ in $\mathcal{K}\ell(\mathcal{P})$ we define

$$g \odot f := X \xrightarrow{f} \mathcal{P}(Y) \xrightarrow{\mathcal{P}(g)} \mathcal{P}(\mathcal{P}(Z)) \xrightarrow{\mu_Y} \mathcal{P}(Z)$$

Lifted Functor in Kleisli Category

Model NFA $\langle S, \Sigma, \delta, o \rangle$ by coalgebra $c : S \to 1 + \Sigma \times S$ for the functor $FS = 1 + \Sigma \times S$, which is $c : S \to \mathcal{P}(1 + \Sigma \times S)$ in **Sets**.

Lifted Functor in Kleisli Category

Model NFA $\langle S, \Sigma, \delta, o\rangle$ by coalgebra $c : S \to 1 + \Sigma \times S$ for the functor $FS = 1 + \Sigma \times S$, which is $c : S \to \mathcal{P}(1 + \Sigma \times S)$ in **Sets**.

Problem: a map $f : X \to Y$ in $\mathcal{K}\ell(\mathcal{P})$ is $f : X \to \mathcal{P}(Y)$ in **Sets** so $Ff : FX \to F\mathcal{P}(Y)$

Model NFA $\langle S, \Sigma, \delta, o\rangle$ by coalgebra $c : S \to 1 + \Sigma \times S$ for the functor $FS = 1 + \Sigma \times S$, which is $c : S \to \mathcal{P}(1 + \Sigma \times S)$ in **Sets**.

Problem: a map $f : X \to Y$ in $\mathcal{K}\ell(\mathcal{P})$ is $f : X \to \mathcal{P}(Y)$ in **Sets** so $Ff : FX \to F\mathcal{P}(Y)$

We need a natural transformation $\lambda : F\mathcal{P} \Rightarrow \mathcal{P}F$ (distributive law):

$$1 + \Sigma \times (\mathcal{P}(S)) \xrightarrow{\lambda} \mathcal{P}(1 + \Sigma \times X)$$

Lifted Functor in Kleisli Category

Model NFA $\langle S, \Sigma, \delta, o\rangle$ by coalgebra $c : S \to 1 + \Sigma \times S$ for the functor $FS = 1 + \Sigma \times S$, which is $c : S \to \mathcal{P}(1 + \Sigma \times S)$ in **Sets**.

Problem: a map $f : X \to Y$ in $\mathcal{K}\ell(\mathcal{P})$ is $f : X \to \mathcal{P}(Y)$ in **Sets** so $Ff : FX \to F\mathcal{P}(Y)$

We need a natural transformation $\lambda : F\mathcal{P} \Rightarrow \mathcal{P}F$ (distributive law):

$$1 + \Sigma \times (\mathcal{P}(S)) \xrightarrow{\lambda} \mathcal{P}(1 + \Sigma \times X)$$

- $* \mapsto \{*\}(1 = \{*\})$
- $(\sigma, S) = \{(\sigma, x) | x \in S\}$ for $\sigma \in \Sigma$ and $S \subseteq X$.

For example: $\delta(s)(\sigma) = \{x, y, z\}$ then $(\lambda \circ c)(s) = \{(\sigma, x), (\sigma, y), (\sigma, z)\}$.

Lifted Functor in Kleisli Category

Model NFA $\langle S, \Sigma, \delta, o \rangle$ by coalgebra $c : S \to 1 + \Sigma \times S$ for the functor $FS = 1 + \Sigma \times S$, which is $c : S \to \mathcal{P}(1 + \Sigma \times S)$ in **Sets**.

Problem: a map $f : X \to Y$ in $\mathcal{K}\ell(\mathcal{P})$ is $f : X \to \mathcal{P}(Y)$ in **Sets** so $Ff : FX \to F\mathcal{P}(Y)$

We need a natural transformation $\lambda : F\mathcal{P} \Rightarrow \mathcal{P}F$ (distributive law):

$$1 + \Sigma \times (\mathcal{P}(S)) \xrightarrow{\lambda} \mathcal{P}(1 + \Sigma \times X)$$

- $* \mapsto \{*\}(1 = \{*\})$
- $(\sigma, S) = \{(\sigma, x) | x \in S\}$ for $\sigma \in \Sigma$ and $S \subseteq X$.

For example: $\delta(s)(\sigma) = \{x, y, z\}$ then $(\lambda \circ c)(s) = \{(\sigma, x), (\sigma, y), (\sigma, z)\}$.

Call $\overline{F}S = FS$ and $\overline{F}f = \lambda \circ \mathcal{P}(f)$ the *lifted functor*

Initial Algebra ⇒ Final Coalgebra

Theorem [Hasuo, Jacobs, Sokolova 2007]: An initial algebra $\alpha : FA \to A$ for the functor F in **Sets** yields the final coalgebra for $\overline{F}$ in $\mathcal{K}\ell(\mathcal{P})$:

$$\left(\eta_{FA} \circ \alpha^{-1}\right) : A \to \overline{F}A \; in \; \mathcal{K}\ell(\mathcal{P})$$

Theorem [Hasuo, Jacobs, Sokolova 2007]: An initial algebra $\alpha : FA \to A$ for the functor F in **Sets** yields the final coalgebra for $\overline{F}$ in $\mathcal{K}\ell(\mathcal{P})$:

$$(\eta_{FA} \circ \alpha^{-1}) : A \to \overline{F}A \; in \; \mathcal{K}\ell(\mathcal{P})$$

The initial algebra for $FS = 1 + \Sigma \times S$ is $[\mathsf{nil}, \mathsf{cons}] : 1 + \Sigma \times \Sigma^* \to \Sigma^*$:

- $\mathsf{nil}(*) = \varepsilon$
- $\mathsf{cons}(\sigma, w) = \sigma w$

so we get $(\eta_{1+\Sigma\times S} \circ [\mathsf{nil}, \mathsf{cons}]^{-1}) : \Sigma^* \to 1 + \Sigma \times \Sigma^*$ ($\Sigma^* \to \mathcal{P}(1 + \Sigma \times \Sigma^*)$ in **Sets**)

- $(\eta_{1+\Sigma\times S} \circ [\mathsf{nil}, \mathsf{cons}]^{-1})(\varepsilon) = \{*\}$
- $(\eta_{1+\Sigma\times S} \circ [\mathsf{nil}, \mathsf{cons}]^{-1})(\sigma w) = \{(\sigma, w)\}$

Final Coalgebra Nondeterministic Automaton

$$
\begin{tikzcd}
1 + \Sigma \times S \arrow[r, dashed, "1 + \Sigma \times \mathsf{beh}"] & 1 + \Sigma \times \Sigma^* \\
S \arrow[u, "c"] \arrow[r, dashed, "\mathsf{beh}"'] & \Sigma^* \arrow[u, "\eta_{1+\Sigma\times\Sigma^*} \circ [\mathsf{nil}, \mathsf{cons}]^{-1} \ in\ \mathcal{K}\ell(\mathcal{P})."']
\end{tikzcd}
$$

Final Coalgebra Nondeterministic Automaton

$$
\begin{tikzcd}
1 + \Sigma \times S \arrow[r, dashed, "1 + \Sigma \times \mathsf{beh}"] & 1 + \Sigma \times \Sigma^* \\
S \arrow[u, "c"] \arrow[r, dashed, "\mathsf{beh}"'] & \Sigma^* \arrow[u, "\eta_{1+\Sigma\times\Sigma^*} \circ [\mathsf{nil}, \mathsf{cons}]^{-1}"']
\end{tikzcd}
\quad \textit{in } \mathcal{K\ell}(\mathcal{P}).
$$

$$\varepsilon \in \mathsf{beh}(s) \iff * \in c(s) \iff \text{state } s \text{ is accepting}$$

$$\sigma w \in \mathsf{beh}(s) \iff (\sigma, w) \in ((\Sigma \times \mathsf{beh}) \circ c)(s) = \{(\sigma, \mathsf{beh}(t)) \mid (\sigma, t) \in c(s)\} \iff \exists t.(t \in \delta(s)(\sigma) \wedge w \in \mathsf{beh}(t)).$$

Possibly Infinite Behavior

Theorem [Jacobs 2004]: A final coalgebra $\xi : Z \to FZ$ yields a *weakly final* coalgebra

$$(\eta_{FZ} \circ \xi) : Z \to \overline{F}(Z) \ in \ \mathcal{K}\ell(\mathcal{P})$$

Possibly Infinite Behavior

Theorem [Jacobs 2004]: A final coalgebra $\xi : Z \to FZ$ yields a *weakly final* coalgebra

$$(\eta_{FZ} \circ \xi) : Z \to \overline{F}(Z) \textit{ in } \mathcal{K}\ell(\mathcal{P})$$

$$\begin{tikzcd}
\overline{F}S \arrow[r, rightsquigarrow, "\overline{F}(\mathsf{beh})"] & \overline{F}Z \\
S \arrow[u, "c"] \arrow[r, rightsquigarrow, "\mathsf{beh}"'] & Z \arrow[u, "\eta_{FZ} \circ \xi"']
\end{tikzcd} \quad \textit{in } \mathcal{K}\ell(\mathcal{P}),$$

beh is not unique. However, we can take beh^∞, the maximal mapping with respect to inclusion.

$\xi : \Sigma^\infty \to 1 + \Sigma \times \Sigma^\infty$ is the final F-coalgebra, defined by $\xi(\varepsilon) = * \in 1$ and $\xi(\sigma w) = (\sigma, w)$ ($\Sigma^\infty = \Sigma^* \cup \Sigma^\omega$).

$\xi : \Sigma^\infty \to 1 + \Sigma \times \Sigma^\infty$ is the final F-coalgebra, defined by $\xi(\varepsilon) = * \in 1$ and $\xi(\sigma w) = (\sigma, w)$ $(\Sigma^\infty = \Sigma^* \cup \Sigma^\omega)$.

$$\begin{tikzcd}
1 + \Sigma \times S \arrow[r, rightsquigarrow, "1 + \Sigma \times \mathsf{beh}_c^\infty"] & 1 + \Sigma \times \Sigma^\infty \\
S \arrow[u, "c"] \arrow[r, rightsquigarrow, "\mathsf{beh}_c^\infty"'] & \Sigma^\infty \arrow[u, "J\xi"', "\cong"]
\end{tikzcd} \qquad \textit{in } \mathcal{K}\ell(\mathcal{P}).$$

$$\varepsilon \in \mathsf{beh}^\infty(s) \iff * \in c(s) \iff \text{state } s \text{ is accepting}$$

$$\sigma w \in \mathsf{beh}^\infty(s) \iff \exists t. \Big(s \xrightarrow{\sigma} t \wedge w \in \mathsf{beh}^\infty(t) \Big).$$

Idea: split $S = S_1 \cup S_2$ for S_1 non-accepting and S_2 accepting

Idea: split $S = S_1 \cup S_2$ for S_1 non-accepting and S_2 accepting

$$
\begin{tikzcd}
\Sigma \times S \arrow[r, squiggly, "\Sigma \times {[\mathsf{beh}_1, \mathsf{beh}_2]}"] & \Sigma \times \Sigma^\omega \\
S_1 \arrow[u, "c_1"] \arrow[r, squiggly, "\mathsf{beh}_1"'] \arrow[ur, phantom, "="'{below}, "\mu"] & \Sigma^\omega \arrow[u, "\eta_{\Sigma^\omega} \circ d"']
\end{tikzcd}
\qquad
\begin{tikzcd}
\Sigma \times S \arrow[r, squiggly, "\Sigma \times {[\mathsf{beh}_1, \mathsf{beh}_2]}"] & \Sigma \times \Sigma^\omega \\
S_2 \arrow[u, "c_2"] \arrow[r, squiggly, "\mathsf{beh}_2"'] \arrow[ur, phantom, "\underset{\nu}{=}"] & \Sigma^\omega \arrow[u, "\eta_{\Sigma^\omega} \circ d"']
\end{tikzcd}
\qquad \textit{in } \mathcal{K}\ell(\mathcal{P}).
$$

Idea: split $S = S_1 \cup S_2$ for S_1 non-accepting and S_2 accepting

$$\begin{tikzcd} \Sigma \times S \arrow[r, rightsquigarrow, "\Sigma \times [\mathsf{beh}_1, \mathsf{beh}_2]"] & \Sigma \times \Sigma^\omega \\ S_1 \arrow[u, "c_1"] \arrow[r, rightsquigarrow, "\mathsf{beh}_1"'] \arrow[ur, phantom, "\overline{\overline{\mu}}"] & \Sigma^\omega \arrow[u, "\eta_{\Sigma^\omega} \circ d"'] \end{tikzcd} \qquad \begin{tikzcd} \Sigma \times S \arrow[r, rightsquigarrow, "\Sigma \times [\mathsf{beh}_1, \mathsf{beh}_2]"] & \Sigma \times \Sigma^\omega \\ S_2 \arrow[u, "c_2"] \arrow[r, rightsquigarrow, "\mathsf{beh}_2"'] \arrow[ur, phantom, "\overline{\overline{\nu}}"] & \Sigma^\omega \arrow[u, "\eta_{\Sigma^\omega} \circ d"'] \end{tikzcd} \qquad \text{in } \mathcal{K}\ell(\mathcal{P}).$$

$$\mathsf{beh}_1 \stackrel{\mu}{=} (\eta_{\Sigma^\omega} \circ d)^{-1} \odot \overline{F}[\mathsf{beh}_1, \mathsf{beh}_2] \odot c_1$$

$$\mathsf{beh}_2 \stackrel{\nu}{=} (\eta_{\Sigma^\omega} \circ d)^{-1} \odot \overline{F}[\mathsf{beh}_1, \mathsf{beh}_2] \odot c_2$$

$$\mathsf{beh}_1 \stackrel{\mu}{=} (\eta_{\Sigma^\omega} \circ d)^{-1} \odot \overline{F}[\mathsf{beh}_1, \mathsf{beh}_2] \odot c_1 \qquad \mathsf{beh}_2 \stackrel{\nu}{=} (\eta_{\Sigma^\omega} \circ d)^{-1} \odot \overline{F}[\mathsf{beh}_1, \mathsf{beh}_2] \odot c_2$$

$$\mathsf{beh}_1 \overset{\mu}{=} (\eta_{\Sigma^\omega} \circ d)^{-1} \odot \overline{F}[\mathsf{beh}_1, \mathsf{beh}_2] \odot c_1 \qquad \mathsf{beh}_2 \overset{\nu}{=} (\eta_{\Sigma^\omega} \circ d)^{-1} \odot \overline{F}[\mathsf{beh}_1, \mathsf{beh}_2] \odot c_2$$

Rewrite to:

$$\mathsf{beh}_1 \overset{\mu}{=} \Diamond_\delta([\mathsf{beh}_1, \mathsf{beh}_2]) \upharpoonright S_1 \qquad \mathsf{beh}_2 \overset{\nu}{=} \Diamond_\delta([\mathsf{beh}_1, \mathsf{beh}_2]) \upharpoonright S_2$$

Where $\Diamond_\delta : (\mathcal{P}(\Sigma^\omega))^S \to (\mathcal{P}(\Sigma^\omega))^S$ is given by

$$\Diamond_\delta(\mathsf{beh})(s) = \{\sigma \cdot w \mid s' \in \delta(s)(\sigma), w \in \mathsf{beh}(s')\}.$$

$$\mathsf{beh}_1 \stackrel{\mu}{=} \Diamond_\delta([\mathsf{beh}_1, \mathsf{beh}_2]) \upharpoonright S_1 \qquad \mathsf{beh}_2 \stackrel{\nu}{=} \Diamond_\delta([\mathsf{beh}_1, \mathsf{beh}_2]) \upharpoonright S_2$$

Where $\Diamond_\delta : (\mathcal{P}(\Sigma^\omega))^S \to (\mathcal{P}(\Sigma^\omega))^S$ is given by

$$\Diamond_\delta(\mathsf{beh})(s) = \{\sigma \cdot w \mid s' \in \delta(s)(\sigma), w \in \mathsf{beh}(s')\}.$$

Definition: The *solution* to this *equational system* is calculated as follows:

- Intermediate solution $l_1^{(1)} := \mu u_1.f_1(u_1, u_2)$
- $l^{\mathrm{sol}} := \nu u_2.f_2\left(l_1^{(1)}(u_2), u_2\right)$
- $l_1^{\mathrm{sol}} = l_1^{(1)}\left(l_2^{\mathrm{sol}}\right)$

$$\mathsf{beh}_1 \overset{\mu}{=} \Diamond_\delta([\mathsf{beh}_1, \mathsf{beh}_2]) \upharpoonright S_1 \qquad \mathsf{beh}_2 \overset{\nu}{=} \Diamond_\delta([\mathsf{beh}_1, \mathsf{beh}_2]) \upharpoonright S_2$$

Where $\Diamond_\delta : (\mathcal{P}(\Sigma^\omega))^S \to (\mathcal{P}(\Sigma^\omega))^S$ is given by

$$\Diamond_\delta(\mathsf{beh})(s) = \{\sigma \cdot w \mid s' \in \delta(s)(\sigma), w \in \mathsf{beh}(s')\}.$$

Definition: The *solution* to this *equational system* is calculated as follows:

- Intermediate solution $l_1^{(1)} := \mu u_1.f_1(u_1, u_2)$
- $l^{\mathrm{sol}} := \nu u_2.f_2\left(l_1^{(1)}(u_2), u_2\right)$
- $l_1^{\mathrm{sol}} = l_1^{(1)}\left(l_2^{\mathrm{sol}}\right)$

Concretely:

- $l_1^{(1)} := \mu u_1.\Diamond_\delta([u_1, u_2]) \upharpoonright S_1$
- $l_2^{\mathrm{sol}} := \nu u_2.u_2\Diamond_\delta([\mu u_1.\Diamond_\delta([u_1, u_2]) \upharpoonright S_1, u_2]) \upharpoonright S_2, u_2)$
- $l_1^{\mathrm{sol}} = \mu u_1.\Diamond_\delta\big(\big[u_1, \nu u_2.u_2 \overset{\nu}{=} \Diamond_\delta([\mu u_1'.\Diamond_\delta([u_1', u_2]) \upharpoonright S_1, u_2]) \upharpoonright S_2, u_2\big)\big]\big) \upharpoonright S_1$

Let $A = \langle S, \Sigma, \delta, s_0, F \rangle$ be a Büchi automaton. Take $S_1 = S \setminus F$, $S_2 = F$. Model δ by coalgebras $c_1 : S_1 \to \mathcal{P}(\Sigma \times S)$, $c_2 : S_2 \to \mathcal{P}(\Sigma \times S)$. Take the initial algebra $d : \Sigma^\omega \to \Sigma \times \Sigma^\omega$ defined by $d(\sigma w) = (\sigma, w)$ in **Sets**.

$$\begin{tikzcd} \Sigma \times S \arrow[r, squiggly, "\Sigma \times [\mathsf{beh}_1, \mathsf{beh}_2]"] & \Sigma \times \Sigma^\omega \\ S_1 \arrow[u, "c_1"] \arrow[r, squiggly, "\mathsf{beh}_1"'] \arrow[ur, phantom, "=_\mu"] & \Sigma^\omega \arrow[u, "\eta_{\Sigma^\omega} \circ d"'] \end{tikzcd} \qquad \begin{tikzcd} \Sigma \times S \arrow[r, squiggly, "\Sigma \times [\mathsf{beh}_1, \mathsf{beh}_2]"] & \Sigma \times \Sigma^\omega \\ S_2 \arrow[u, "c_2"] \arrow[r, squiggly, "\mathsf{beh}_2"'] \arrow[ur, phantom, "=_\nu"] & \Sigma^\omega \arrow[u, "\eta_{\Sigma^\omega} \circ d"'] \end{tikzcd} \qquad \textit{in } \mathcal{K}\ell(\mathcal{P}).$$

$$\mathsf{beh}_1 \overset{\mu}{=} \Diamond_\delta([\mathsf{beh}_1, \mathsf{beh}_2]) \restriction S_1 \qquad \mathsf{beh}_2 \overset{\nu}{=} \Diamond_\delta([\mathsf{beh}_1, \mathsf{beh}_2]) \restriction S_2$$

Where $\Diamond_\delta : (\mathcal{P}(\Sigma^\omega))^S \to (\mathcal{P}(\Sigma^\omega))^S$ is given by

$$\Diamond_\delta(\mathsf{beh})(s) = \{\sigma \cdot w \mid s' \in \delta(s)(\sigma), w \in \mathsf{beh}(s')\}.$$

Let $A = \langle S, \Sigma, \delta, s_0, F \rangle$ be a Büchi automaton. Take $S_1 = S \setminus F$, $S_2 = F$. Model δ by coalgebras $c_1 : S_1 \to \mathcal{P}(\Sigma \times S)$, $c_2 : S_2 \to \mathcal{P}(\Sigma \times S)$. Take the initial algebra $d : \Sigma^\omega \to \Sigma \times \Sigma^\omega$ defined by $d(\sigma w) = (\sigma, w)$ in **Sets**.

$$
\begin{tikzcd}
\Sigma \times S \arrow[r, squiggly, "\Sigma \times [\mathsf{beh}_1, \mathsf{beh}_2]"] & \Sigma \times \Sigma^\omega \\
S_1 \arrow[u, "c_1"] \arrow[r, squiggly, "\mathsf{beh}_1"'] \arrow[ur, phantom, "=_\mu"] & \Sigma^\omega \arrow[u, "\eta_{\Sigma^\omega} \circ d"']
\end{tikzcd}
\qquad
\begin{tikzcd}
\Sigma \times S \arrow[r, squiggly, "\Sigma \times [\mathsf{beh}_1, \mathsf{beh}_2]"] & \Sigma \times \Sigma^\omega \\
S_2 \arrow[u, "c_2"] \arrow[r, squiggly, "\mathsf{beh}_2"'] \arrow[ur, phantom, "=_\nu"] & \Sigma^\omega \arrow[u, "\eta_{\Sigma^\omega} \circ d"']
\end{tikzcd}
\qquad \textit{in } \mathcal{K}\ell(\mathcal{P}).
$$

$$\mathsf{beh}_1 \overset{\mu}{=} \Diamond_\delta([\mathsf{beh}_1, \mathsf{beh}_2]) \upharpoonright S_1 \qquad \mathsf{beh}_2 \overset{\nu}{=} \Diamond_\delta([\mathsf{beh}_1, \mathsf{beh}_2]) \upharpoonright S_2$$

Where $\Diamond_\delta : (\mathcal{P}(\Sigma^\omega))^S \to (\mathcal{P}(\Sigma^\omega))^S$ is given by

$$\Diamond_\delta(\mathsf{beh})(s) = \{\sigma \cdot w \mid s' \in \delta(s)(\sigma), w \in \mathsf{beh}(s')\}.$$

Theorem [Urabe, Shimizu, Hasuo 2016]: The solutions $\mathsf{beh}_1, \mathsf{beh}_2$ to the system of equations coincide with the accepted language of the Büchi Automaton A.

Alternate Proof of Coincidence Result

Problem: system of fixed point equations is convoluted.

Alternate Proof of Coincidence Result

Problem: system of fixed point equations is convoluted.

Alternate derivation using game semantics:
Game Semantics Theorem: $s \vDash^T \varphi \iff$ verifier has a winning strategy in $\mathcal{G}(\varphi, T)$

Outline:
- Convert system of equations to modal mu-calculus formula
- Apply game semantics theorem
- Prove: V has a winning strategy in $\mathcal{G}(\varphi, T)$ from state $(x_i, w) \iff w \in \mathsf{beh}(s_i)$

Proof Outline

Converting formula:

$$l_{\mathrm{sol}}^{2} = \nu u_2 . \Diamond_{\delta}^{2} \left[\left(\mu u_1' . \Diamond_{\delta}^{1} [u_1', u_2] \right), u_2 \right]$$

$$\overline{\varphi_2} = \nu u_2 . (p_2 \wedge \Diamond((\mu u_1' . p_1 \wedge \Diamond((p_1 \wedge u_1') \vee (p_2 \wedge u_2))) \vee (p_2 \wedge u_2))))$$

Proof Outline

Converting formula:

$$l_{\text{sol}}^{2} = \nu u_2.\Diamond_{\delta}^{2}\left[\left(\mu u_1'.\Diamond_{\delta}^{1}[u_1', u_2]\right), u_2\right]$$

$$\overline{\varphi_2} = \nu u_2.(p_2 \wedge \Diamond((\mu u_1'.p_1 \wedge \Diamond((p_1 \wedge u_1') \vee (p_2 \wedge u_2))) \vee (p_2 \wedge u_2))))$$

Defining Transition System for Büchi Automaton A:

Let $A = (S_1 \cup S_2, \Sigma, \delta)$ be a Büchi automaton. Let Transition System (TS) over the set of propositional variables $\{p_1, p_2\}$, denoted as T_A, as follows:

- States: (s, w) for $s \in S$ and $w \in \Sigma^\omega$
- Transition $(s, \sigma w) \to (s', w)$ for $s, s' \in S$, $\sigma \in \Sigma$, $w \in \Sigma^\omega$, iff $s' \in \delta(s)(\sigma)$
- Labeling function: $\lambda((s, w)) = \{p_i\}$ iff $s \in S_i$

Proof Outline

Converting formula:

$$l_{\mathrm{sol}}^{2} = \nu u_2 . \Diamond_{\delta}^{2} \left[\left(\mu u_1' . \Diamond_{\delta}^{1} [u_1', u_2] \right), u_2 \right]$$

$$\overline{\varphi_2} = \nu u_2 . (p_2 \wedge \Diamond((\mu u_1' . p_1 \wedge \Diamond((p_1 \wedge u_1') \vee (p_2 \wedge u_2))) \vee (p_2 \wedge u_2))))$$

Defining Transition System for Büchi Automaton A:

Let $A = (S_1 \cup S_2, \Sigma, \delta)$ be a Büchi automaton. Let Transition System (TS) over the set of propositional variables $\{p_1, p_2\}$, denoted as T_A, as follows:

- States: (s, w) for $s \in S$ and $w \in \Sigma^\omega$
- Transition $(s, \sigma w) \to (s', w)$ for $s, s' \in S$, $\sigma \in \Sigma$, $w \in \Sigma^\omega$, iff $s' \in \delta(s)(\sigma)$
- Labeling function: $\lambda((s, w)) = \{p_i\}$ iff $s \in S_i$

Game Semantics Theorem: $(s, w) \overset{T}{\vDash} \varphi$ iff Verifier has a winning strategy in $\mathcal{G}(\varphi, T_A)$ from state $(\varphi, (s, w))$

Proof Outline

Converting formula:

$$l_{\text{sol}}^2 = \nu u_2.\Diamond_\delta^2 \left[\left(\mu u_1'.\Diamond_\delta^1[u_1', u_2]\right), u_2\right]$$

$$\overline{\varphi_2} = \nu u_2.(p_2 \land \Diamond((\mu u_1'.p_1 \land \Diamond((p_1 \land u_1') \lor (p_2 \land u_2))) \lor (p_2 \land u_2))))$$

Defining Transition System for Büchi Automaton A:

Let $A = (S_1 \cup S_2, \Sigma, \delta)$ be a Büchi automaton. Let Transition System (TS) over the set of propositional variables $\{p_1, p_2\}$, denoted as T_A, as follows:

- States: (s, w) for $s \in S$ and $w \in \Sigma^\omega$
- Transition $(s, \sigma w) \to (s', w)$ for $s, s' \in S$, $\sigma \in \Sigma$, $w \in \Sigma^\omega$, iff $s' \in \delta(s)(\sigma)$
- Labeling function: $\lambda((s, w)) = \{p_i\}$ iff $s \in S_i$

Game Semantics Theorem: $(s, w) \overset{T}{\vDash} \varphi$ iff Verifier has a winning strategy in $\mathcal{G}(\varphi, T_A)$ from state $(\varphi, (s, w))$

Lemma: Verifier has a winning strategy in $\mathcal{G}(\varphi, T_A)$ from state iff the Büchi automaton accepts w from s.

Conclusion

1. Büchi Automata
2. Coalgebra Determinstic Finite Automata
3. Coalgebra Nondeterministic Finite Automata
4. Coalgebra Possibly Infinite Behavior Nondeterministic Finite Automata
5. Coalgebra Büchi Automata
6. Outline Derivation using Game Semantics