

Coalgebraic Representation of Büchi Automata

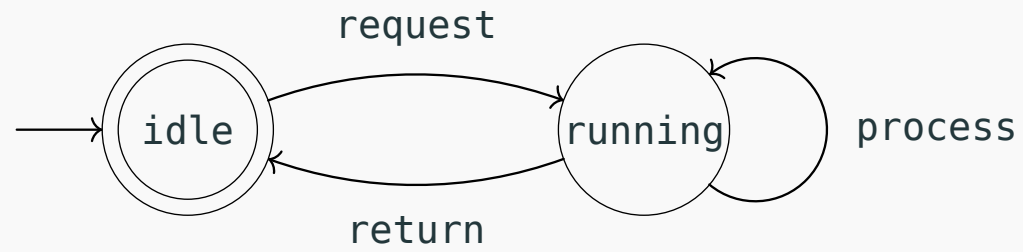
Research Internship Presentation

Jorrit de Boer

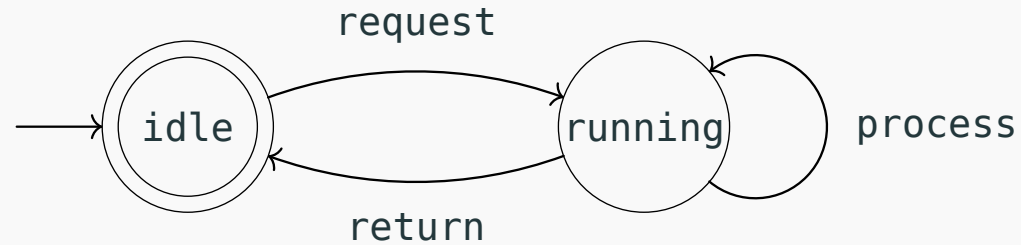
17 January 2025

Büchi Automata

Büchi Automata



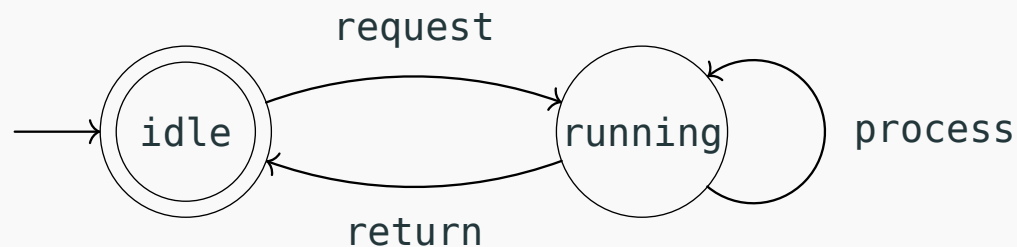
Büchi Automata



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
- $\delta : S \times \Sigma \rightarrow \mathcal{P}(S)$: transition function
- $F \subseteq S$: set of *final* (or *accepting*) states.



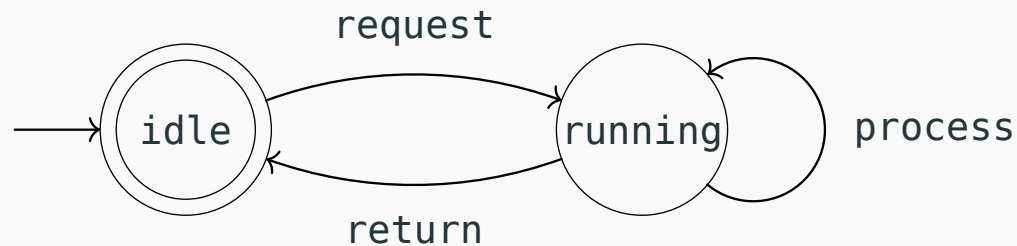
Definition: A (nondeterministic) *Büchi Automaton*

$A = \langle S, \Sigma, \delta, s_0, F \rangle$, where:

- S : finite set of states
- Σ : alphabet
- $s_0 \in S$: initial state
- $\delta : S \times \Sigma \rightarrow \mathcal{P}(S)$: 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 infinitely 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
- $\delta : S \times \Sigma \rightarrow \mathcal{P}(S)$: 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 infinitely often.

Accepted language:

$(\text{request} \cdot \text{process}^* \cdot \text{return})^\omega$

Coalgebra

Final Coalgebra Deterministic Finite Automata

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

The final coalgebra for F is $\langle e, d \rangle : 2^{\Sigma^*} \rightarrow 2 \times (2^{\Sigma^*})^\Sigma$. 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 $\delta : S \times \Sigma \rightarrow S$, $o : S \rightarrow 2$ ($2 = \{0, 1\}$). Can be represented by a coalgebra $\langle o, \delta \rangle : S \rightarrow 2 \times S^\Sigma$ for functor $FS = 2 \times S^\Sigma$

The final coalgebra for F is $\langle e, d \rangle : 2^{\Sigma^*} \rightarrow 2 \times (2^{\Sigma^*})^\Sigma$. 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 $aw \in L$

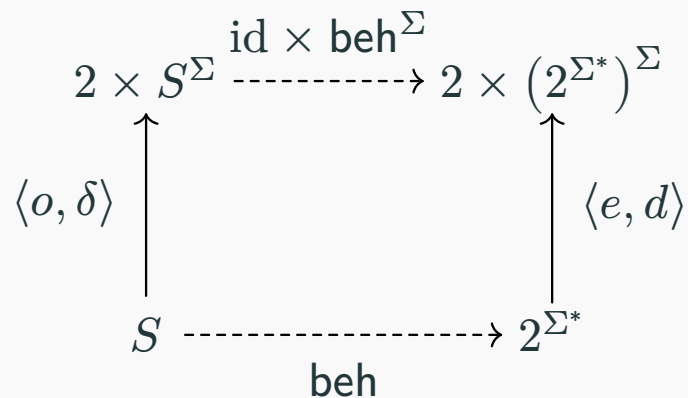
$$\begin{array}{ccc}
 2 \times S^\Sigma & \xrightarrow{\text{id} \times \text{beh}^\Sigma} & 2 \times (2^{\Sigma^*})^\Sigma \\
 \uparrow \langle o, \delta \rangle & & \uparrow \langle e, d \rangle \\
 S & \xrightarrow{\text{beh}} & 2^{\Sigma^*}
 \end{array}$$

Final Coalgebra Deterministic Finite Automata

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

The final coalgebra for F is $\langle e, d \rangle : 2^{\Sigma^*} \rightarrow 2 \times (2^{\Sigma^*})^\Sigma$. 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 $aw \in L$



Following the paths through the diagram we obtain:

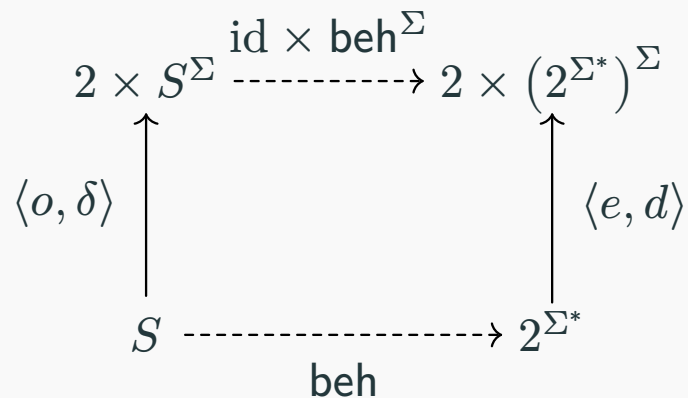
- $\text{beh}(s)(\varepsilon) = o(s)$, and
- $\text{beh}(s)(\sigma w) = \text{beh}(\delta(s)(\sigma))(w)$,

Final Coalgebra Deterministic Finite Automata

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

The final coalgebra for F is $\langle e, d \rangle : 2^{\Sigma^*} \rightarrow 2 \times (2^{\Sigma^*})^\Sigma$. 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 $aw \in L$

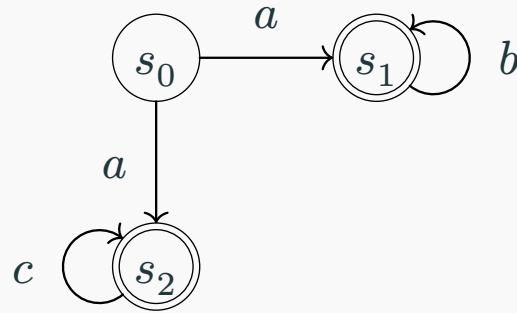


Following the paths through the diagram we obtain:

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

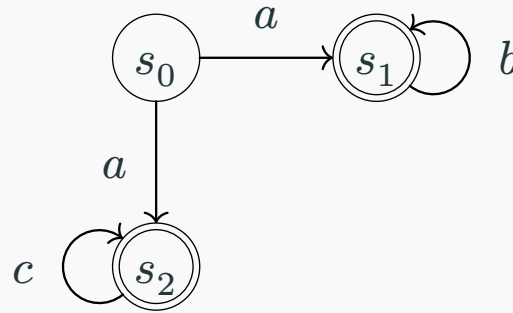
So beh captures exactly the accepted language of the automaton!

Nondeterministic Finite Automata



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

Nondeterministic Finite Automata



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

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

Kleisli Category of the monad \mathcal{P} :

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

Kleisli Category of the monad \mathcal{P} :

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

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

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

- **objects**: the same as for **Sets**, sets
- **morphisms**: $f : X \rightarrow Y$ in $\mathcal{K}\ell(\mathcal{P})$ is $f : X \rightarrow \mathcal{P}(Y)$ in **Sets**.

For morphisms $f : X \rightarrow Y$ and $g : Y \rightarrow Z$ 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 NDA $\langle S, \Sigma, \delta, o \rangle$ by coalgebra $c : S \rightarrow 1 + \Sigma \times S$ for the functor $FS = 1 + \Sigma \times S$, which is $c : S \rightarrow \mathcal{P}(1 + \Sigma \times S)$ in **Sets**.

Lifted Functor in Kleisli Category

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

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

Lifted Functor in Kleisli Category

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

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

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

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

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

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

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*

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

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

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

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

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

- $\text{nil}(\ast) = \varepsilon$
- $\text{cons}(\sigma, w) = \sigma w$

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

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

Final Coalgebra Nondeterministic Automaton

$$\begin{array}{ccc}
 1 + \Sigma \times S & \xrightarrow{1 + \Sigma \times \text{beh}} & 1 + \Sigma \times \Sigma^* \\
 \uparrow c & & \uparrow \eta_{1+\Sigma \times \Sigma^*} \circ [\text{nil}, \text{cons}]^{-1} \text{ in } \mathcal{K}\ell(\mathcal{P}). \\
 S & \xrightarrow{\text{beh}} & \Sigma^*
 \end{array}$$

Final Coalgebra Nondeterministic Automaton

$$\begin{array}{ccc}
 1 + \Sigma \times S & \xrightarrow{1 + \Sigma \times \text{beh}} & 1 + \Sigma \times \Sigma^* \\
 \uparrow c & & \uparrow \eta_{1+\Sigma \times \Sigma^*} \circ [\text{nil}, \text{cons}]^{-1} \text{ in } \mathcal{K}\ell(\mathcal{P}). \\
 S & \xrightarrow{\text{beh}} & \Sigma^*
 \end{array}$$

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

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

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

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

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

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

$$\begin{array}{ccc} \overline{F}S & \overset{\overline{F}(\text{beh})}{\rightsquigarrow} & \overline{F}Z \\ \uparrow c & & \uparrow \eta_{FZ} \circ \xi \\ S & \overset{\text{beh}}{\rightsquigarrow} & Z \end{array} \text{ in } \mathcal{K}\ell(\mathcal{P}),$$

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

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

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

$$\begin{array}{ccc}
 & 1 + \Sigma \times \text{beh}_c^\infty & \\
 1 + \Sigma \times S & \rightsquigarrow & 1 + \Sigma \times \Sigma^\infty \\
 \uparrow c & & \uparrow J\xi \\
 S & \rightsquigarrow_{\text{beh}_c^\infty} & \Sigma^\infty
 \end{array}
 \quad \text{in } \mathcal{K}\ell(\mathcal{P}).$$

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

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

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

$$\begin{array}{ccc}
 \Sigma \times [\text{beh}_1, \text{beh}_2] & & \Sigma \times [\text{beh}_1, \text{beh}_2] \\
 \Sigma \times S \rightsquigarrow \Sigma \times \Sigma^\omega & & \Sigma \times S \rightsquigarrow \Sigma \times \Sigma^\omega \\
 \uparrow c_1 & \xrightarrow{\quad \mu \quad} & \uparrow \eta_{\Sigma^\omega} \circ d \\
 S_1 \rightsquigarrow \Sigma^\omega & & S_2 \rightsquigarrow \Sigma^\omega \\
 \text{beh}_1 & & \text{beh}_2
 \end{array}$$

in $\mathcal{K}\ell(\mathcal{P})$.

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

$$\begin{array}{ccc}
 \begin{array}{ccc}
 \Sigma \times [\text{beh}_1, \text{beh}_2] \\
 \Sigma \times S \rightsquigarrow \Sigma \times \Sigma^\omega \\
 \uparrow c_1 \quad \quad \quad \uparrow \eta_{\Sigma^\omega} \circ d \\
 \text{=}_{\mu} \\
 S_1 \rightsquigarrow \Sigma^\omega \\
 \text{beh}_1
 \end{array}
 &
 \begin{array}{ccc}
 \Sigma \times [\text{beh}_1, \text{beh}_2] \\
 \Sigma \times S \rightsquigarrow \Sigma \times \Sigma^\omega \\
 \uparrow c_2 \quad \quad \quad \uparrow \eta_{\Sigma^\omega} \circ d \\
 \text{=}_{\nu} \\
 S_2 \rightsquigarrow \Sigma^\omega \\
 \text{beh}_2
 \end{array}
 &
 \text{in } \mathcal{KL}(\mathcal{P}).
 \end{array}$$

$$u_1 \stackrel{=}{=}_{\mu} (\eta_{\Sigma^\omega} \circ d)^{-1} \odot \overline{F}[u_1, u_2] \odot c_1$$

$$u_2 \stackrel{=}{=}_{\nu} (\eta_{\Sigma^\omega} \circ d)^{-1} \odot \overline{F}[u_1, u_2] \odot c_2$$

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

$$\begin{array}{ccc}
 \begin{array}{ccc}
 \Sigma \times [\text{beh}_1, \text{beh}_2] \\
 \Sigma \times S \rightsquigarrow \Sigma \times \Sigma^\omega \\
 \uparrow c_1 \quad \quad \quad \uparrow \eta_{\Sigma^\omega} \circ d \\
 S_1 \rightsquigarrow \Sigma^\omega \\
 \text{beh}_1
 \end{array} & \overset{\mu}{=} & \begin{array}{ccc}
 \Sigma \times [\text{beh}_1, \text{beh}_2] \\
 \Sigma \times S \rightsquigarrow \Sigma \times \Sigma^\omega \\
 \uparrow c_2 \quad \quad \quad \uparrow \eta_{\Sigma^\omega} \circ d \\
 S_2 \rightsquigarrow \Sigma^\omega \\
 \text{beh}_2
 \end{array}
 \end{array}
 \quad \text{in } \mathcal{K}\ell(\mathcal{P}).$$

$$u_1 \overset{\mu}{=} \Diamond_\delta([u_1, u_2]) \upharpoonright S_1 \quad u_2 \overset{\nu}{=} \Diamond_\delta([u_1, u_2]) \upharpoonright S_2$$

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

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

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

Using Game Semantics to derive

outline this? steps and saying these things are possible?

Conclusion
