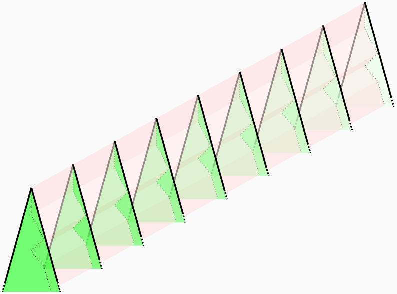
Automata-theoretic Synthesis for Probabilistic Environments

Christoph Welzel

July 31, 2018

Informatik 7, RWTH Aachen



Word-Automata

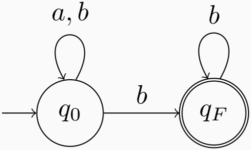
ω-regular Languages

Theorem
The class of recognizable languages coincides for NBAs, NPAs and DPAs and is called ω-regular languages. DBAs are strictly less expressive.

$$\mathcal{L} = \{\alpha \in \{a, b\}^\omega \mid a \notin \mathrm{Inf}(\alpha)\}$$

Theorem
The class of recognizable languages coincides for NBAs, NPAs and DPAs and is called ω-regular languages. DBAs are strictly less expressive.

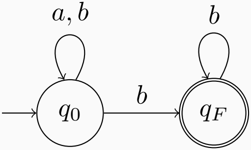
$$\mathcal{L} = \{\alpha \in \{a, b\}^\omega \mid a \notin \mathrm{Inf}(\alpha)\}$$

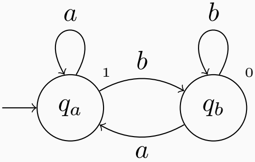


ω-regular Languages

Theorem
The class of recognizable languages coincides for NBAs, NPAs and DPAs and is called ω-regular languages. DBAs are strictly less expressive. However, NBA-determinisation is inherently costly ($> n!$).

$$\mathcal{L} = \{\alpha \in \{a, b\}^{\omega} \mid a \notin \mathsf{Inf}(\alpha)\}$$

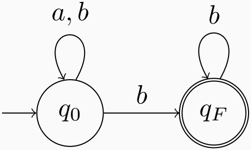


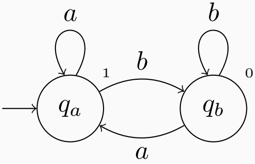


ω-regular Languages

Theorem
The class of recognizable languages coincides for NBAs, NPAs and DPAs and is called ω-regular languages. DBAs are strictly less expressive. However, NBA-determinisation is inherently costly ($> n!$).

$$\mathcal{L} = \{\alpha \in \{a, b\}^\omega \mid a \notin \mathsf{Inf}(\alpha)\}$$





b^ω,

$b^{n_0} a b^\omega$,

$b^{n_0} a b^{n_1} a b^\omega$,

$\vdots$

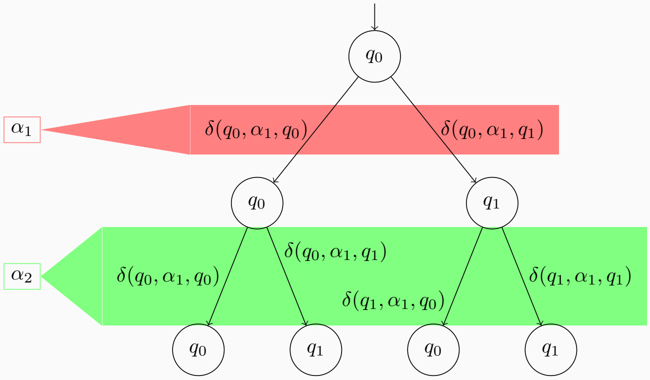
Probabilistic Büchi Automata

$$\mathcal{A} = \left(Q, \Sigma, \delta : Q \times \Sigma \times Q \to [0,1], q_0, F\right)$$

- Q: finite state set
- Σ: finite alphabet
- $q_0 \in Q$: initial state
- $\delta : Q \times \Sigma \times Q \to [0,1]$: transition probability function
- $F \subseteq Q$: final states

Probabilistic Büchi Automata

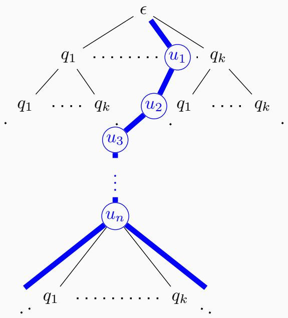
$$\mathcal{A} = (Q, \Sigma, \delta : Q \times \Sigma \times Q \to [0,1], q_0, F)$$



Probabilistic Büchi Automata

$$\mathcal{A} = (Q, \Sigma, \delta : Q \times \Sigma \times Q \to [0,1], q_0, F)$$

- cylindric sets:
  $\mathrm{cyl}(u) = \{u \cdot \alpha : \alpha \in Q^\omega\}$

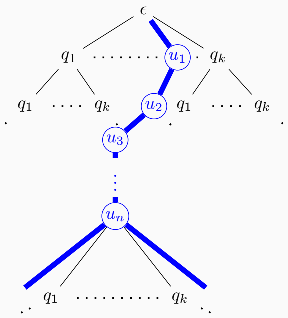


Probabilistic Büchi Automata

$$\mathcal{A} = (Q, \Sigma, \delta : Q \times \Sigma \times Q \to [0,1], q_0, F)$$

- cylindric sets:

  $\mathsf{cyl}(u) = \{u \cdot \alpha : \alpha \in Q^\omega\}$



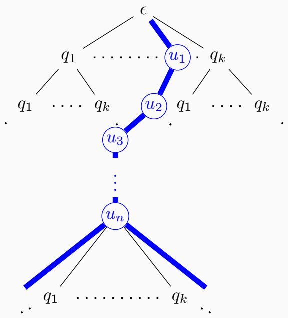
- α induces probability space

$$(Q^\omega, \mathcal{B}(Q), \mu_\alpha)$$

Probabilistic Büchi Automata

$$\mathcal{A} = \left(Q, \Sigma, \delta : Q \times \Sigma \times Q \to [0,1], q_0, F\right)$$

- cylindric sets:
  $\mathrm{cyl}(u) = \{u \cdot \alpha : \alpha \in Q^\omega\}$



- α induces probability space

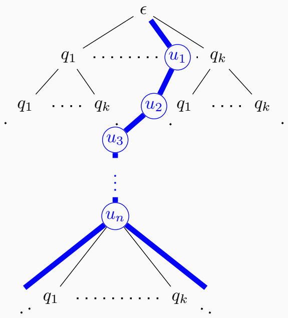
$$(Q^\omega, \mathcal{B}(Q), \mu_\alpha)$$

- positive:
  $\mu_\alpha(\mathrm{Acc}_{\text{Büchi}}(F)) > 0$

Probabilistic Büchi Automata

$$\mathcal{A} = (Q, \Sigma, \delta : Q \times \Sigma \times Q \to [0,1], q_0, F)$$

- cylindric sets:
  $\text{cyl}(u) = \{u \cdot \alpha : \alpha \in Q^\omega\}$



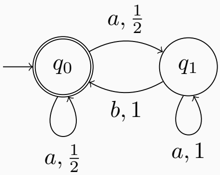
- α induces probability space

$$(Q^\omega, \mathcal{B}(Q), \mu_\alpha)$$

- positive:
  $\mu_\alpha(\text{Acc}_{\text{Büchi}}(F)) > 0$
- almost-sure:
  $\mu_\alpha(\text{Acc}_{\text{Büchi}}(F)) = 1$

Probabilistic Automata - Examples

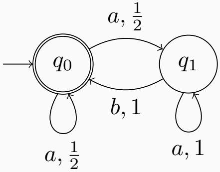
Positive Acceptance



$$\underbrace{\left\{ a^{k_1} b a^{k_2} b \cdots : \substack{k_i > 0 \text{ for all } i > 0, \\ \prod_{i>0} \left(1 - \frac{1}{2}^{k_i}\right) > 0} \right\}}_{\mathcal{L}_1}$$

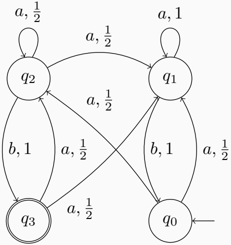
Probabilistic Automata - Examples

Positive Acceptance



$$\underbrace{\left\{a^{k_1} b a^{k_2} b \cdots : \substack{k_i > 0 \text{ for all } i > 0, \\ \prod_{i>0}\left(1 - \frac{1}{2}^{k_i}\right) > 0}\right\}}_{\mathcal{L}_1}$$

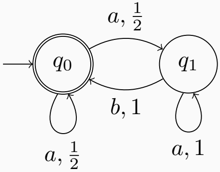
Almost-Sure Acceptance



$$\underbrace{\left\{a^{k_1} b a^{k_2} b \cdots : \substack{k_i > 0 \text{ for all } i > 0 \\ \prod_{i>0}\left(1 - \frac{1}{2}^{k_i}\right) = 0}\right\}}_{\mathcal{L}_2}$$

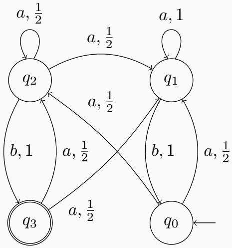
Probabilistic Automata - Examples

Positive Acceptance



$$\underbrace{\left\{ a^{k_1} b a^{k_2} b \cdots : \substack{k_i > 0 \text{ for all } i>0, \\ \prod_{i>0}\left(1-\frac{1}{2}^{k_i}\right)>0} \right\}}_{\mathcal{L}_1}$$

Almost-Sure Acceptance



$$\underbrace{\left\{ a^{k_1} b a^{k_2} b \cdots : \substack{k_i > 0 \text{ for all } i>0 \\ \prod_{i>0}\left(1-\frac{1}{2}^{k_i}\right)=0} \right\}}_{\mathcal{L}_2}$$

$$\overline{\mathcal{L}_1} = \mathcal{L}_2 \cup \underbrace{b\Sigma^\omega + \Sigma^* bb\Sigma^\omega + \Sigma^* a^\omega}_{\omega-\text{regular}}$$

Probabilistic Automata - Properties

Positive Acceptance

Almost-Sure Acceptance

Probabilistic Automata - Properties

Positive Acceptance

- strictly subsumes ω-regular

Almost-Sure Acceptance

Probabilistic Automata - Properties

Positive Acceptance

- strictly subsumes ω-regular

Almost-Sure Acceptance

$$\left\{ a^{k_1} b a^{k_2} b \cdots : \begin{array}{c} k_i > 0 \text{ for all } i > 0, \\ \prod_{i>0}\left(1 - \frac{1}{2}^{k_i}\right) > 0 \end{array} \right\}$$

Probabilistic Automata - Properties

Positive Acceptance

- strictly subsumes ω-regular
- recognizable languages form Boolean-algebra

Almost-Sure Acceptance

Probabilistic Automata - Properties

Positive Acceptance

- strictly subsumes ω-regular
- recognizable languages form Boolean-algebra
- undecidable emptiness

Almost-Sure Acceptance

Probabilistic Automata - Properties

Positive Acceptance

- strictly subsumes ω-regular
- recognizable languages form Boolean-algebra
- undecidable emptiness

Almost-Sure Acceptance

- incomparable with ω-regular

Probabilistic Automata - Properties

Positive Acceptance

- strictly subsumes ω-regular
- recognizable languages form Boolean-algebra
- undecidable emptiness

$$\{\alpha \in \{a, b\}^\omega \mid a \notin \mathsf{Inf}(\alpha)\}$$

Almost-Sure Acceptance

- incomparable with ω-regular

Probabilistic Automata - Properties

Positive Acceptance

- strictly subsumes ω-regular
- recognizable languages form Boolean-algebra
- undecidable emptiness

$$\{\alpha \in \{a, b\}^\omega \mid a \notin \mathsf{Inf}(\alpha)\}$$

Almost-Sure Acceptance

- incomparable with ω-regular

$$\overline{\left\{ a^{k_1} b a^{k_2} b \cdots : \substack{k_i > 0 \text{ for all } i > 0, \\ \prod_{i>0}\left(1 - \frac{1}{2}^{k_i}\right) > 0} \right\}}$$

$$= \left\{ a^{k_1} b a^{k_2} b \cdots : \substack{k_i > 0 \text{ for all } i > 0 \\ \prod_{i>0}\left(1 - \frac{1}{2}^{k_i}\right) = 0} \right\}$$

$$\cup\ b\Sigma^\omega + \Sigma^* bb\Sigma^\omega + \Sigma^* a^\omega$$

Probabilistic Automata - Properties

Positive Acceptance

- strictly subsumes ω-regular
- recognizable languages form Boolean-algebra
- undecidable emptiness

Almost-Sure Acceptance

- incomparable with ω-regular
- decidable emptiness

Probabilistic Automata - Properties

Positive Acceptance

- strictly subsumes ω-regular
- recognizable languages form Boolean-algebra
- undecidable emptiness

Almost-Sure Acceptance

- incomparable with ω-regular
- decidable emptiness
- Parity-condition coincides with positive acceptance of Büchi- or Parity-condition

Tree-Automata

Definition

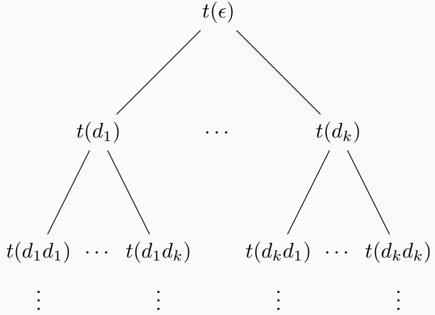
$$\mathcal{A} = (Q, q_0, D, \Sigma, \Delta, \mathsf{Acc} \subseteq Q^\omega)$$

- Q: finite state set
- $q_0 \in Q$: initial state
- D: finite set of directions
- Σ: finite alphabet
- Δ: finite set of transitions
- Acc: accepted language

Definition

$$\mathcal{A} = (Q, q_0, D, \Sigma, \Delta, \mathrm{Acc} \subseteq Q^\omega)$$

D-ary Σ-tree: $t : D^* \to \Sigma$

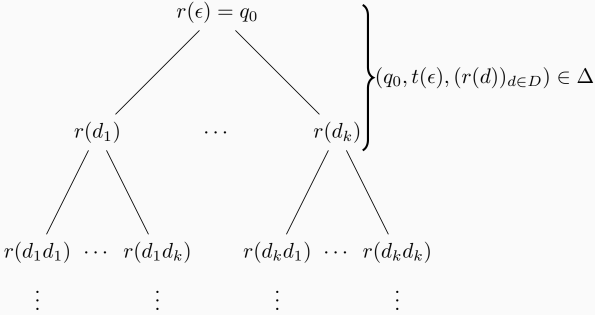


Definition

$$\mathcal{A} = (Q, q_0, D, \Sigma, \Delta, \mathsf{Acc} \subseteq Q^\omega)$$

D-ary Σ-tree: $t : D^* \to \Sigma$

D-ary Q-run: $r : D^* \to Q$

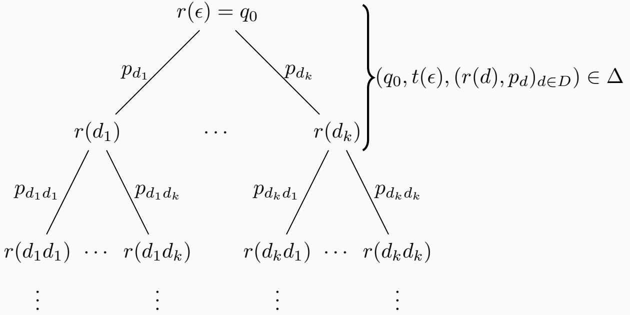


Definition

$$\mathcal{A} = (Q, q_0, D, \Sigma, \Delta, \mathsf{Acc} \subseteq Q^{\omega})$$

D-ary Σ-tree: $t : D^* \to \Sigma$

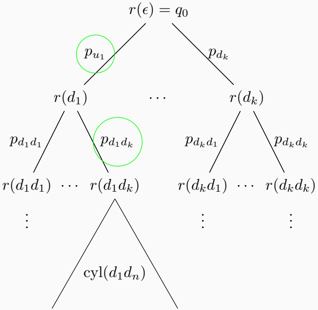
D-ary Q-run: $r : D^* \to Q$



Definition

$$\mathcal{A} = (Q, q_0, D, \Sigma, \Delta, \mathsf{Acc} \subseteq Q^\omega)$$

D-ary Σ-tree: $t : D^* \to \Sigma$ D-ary Q-run: $r : D^* \to Q$



$$(D^\omega, \mathcal{B}(D), \mu_r) \Rightarrow \mu_r(\{\rho \in D^\omega \mid r(\rho) \in \mathsf{Acc}\})$$

Tree Automata - Example

$$\mathcal{A} = (Q = \{q_a, q_b\}, q_a, D = \{0, 1\}, \Sigma = \{a, b\}, \Delta, F = \{q_a\})$$

Tree Automata - Example

$$\mathcal{A} = \left(Q = \{q_a, q_b\}, q_a, D = \{0,1\}, \Sigma = \{a, b\}, \Delta, F = \{q_a\}\right)$$

$$\Delta = \left\{(q, \sigma, q_\sigma, q_\sigma) : {}_{\sigma \in \{a,b\}}^{q \in Q,}\right\} \text{ or } \left\{\left(q, \sigma, q_\sigma, \frac{1}{2}, q_\sigma, \frac{1}{2}\right) : {}_{\sigma \in \{a,b\}}^{q \in Q,}\right\}$$

Tree Automata - Example

$$\mathcal{A} = (Q = \{q_a, q_b\}, q_a, D = \{0,1\}, \Sigma = \{a,b\}, \Delta, F = \{q_a\})$$

$$\Delta = \left\{(q, \sigma, q_\sigma, q_\sigma) : {}_{\sigma\in\{a,b\}}^{q\in Q,}\right\} \text{ or } \left\{\left(q, \sigma, q_\sigma, \frac{1}{2}, q_\sigma, \frac{1}{2}\right) : {}_{\sigma\in\{a,b\}}^{q\in Q,}\right\}$$

$$\mathcal{L} = \left\{\begin{matrix} t : \{0,1\}^* \to \{a,b\} \mid \\ A^t_\infty = D^\omega \end{matrix}\right\} \text{ or } \left\{\begin{matrix} t : \{0,1\}^* \to \{a,b\} \mid \\ \mu_{\frac{1}{2}}(A^t_\infty) = 1 \; (\mu_{\frac{1}{2}}(A^t_\infty) > 0) \end{matrix}\right\}$$

where

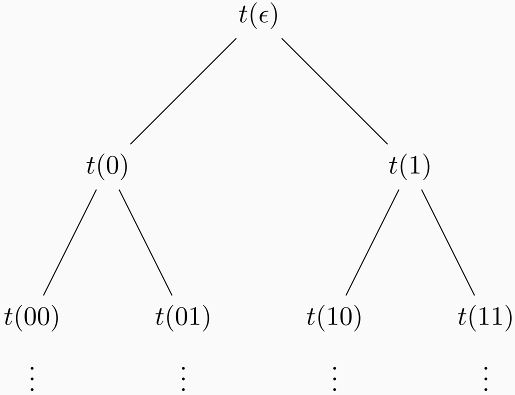
$$A^t_\infty = \{\alpha_1\alpha_2 \dots \in D^\omega \mid a \in \mathrm{Inf}(t(\epsilon)t(\alpha_1)t(\alpha_1\alpha_2)\dots)\}$$

Alternating Tree Automata - Definition

$$\mathcal{A} = (Q, q_0, D, \Sigma, \Delta, \mathsf{Acc} \subseteq Q^\omega)$$

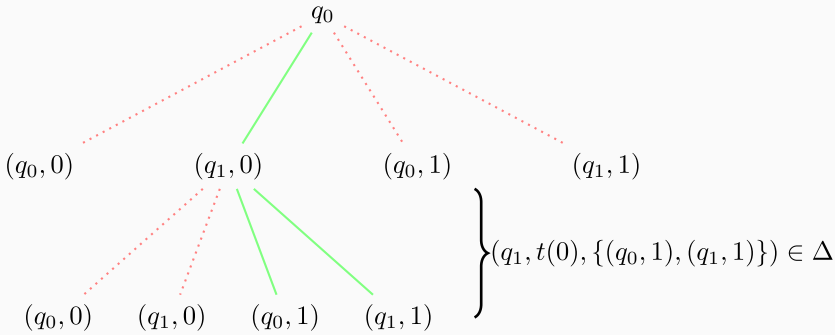
Alternating Tree Automata - Definition

$$\mathcal{A} = (Q, q_0, D, \Sigma, \Delta, \mathsf{Acc} \subseteq Q^\omega)$$

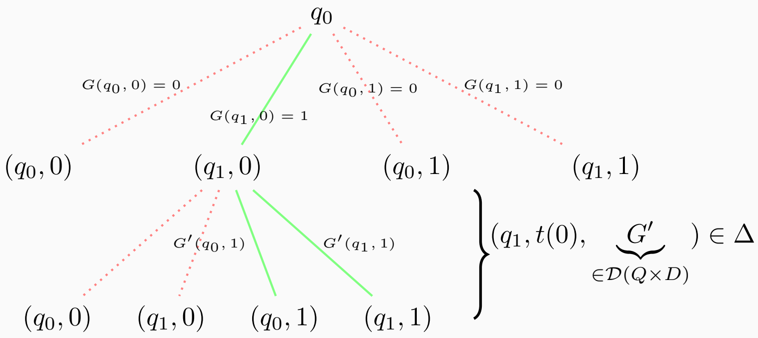


Alternating Tree Automata - Definition

$$\mathcal{A} = (Q, q_0, D, \Sigma, \Delta, \mathsf{Acc} \subseteq Q^\omega)$$



$$\mathcal{A} = (Q, q_0, D, \Sigma, \Delta, \mathsf{Acc} \subseteq Q^\omega)$$



$$\Rightarrow \left((Q \times D)^\omega, \mathcal{B}(Q \times D), \mu_r\right)$$

Alternating Tree Automaton - Example

$$\left\{ t : \{0,1\}^* \to \{a,b\} \mid \substack{\text{there is } u \in \{0,1\}^* \\ \text{with } t(u)=a} \right\}$$

- search and found state:

  q_s, q_f

Alternating Tree Automaton - Example

$$\left\{ t : \{0,1\}^* \to \{a, b\} \mid {}^{\text{there is } u \in \{0,1\}^*}_{\text{with } t(u)=a} \right\}$$

- search and found state:

  q_s, q_f
- non-deterministically move

  q_s

Alternating Tree Automaton - Example

$$\left\{ t : \{0,1\}^* \to \{a, b\} \mid {}^{\text{there is } u \in \{0,1\}^*}_{\text{with } t(u)=a} \right\}$$

- search and found state:

  q_s, q_f
- non-deterministically move

  q_s
- transform into positive

  sink-state q_f when reading a

Alternating Tree Automaton - Example

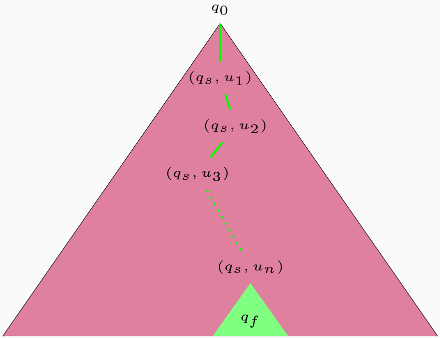
$$\left\{ t : \{0,1\}^* \to \{a,b\} \mid {}^{\text{there is } u \in \{0,1\}^*}_{\text{with } t(u)=a} \right\}$$

- search and found state:
  q_s, q_f
- non-deterministically move
  q_s
- transform into positive
  sink-state q_f when reading a
- use $\{q_f\}$ as Büchi-condition

Alternating Tree Automaton - Example

$$\left\{ t : \{0,1\}^* \to \{a,b\} \mid {}^{\text{there is } u \in \{0,1\}^*}_{\text{with } t(u)=a} \right\}$$

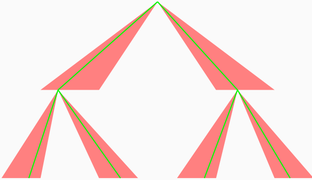
- search and found state: q_s, q_f
- non-deterministically move q_s
- transform into positive sink-state q_f when reading a
- use $\{q_f\}$ as Büchi-condition



Weighted Descent Tree Automata - Structural Properties

choiceless (c.l.): $|\{G{:}(q,\sigma,G)\in\Delta\}|=1$ for all $q\in Q,\sigma\in\Sigma$

uni-directional (u.d.): for every $G\in\mathcal{G}(\mathcal{A}),d\in D$ exists *at most* one $p\in Q$ with $G(d,p)>0$



Theorem (Simulation Theorem)
There is an effective construction which, when given an APTA, produces an equivalent PTA. Furthermore, given an ABTA, there is a way to effectively construct an equivalent BTA.

- recognizable languages form a Boolean-algebra
- decidable emptiness

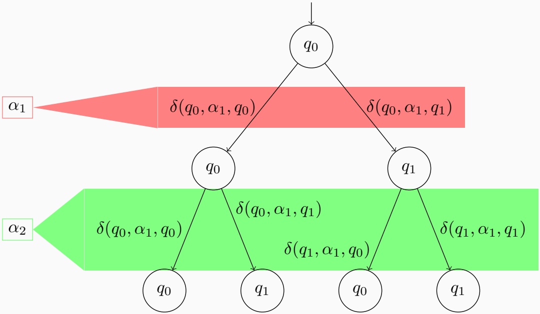
Weighted Descent Tree Automata - Word-Automata

for unary trees:

Weighted Descent Tree Automata - Word-Automata

for unary trees:

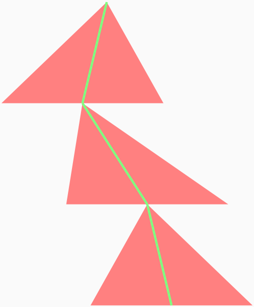
- c.l. WDTAs "equivalent" to PPAs



Weighted Descent Tree Automata - Word-Automata

for unary trees:

- c.l. WDTAs "equivalent" to PPAs
- u.d. WDTAs "equivalent" to ω-regular



Weighted Descent Tree Automata - Word-Automata

for unary trees:

- c.l. WDTAs "equivalent" to PPAs
- u.d. WDTAs "equivalent" to ω-regular

$\Rightarrow$ undecidable emptiness for (c.l.) WDTAs with positive Büchi-acceptance and almost-sure Parity-acceptance

Weighted Descent Tree Automata - Word-Automata

for unary trees:

- c.l. WDTAs "equivalent" to PPAs
- u.d. WDTAs "equivalent" to ω-regular

$\Rightarrow$ undecidable emptiness for (c.l.) WDTAs with positive Büchi-acceptance and almost-sure Parity-acceptance

$\Rightarrow$ Simulation Theorem does not translate to WDTAs

Partially Observable Markov Decision Processes

$$\mathcal{S} = \left(S, s_0, A, (\tau_a)_{a \in A}, \sim\right)$$

Partially Observable Markov Decision Processes

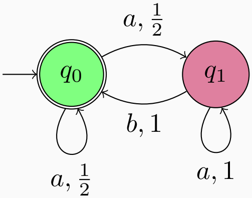
$$\mathcal{S} = \left(S, s_0, A, (\tau_a)_{a \in A}, \sim\right)$$

- S: finite state set
- $s_0 \in S$: initial state
- A: finite state of actions
- $\tau_a \in \mathcal{D}(S \times S)$: transition probabilities
- $\sim$: observable equivalence classes
- strategy $f : [S]^*_\sim \to A$

$\Rightarrow$ $(S^\omega, \mathcal{B}(S), \mu_f)$

Partially Observable Markov Decision Processes

$$\mathcal{S} = \left(S, s_0, A, (\tau_a)_{a \in A}, \sim\right)$$

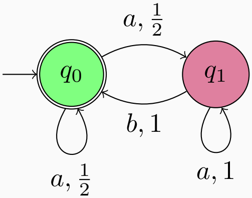


Partially Observable Markov Decision Processes

$$\mathcal{S} = \left(S, s_0, A, (\tau_a)_{a \in A}, \sim\right)$$

$$f(\epsilon) = a$$

$$f(p_1 \dots p_n) = \begin{cases} a & \text{if } p_n = q_0, \\ b & \text{otherwise.} \end{cases}$$

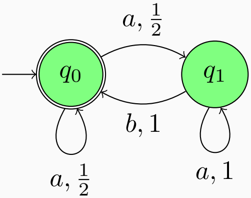


Partially Observable Markov Decision Processes

$$\mathcal{S} = \left(S, s_0, A, (\tau_a)_{a \in A}, \sim\right)$$

$$f(\epsilon) = a$$

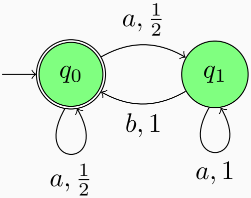
$$f(p_1 \ldots p_n) = \begin{cases} a & \text{if } p_n = q_0, \\ b & \text{otherwise.} \end{cases}$$



Partially Observable Markov Decision Processes

$$\mathcal{S} = \left(S, s_0, A, (\tau_a)_{a\in A}, \sim\right)$$

$\mu_f(\mathsf{Acc}_{\text{Büchi}}(F)) > 0$ if and only if $f \in \mathcal{L}(\mathcal{P})$.



Partially Observable Markov Decision Processes

$$\mathcal{S} = \left(S, s_0, A, (\tau_a)_{a \in A}, \sim\right)$$

Computing Strategies	Fully observable	Partially observable
Positive Büchi	PTIME	Undecidable
Almost-Sure Büchi	PTIME	EXPTIME
Positive Parity	PTIME	Undecidable
Almost-Sure Parity	PTIME	Undecidable

WDTAs and POMDPs

$$\mathcal{A} = (Q, q_0, D, \Sigma, \Delta, \mathrm{par}) \text{ and } \mathcal{S} = \left(S, s_0, A, (\tau_a)_{a \in A}, \sim\right)$$

WDTAs and POMDPs

$$\mathcal{A} = (Q, q_0, D, \Sigma, \Delta, \mathrm{par}) \text{ and } \mathcal{S} = \left(S, s_0, A, (\tau_a)_{a \in A}, \sim\right)$$

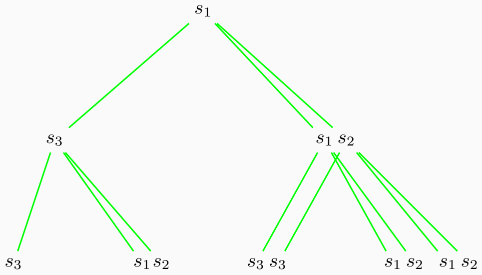
- strategies are
  $[S]_\sim$-ary A-trees

WDTAs and POMDPs

$$\mathcal{A} = (Q, q_0, D, \Sigma, \Delta, \mathrm{par}) \text{ and } \mathcal{S} = \big(S, s_0, A, (\tau_a)_{a \in A}, \sim\big)$$

- strategies are $[S]_\sim$-ary A-trees
- translate POMDP to c.l. WDTAs

"directions are observations"

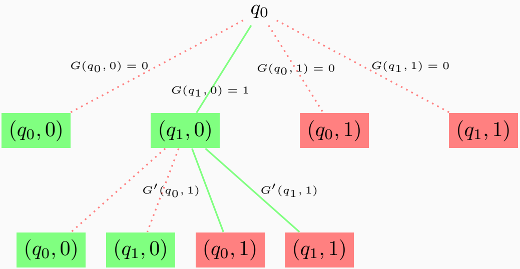


WDTAs and POMDPs

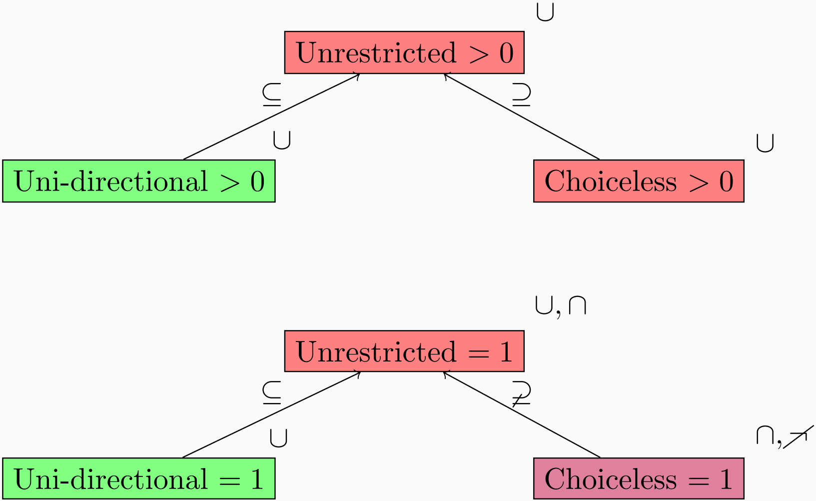
$$\mathcal{A} = (Q, q_0, D, \Sigma, \Delta, \mathsf{par}) \text{ and } \mathcal{S} = \big(S, s_0, A, (\tau_a)_{a \in A}, \sim\big)$$

- strategies are $[S]_\sim$-ary A-trees
- translate POMDP to c.l. WDTAs
- use POMDPs as emptiness game for c.l. WDTAs

"play the run"

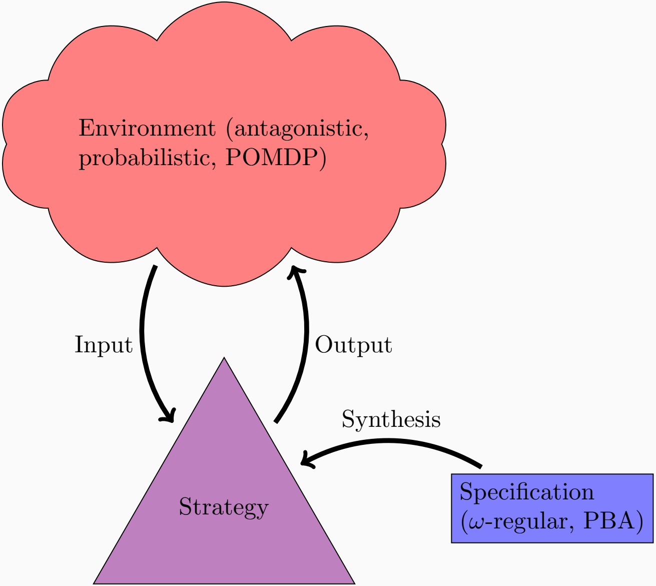


Weighted Descent Tree Automata - Overview



Synthesis

Setting


Environment (antagonistic, probabilistic, POMDP)
Input
Output
Synthesis
Strategy
Specification (ω-regular, PBA)

Setting

Definition (Synthesis Problem)
Given a logic $\mathbb{L}$. Compute for every formula $\phi(\cdot,\cdot) \in \mathbb{L}$ over inputs I and outputs J an algorithm $S : I^+ \to J$ such that $\phi(\alpha, S(\alpha_1)S(\alpha_1\alpha_2)\dots)$ is true for all $\alpha_1\alpha_2 \dots \in I^\omega$ or prove that such an S cannot exist.

Definition (Synthesis Problem)
Given a logic $\mathbb{L}$. Compute for every formula $\phi(\cdot,\cdot) \in \mathbb{L}$ over inputs I and outputs J an algorithm $S : I^{+} \rightarrow J$ such that $\phi(\alpha, S(\alpha_1)S(\alpha_1\alpha_2)\dots)$ is true for all $\alpha_1\alpha_2 \dots \in I^{\omega}$ or prove that such an S cannot exist.

$$(j_0, i_0)\,(j_1, i_1)\cdots \in \mathcal{L}_{\phi} \text{ iff } \phi(i_0 i_1 \dots, j_1 j_2 \dots) \text{ is true.}$$

Antagonistic Environment - ω-regular Specification

- ω-regular specification

Antagonistic Environment - ω-regular Specification

- ω-regular specification

$\Rightarrow$ DPA $(Q, J \times I, \delta, q_0, \mathrm{par})$

Antagonistic Environment - ω-regular Specification

- ω-regular specification

$\Rightarrow$ DPA $(Q, J \times I, \delta, q_0, \mathrm{par})$

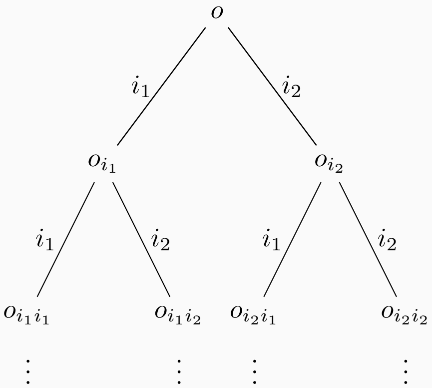
$\Rightarrow$ PTA $(Q, q_0, I, J, \Delta, \mathrm{par})$
with
$\Delta = \left\{ \substack{\left(q, i, (\delta(q,(i,j)))_{j \in J}\right) : \\ q \in Q, i \in I} \right\}$

Antagonistic Environment - ω-regular Specification

- ω-regular specification

$\Rightarrow$ DPA $(Q, J \times I, \delta, q_0, \mathrm{par})$

$\Rightarrow$ PTA $(Q, q_0, I, J, \Delta, \mathrm{par})$
with
$\Delta = \left\{ \substack{(q,i,(\delta(q,(i,j)))_{j\in J}): \\ q\in Q, i\in I} \right\}$



Antagonistic Environment - PBA Specification

$$(Q, J \times I, \delta, q_0, F)$$

Antagonistic Environment - PBA Specification

$$(Q, J \times I, \delta, q_0, F)$$

- Partially Observable Stochastic Game (POSG):

Antagonistic Environment - PBA Specification

$$(Q, J \times I, \delta, q_0, F)$$

- Partially Observable Stochastic Game (POSG):
- POMDP with antagonistic players (Eve & Adam)

Antagonistic Environment - PBA Specification

$$(Q, J \times I, \delta, q_0, F)$$

- Partially Observable Stochastic Game (POSG):
- POMDP with antagonistic players (Eve & Adam)
- $G = \left(S, s_0, E, A, (\tau_{e,a})_{e \in E, a \in A}, \sim_E, \sim_A, F'\right)$

Antagonistic Environment - PBA Specification

$$(Q, J \times I, \delta, q_0, F)$$

- Partially Observable Stochastic Game (POSG):
- POMDP with antagonistic players (Eve & Adam)
- $G = \left(S, s_0, E, A, (\tau_{e,a})_{e \in E, a \in A}, \sim_E, \sim_A, F'\right)$
- $\begin{array}{r} \textsc{Eve}\ f : [S]^*_{\sim_A} \to A \\ \textsc{Adam}\ g : [S]^*_{\sim_E} \to E \end{array} \Rightarrow (S^\omega, \mathcal{B}(S), \mu_{f,g})$

Antagonistic Environment - PBA Specification

$$(Q, J \times I, \delta, q_0, F)$$

- Partially Observable Stochastic Game (POSG):
- POMDP with antagonistic players ($\textsc{Eve}$ & $\textsc{Adam}$)
- $G = \left(S, s_0, E, A, (\tau_{e,a})_{e \in E, a \in A}, \sim_E, \sim_A, F'\right)$
- $\begin{array}{c} \textsc{Eve}\ f : [S]^*_{\sim_A} \to A \\ \textsc{Adam}\ g : [S]^*_{\sim_E} \to E \end{array} \Rightarrow (S^\omega, \mathcal{B}(S), \mu_{f,g})$
- f almost-surely wins for $\textsc{Eve}$ if $\mu_{f,g}(\mathsf{Acc}_{\text{Büchi}}(F)) = 1$ for all g

Antagonistic Environment - PBA Specification

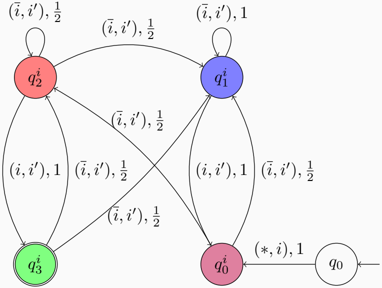
$$(Q, J \times I, \delta, q_0, F) \Rightarrow \left(S, s_0, E, A, (\tau_{e,a})_{e \in E, a \in A}, \sim_E, \sim_A, F'\right)$$

- Partially Observable Stochastic Game (POSG):
- POMDP with antagonistic players ($\textsc{Eve}$ & $\textsc{Adam}$)
- $G = \left(S, s_0, E, A, (\tau_{e,a})_{e \in E, a \in A}, \sim_E, \sim_A, F'\right)$
- $\begin{array}{r} \textsc{Eve}\ f : [S]^*_{\sim_A} \to A \\ \textsc{Adam}\ g : [S]^*_{\sim_E} \to E \end{array} \Rightarrow (S^\omega, \mathcal{B}(S), \mu_{f,g})$
- f almost-surely wins for $\textsc{Eve}$ if $\mu_{f,g}(\mathsf{Acc}_{\text{Büchi}}(F)) = 1$ for all g

Antagonistic Environment - PBA Specification

"input-/output-game with unobservable stochastic background"

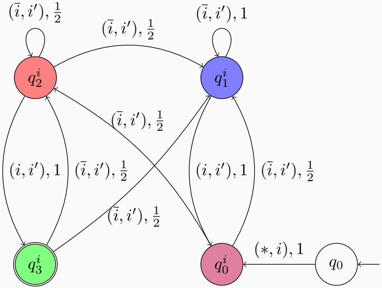
$$\left\{ a^{k_1} b a^{k_2} b \cdots : \begin{array}{c} k_i > 0 \text{ for all } i > 0 \\ \prod_{i>0}\left(1 - \frac{1}{2}^{k_i}\right) = 0 \\ a\text{:last input differs from current output} \\ b\text{:last input is equal to current output} \end{array} \right\}$$

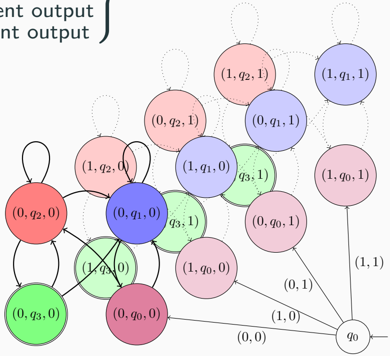


Antagonistic Environment - PBA Specification

"input-/output-game with unobservable stochastic background"

$$\left\{ a^{k_1} b a^{k_2} b \cdots : \begin{array}{c} k_i>0 \text{ for all } i>0 \\ \prod_{i>0}\left(1-\frac{1}{2}^{k_i}\right)=0 \\ a\text{:last input differs from current output} \\ b\text{:last input is equal to current output} \end{array} \right\}$$





Definition
Given a POMDP $\mathcal{S} = \left(S, s_0, A, (\tau_a)_{a \in A}, \sim\right)$ and a specification $\phi \subseteq S^\omega$ such that $\phi \in \mathcal{B}(S)$. The qualitative synthesis problem $(\mathcal{S}, \phi)$ demands the computation of a strategy $s : [S]^*_\sim \to A$ such that $\mu_s(\phi) = 1$.

POMDP - ω-regular Specification

$$\mathcal{S} = \left(S, s_0, A, (\tau_a)_{a \in A}, \sim\right)$$

POMDP - ω-regular Specification

$$\mathcal{S} = \left(S, s_0, A, (\tau_a)_{a \in A}, \sim\right)$$

- ω-regular specification

POMDP - ω-regular Specification

$$\mathcal{S} = \left(S, s_0, A, (\tau_a)_{a \in A}, \sim\right)$$

- ω-regular specification

$\Rightarrow$ DPA $\mathcal{P}$

POMDP - ω-regular Specification

$$\mathcal{S} = \left(S, s_0, A, (\tau_a)_{a \in A}, \sim\right)$$

- ω-regular specification

$\Rightarrow$ DPA $\mathcal{P}$

- natural product construction $\mathcal{S} \otimes \mathcal{P}$ yields POMDP with associated Parity-condition

POMDP - ω-regular Specification

$$\mathcal{S} = \left(S, s_0, A, (\tau_a)_{a \in A}, \sim\right)$$

- ω-regular specification

$\Rightarrow$ DPA $\mathcal{P}$

- natural product construction $\mathcal{S} \otimes \mathcal{P}$ yields POMDP with associated Parity-condition

Computing Strategies	Fully observable	Partially observable
Positive Büchi	PTIME	Undecidable
Almost-Sure Büchi	PTIME	EXPTIME
Positive Parity	PTIME	Undecidable
Almost-Sure Parity	PTIME	Undecidable

POMDP - PBA Specification

$$\mathcal{S} = \left(S, s_0, A, (\tau_a)_{a \in A}, \sim\right) \text{ and } \mathcal{P} = (Q, S, \delta, q_0, F)$$

POMDP - PBA Specification

$$\mathcal{S} = \left(S, s_0, A, (\tau_a)_{a \in A}, \sim\right) \text{ and } \mathcal{P} = (Q, S, \delta, q_0, F)$$

$$\alpha \in S^\omega \Rightarrow (Q, \mathcal{B}(Q), \mu_s)$$

POMDP - PBA Specification

$$\mathcal{S} = \left(S, s_0, A, (\tau_a)_{a\in A}, \sim\right) \text{ and } \mathcal{P} = (Q, S, \delta, q_0, F)$$

$$\alpha \in S^\omega \Rightarrow (Q, \mathcal{B}(Q), \mu_s)$$

Definition
Given $(\Omega_1, \mathcal{F}_1)$, $(\Omega_2, \mathcal{F}_2)$, then $K : \Omega_1 \times \mathcal{F}_2 \to [0,1]$ is a Markov-kernel if

1. $K(\cdot, A)$ is measurable in $\mathcal{F}_1$ for all $A \in \mathcal{F}_2$,
2. $K(\omega, \cdot)$ is a probability measure on $(\Omega_2, \mathcal{F}_2)$ for every $\omega \in \Omega_1$.

POMDP - PBA Specification

$$\mathcal{S} = \left(S, s_0, A, (\tau_a)_{a \in A}, \sim\right) \text{ and } \mathcal{P} = (Q, S, \delta, q_0, F)$$

$$K : S^\omega \times \mathcal{B}(Q) \to [0,1] \text{ with } K(\alpha, A) = \mu_\alpha(A)$$

POMDP - PBA Specification

$$\mathcal{S} = \left(S, s_0, A, (\tau_a)_{a \in A}, \sim\right) \text{ and } \mathcal{P} = (Q, S, \delta, q_0, F)$$

$$K : S^\omega \times \mathcal{B}(Q) \to [0,1] \text{ with } K(\alpha, A) = \mu_\alpha(A)$$

- for a strategy s: $\displaystyle\int_{\alpha \in S^\omega} K(\alpha, \mathrm{Acc}_{\text{Büchi}}(F)) d\mu_s = 1$ iff

  $\mu_\alpha(\mathrm{Acc}_{\text{Büchi}}(F)) = 1$ for almost all α

POMDP - PBA Specification

$$\mathcal{S} = \left(S, s_0, A, (\tau_a)_{a \in A}, \sim\right) \text{ and } \mathcal{P} = (Q, S, \delta, q_0, F)$$

$$K : S^\omega \times \mathcal{B}(Q) \to [0,1] \text{ with } K(\alpha, A) = \mu_\alpha(A)$$

- for a strategy s: $\int_{\alpha \in S^\omega} K(\alpha, \mathrm{Acc}_{\text{Büchi}}(F)) d\mu_s = 1$ iff $\mu_\alpha(\mathrm{Acc}_{\text{Büchi}}(F)) = 1$ for almost all α
- measure $\mu_s \otimes K$ on $(S^\omega \times Q^\omega, \mathcal{B}(S) \otimes \mathcal{B}(Q))$ uniquely determined by $\int_{\mathrm{cyl}(u)} K(\cdot, \mathrm{cyl}(v)) d\mu_s$

POMDP - PBA Specification

$$\mathcal{S} = \left(S, s_0, A, (\tau_a)_{a \in A}, \sim\right) \text{ and } \mathcal{P} = (Q, S, \delta, q_0, F)$$

$$K : S^\omega \times \mathcal{B}(Q) \to [0,1] \text{ with } K(\alpha, A) = \mu_\alpha(A)$$

- for a strategy s: $\int_{\alpha \in S^\omega} K(\alpha, \mathrm{Acc}_{\text{Büchi}}(F)) d\mu_s = 1$ iff $\mu_\alpha(\mathrm{Acc}_{\text{Büchi}}(F)) = 1$ for almost all α
- measure $\mu_s \otimes K$ on $(S^\omega \times Q^\omega, \mathcal{B}(S) \otimes \mathcal{B}(Q))$ uniquely determined by $\int_{\mathrm{cyl}(u)} K(\cdot, \mathrm{cyl}(v)) d\mu_s$
- $\int_{\mathrm{cyl}(u)} K(\cdot, \mathrm{cyl}(v)) d\mu_s = \mu_s(\mathrm{cyl}(u)) \cdot \mu_u(\mathrm{cyl}(v))$ for $|u| = |v|$

POMDP - PBA Specification

$$\mathcal{S} = \left(S, s_0, A, (\tau_a)_{a \in A}, \sim\right) \text{ and } \mathcal{P} = (Q, S, \delta, q_0, F)$$

$$K : S^\omega \times \mathcal{B}(Q) \to [0,1] \text{ with } K(\alpha, A) = \mu_\alpha(A)$$

- for a strategy s: $\displaystyle\int_{\alpha \in S^\omega} K(\alpha, \mathrm{Acc}_{\text{Büchi}}(F)) d\mu_s = 1$ iff $\mu_\alpha(\mathrm{Acc}_{\text{Büchi}}(F)) = 1$ for almost all α
- measure $\mu_s \otimes K$ on $(S^\omega \times Q^\omega, \mathcal{B}(S) \otimes \mathcal{B}(Q))$ uniquely determined by $\int_{\mathrm{cyl}(u)} K(\cdot, \mathrm{cyl}(v)) d\mu_s$
- $\int_{\mathrm{cyl}(u)} K(\cdot, \mathrm{cyl}(v)) d\mu_s = \mu_s(\mathrm{cyl}(u)) \cdot \mu_u(\mathrm{cyl}(v))$ for $|u| = |v|$

$\Rightarrow$ induces product construction for $\mathcal{S}$ and $\mathcal{P}$

POMDP - PBA Specification

$$\mathcal{S} = \left(S, s_0, A, (\tau_a)_{a \in A}, \sim\right) \text{ and } \mathcal{P} = (Q, S, \delta, q_0, F)$$

$$\mathcal{S} \otimes \mathcal{P} = \left(S \times Q, (s_0, q_0), A, \left(\tau'_a\right)_{a \in A}, \sim'\right)$$

POMDP - PBA Specification

$$\mathcal{S} = \left(S, s_0, A, (\tau_a)_{a \in A}, \sim\right) \text{ and } \mathcal{P} = (Q, S, \delta, q_0, F)$$

$$\mathcal{S} \otimes \mathcal{P} = \left(S \times Q, (s_0, q_0), A, \left(\tau'_a\right)_{a \in A}, \sim'\right)$$

- $\tau'_a((s, q), (z, p)) = \tau_a(s, z) \cdot \delta(p, s, q)$

POMDP - PBA Specification

$$\mathcal{S} = \left(S, s_0, A, (\tau_a)_{a \in A}, \sim\right) \text{ and } \mathcal{P} = (Q, S, \delta, q_0, F)$$

$$\mathcal{S} \otimes \mathcal{P} = \left(S \times Q, (s_0, q_0), A, \left(\tau'_a\right)_{a \in A}, \sim'\right)$$

- $\tau'_a((s, q), (z, p)) = \tau_a(s, z) \cdot \delta(p, s, q)$
- $(s, q) \sim' (z, p)$ iff $s \sim z$

POMDP - PBA Specification

$$\mathcal{S} = \left(S, s_0, A, (\tau_a)_{a\in A}, \sim\right) \text{ and } \mathcal{P} = (Q, S, \delta, q_0, F)$$

$$\mathcal{S} \otimes \mathcal{P} = \left(S \times Q, (s_0, q_0), A, \left(\tau'_a\right)_{a\in A}, \sim'\right)$$

- $\tau'_a((s,q),(z,p)) = \tau_a(s,z) \cdot \delta(p,s,q)$
- $(s,q) \sim' (z,p)$ iff $s \sim z$

$\Rightarrow$ strategies can be translated from $\mathcal{S}$ to $\mathcal{S} \otimes \mathcal{P}$

POMDP - PBA Specification

$$\mathcal{S} = \left(S, s_0, A, (\tau_a)_{a \in A}, \sim\right) \text{ and } \mathcal{P} = (Q, S, \delta, q_0, F)$$

$$\mathcal{S} \otimes \mathcal{P} = \left(S \times Q, (s_0, q_0), A, \left(\tau'_a\right)_{a \in A}, \sim'\right)$$

- $\tau'_a((s, q), (z, p)) = \tau_a(s, z) \cdot \delta(p, s, q)$
- $(s, q) \sim' (z, p)$ iff $s \sim z$

$\Rightarrow$ strategies can be translated from $\mathcal{S}$ to $\mathcal{S} \otimes \mathcal{P}$

$\Rightarrow$ $\mu_s(\mathrm{Acc}_{\text{Büchi}}(S \times F)) = \int_{S^\omega} K(\cdot, \mathrm{Acc}_{\text{Büchi}}(F)) d\mu_s$

POMDP - PBA Specification

$$\mathcal{S} = \left(S, s_0, A, (\tau_a)_{a \in A}, \sim\right) \text{ and } \mathcal{P} = (Q, S, \delta, q_0, F)$$

$$\mathcal{S} \otimes \mathcal{P} = \left(S \times Q, (s_0, q_0), A, \left(\tau'_a\right)_{a \in A}, \sim'\right)$$

- $\tau'_a((s, q), (z, p)) = \tau_a(s, z) \cdot \delta(p, s, q)$
- $(s, q) \sim' (z, p)$ iff $s \sim z$

$\Rightarrow$ strategies can be translated from $\mathcal{S}$ to $\mathcal{S} \otimes \mathcal{P}$

$\Rightarrow$ $\mu_s(\mathrm{Acc}_{\text{Büchi}}(S \times F)) = \int_{S^\omega} K(\cdot, \mathrm{Acc}_{\text{Büchi}}(F)) d\mu_s$

$\Rightarrow$ almost-surely accepting strategy in $\mathcal{S} \otimes \mathcal{P}$ is qualitatively "good" for $\mathcal{S}$ with specification $\mathcal{P}$

Conclusion

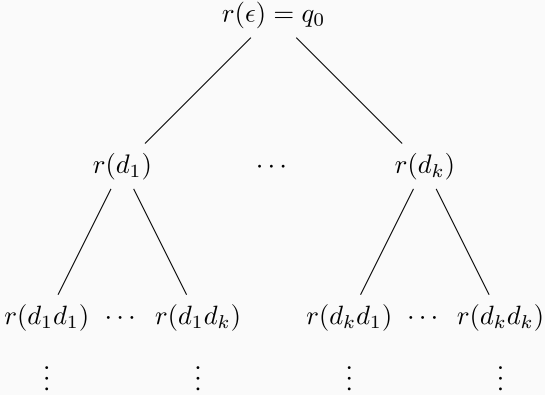
Conclusion & Ideas

- Introduction of WDTAs
- Synthesis for antagonistic environment for almost-sure accepting PBAs in doubly exponential time
- Synthesis for POMDP for almost-sure accepting PBAs in exponential time

* Co-operative POSG as emptiness games for unrestricted WDTAs
* Strategy computation for $\textsc{Eve}$ in equally informed POSG

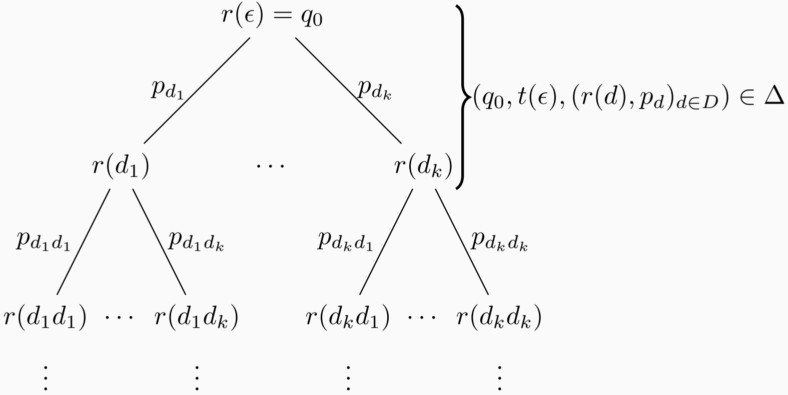
Weighted Descent Tree Automata - Idea

- non-deterministically construct run
- weight individual paths
- acceptance by measure



Weighted Descent Tree Automata - Idea

- non-deterministically construct run
- weight individual paths
- acceptance by measure



Weighted Descent Tree Automata - Idea

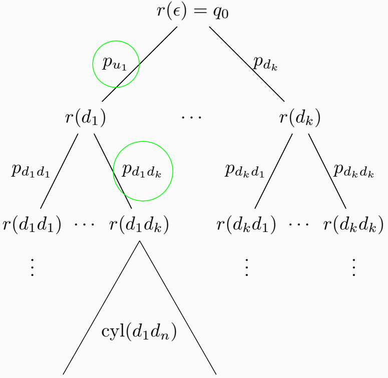
- non-deterministically construct run
- weight individual paths
- acceptance by measure

$$\mu_r(\mathsf{cyl}(u_1 \dots u_n)) = \prod_{1 \le i \le n} p_{u_1 \dots u_i}$$

$$\Rightarrow (D^\omega, \mathcal{B}(D), \mu_r)$$

with

$$\mu_r(\{\rho \in D^\omega \mid r(\rho) \in \mathsf{Acc}\})$$



Weighted Descent Tree Automata - Definition

$$\mathcal{A} = (Q, q_0, D, \Sigma, \Delta, \mathsf{Acc})$$

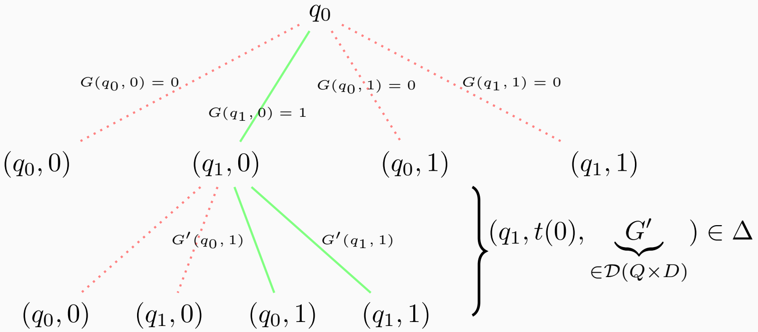
Weighted Descent Tree Automata - Definition

$$\mathcal{A} = (Q, q_0, D, \Sigma, \Delta, \mathsf{Acc})$$

- Q: finite state set
- $q_0 \in Q$: initial state
- D: finite set of directions
- Σ: finite alphabet
- Δ: transitions of the form (q, σ, G)
- Acc: accepted language

Weighted Descent Tree Automata - Definition

$$\mathcal{A} = (Q, q_0, D, \Sigma, \Delta, \mathsf{Acc})$$

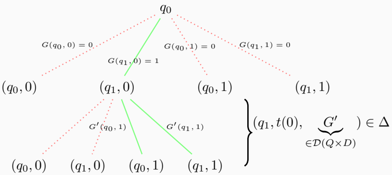


Weighted Descent Tree Automata - Definition

$$\mathcal{A} = (Q, q_0, D, \Sigma, \Delta, \mathrm{Acc})$$

- run: $Q \times D$-ary $\mathcal{G}(\mathcal{A})$-tree

$\Rightarrow \left((Q \times D)^{\omega}, \mathcal{B}(Q \times D), \mu_r\right)$



$$\mathrm{Acc}\,[Q] = \{(d_1, q_1)\,(d_2, q_2) \cdots \in (D \times Q)^{\omega} \mid q_1 q_2 \cdots \in \mathrm{Acc}\}$$

and

$$\mu_r(\mathrm{Acc}\,[Q]) > 0 \text{ or } \mu_r(\mathrm{Acc}\,[Q]) = 1$$

Tree Automata - Example

$$\mathcal{L} = \left\{ t : \{0,1\}^* \to \{a,b\} \mid {}^{a \notin \mathrm{Inf}(t(\epsilon)t(\alpha_1)t(\alpha_1\alpha_2)\ldots)}_{\text{for all } \alpha_1\alpha_2\cdots\in\{0,1\}^\omega} \right\}$$

- Parity-condition ✓
- Büchi-condition ✗

$$\mathcal{A} = (\{q_a, q_b\}, q_a, \{0,1\}, \{a,b\}, \Delta, \{q_a \mapsto 1, q_b \mapsto 0\})$$

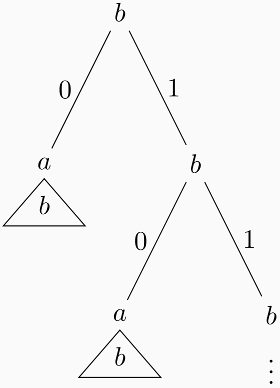
with

$$\Delta = \left\{ (q, \sigma, q_\sigma, q_\sigma) : {}^{q \in Q,}_{\sigma \in \{a,b\}} \right\}$$

Tree Automata - Example

$$\mathcal{L} = \left\{ t : \{0,1\}^* \to \{a,b\} \mid {a \notin \mathrm{Inf}(t(\epsilon)t(\alpha_1)t(\alpha_1\alpha_2)\dots) \atop \text{for all } \alpha_1\alpha_2\dots \in \{0,1\}^\omega} \right\}$$

- Parity-condition ✓
- Büchi-condition ✗



Antagonistic Environment - PBA Specification

$$(Q, J \times I, \delta, q_0, F) \Rightarrow \left(S, s_0, E, A, (\tau_{e,a})_{e \in E, a \in A}, \sim_E, \sim_A, F'\right)$$

- Partially Observable Stochastic Game (POSG):
- POMDP with antagonistic players ($\textsc{Eve}$ & $\textsc{Adam}$)
- $G = \left(S, s_0, E, A, (\tau_{e,a})_{e \in E, a \in A}, \sim_E, \sim_A, F'\right)$
- $\begin{array}{r} \textsc{Eve}\ f : [S]^*_{\sim_A} \to A \\ \textsc{Adam}\ g : [S]^*_{\sim_E} \to E \end{array} \Rightarrow (S^\omega, \mathcal{B}(S), \mu_{f,g})$
- f almost-surely wins for $\textsc{Eve}$ if $\mu_{f,g}(\mathrm{Acc}_{\text{Büchi}}(F)) = 1$ for all g

Antagonistic Environment - PBA Specification

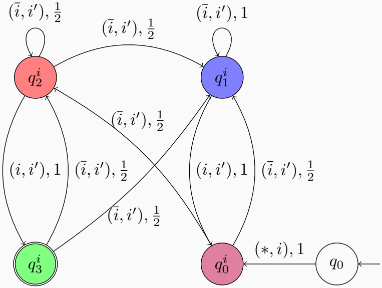
$$(Q, J \times I, \delta, q_0, F) \Rightarrow \left(S, s_0, E, A, (\tau_{e,a})_{e \in E, a \in A}, \sim_E, \sim_A, F'\right)$$

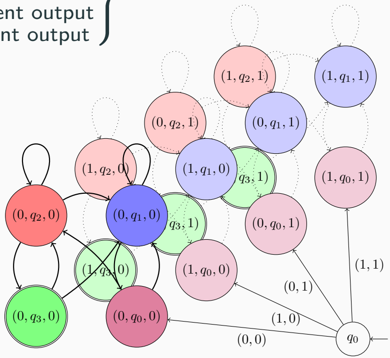
- "input-/output-game with unobservable stochastic background"
- $S = I \times Q \times J \cup \{q_0\}$ with $s_0 = q_0$
- $E = J$, $A = I$
- $\tau_{j,i}((a,q,e),(i',p,j')) = \begin{cases} \delta(q,(i,j),p) & \text{if } i' = i, j' = j, \\ 0 & \text{otherwise.} \end{cases}$
- $\sim_E = \sim_A = \left\{ \substack{((i,q,j),(i,p,j)): \\ i \in I, j \in J, q,p \in Q} \right\}$
- $F' = I \times F \times J$

Antagonistic Environment - PBA Specification

$$(Q, J \times I, \delta, q_0, F) \Rightarrow \left(S, s_0, E, A, (\tau_{e,a})_{e\in E, a\in A}, \sim_E, \sim_A, F'\right)$$

$$\left\{ a^{k_1} b a^{k_2} b \cdots : \begin{array}{c} k_i>0 \text{ for all } i>0 \\ \prod_{i>0}\left(1-\frac{1}{2}^{k_i}\right)=0 \\ a\text{:last input differs from current output} \\ b\text{:last input is equal to current output} \end{array} \right\}$$





POMDP - PBA Specification

$$\mathcal{S} = \left(S, s_0, A, (\tau_a)_{a \in A}, \sim\right) \text{ and } \mathcal{P} = (Q, S, \delta, q_0, F)$$

$$\alpha \in S^\omega \Rightarrow (Q, \mathcal{B}(Q), \mu_s)$$

Definition
Given $(\Omega_1, \mathcal{F}_1)$, $(\Omega_2, \mathcal{F}_2)$, then $K : \Omega_1 \times \mathcal{F}_2 \to [0,1]$ is a Markov-kernel if

1. $K(\cdot, A)$ is measurable in $\mathcal{F}_1$ for all $A \in \mathcal{F}_2$,
2. $K(\omega, \cdot)$ is a probability measure on $(\Omega_2, \mathcal{F}_2)$ for every $\omega \in \Omega_1$.

POMDP - PBA Specification

$$\mathcal{S} = \left(S, s_0, A, (\tau_a)_{a \in A}, \sim\right) \text{ and } \mathcal{P} = (Q, S, \delta, q_0, F)$$

$$K : S^\omega \times \mathcal{B}(Q) \to [0,1] \text{ with } K(\alpha, A) = \mu_\alpha(A)$$

- for a strategy s: $\displaystyle\int_{\alpha \in S^\omega} K(\alpha, \mathsf{Acc}_{\text{Büchi}}(Q)) d\mu_s = 1$ iff

  $\mu_\alpha(\mathsf{Acc}_{\text{Büchi}}(Q)) = 1$ for almost all α.
- measure $\mu_s \otimes K$ on $(S^\omega \times Q^\omega, \mathcal{B}(S) \otimes \mathcal{B}(Q))$ uniquely determined by $\int_{\mathsf{cyl}(u)} K(\cdot, \mathsf{cyl}(v)) d\mu_s$
- $\int_{\mathsf{cyl}(u)} K(\cdot, \mathsf{cyl}(v)) d\mu_s = \mu_s(\mathsf{cyl}(u)) \cdot \mu_u(\mathsf{cyl}(v))$ for $|u| = |v|$

$\Rightarrow$ induces product construction for $\mathcal{S}$ and $\mathcal{P}$

POMDP - PBA Specification

$$\mathcal{S} = \left(S, s_0, A, (\tau_a)_{a \in A}, \sim\right) \text{ and } \mathcal{P} = (Q, S, \delta, q_0, F)$$

$$\mathcal{S} \otimes \mathcal{P} = \left(S \times Q, (q_0, s_0), A, \left(\tau'_a\right)_{a \in A}, \sim'\right)$$

- $\tau'_a((s, q), (z, p)) = \tau_a(s, z) \cdot \delta(p, s, q)$
- $(s, q) \sim' (z, p)$ iff $s \sim z$

$\Rightarrow$ strategies can be translated from $\mathcal{S}$ to $\mathcal{S} \otimes \mathcal{P}$

$\Rightarrow$ $\mu_s(\mathrm{Acc}_{\text{Büchi}}(S \times F)) = \int_{S^\omega} K(\cdot, \mathrm{Acc}_{\text{Büchi}}(F)) d\mu_s$

$\Rightarrow$ almost-surely accepting strategy in $\mathcal{S} \otimes \mathcal{P}$ is qualitatively "good" for $\mathcal{S}$ with specification $\mathcal{P}$