Modeling and Verification of Probabilistic Systems

Joost-Pieter Katoen

Lehrstuhl für Informatik 2
Software Modeling and Verification Group

http://moves.rwth-aachen.de/teaching/ws-1516/movep15/

November 4, 2015

Overview

Summary of previous lectures

Reachability probabilities

Can be obtained as a unique solution of a linear equation system.

Summary of previous lectures

Reachability probabilities

Can be obtained as a unique solution of a linear equation system.

Reachability probabilities are pivotal

1. Repeated reachability
   - = Reachability of the BSCCs containing a goal state
2. Persistence
   - = Reachability of the BSCCs only containing goal states

Aim of this lecture

Reachability probabilities = key to determine the probability of any ω-regular property. (This includes all linear temporal logic formulas.)

Aim of this lecture

Reachability probabilities = key to determine the probability of any ω-regular property. (This includes all linear temporal logic formulas.)

Major steps for Markov chain $\mathcal{D}$

1. Consider first a simple class of properties: regular safety properties.

Aim of this lecture

Reachability probabilities = key to determine the probability of any ω-regular property. (This includes all linear temporal logic formulas.)

Major steps for Markov chain $\mathcal{D}$

1. Consider first a simple class of properties: regular safety properties.
2. All traces refuting such property P are recognized by a deterministic finite-state automaton $\mathcal{A}$.

Aim of this lecture

Reachability probabilities = key to determine the probability of any ω-regular property. (This includes all linear temporal logic formulas.)

Major steps for Markov chain $\mathcal{D}$

1. Consider first a simple class of properties: regular safety properties.
2. All traces refuting such property P are recognized by a deterministic finite-state automaton $\mathcal{A}$.
3. Probability of P = reachability probability in a product of $\mathcal{D}$ and $\mathcal{A}$.

Aim of this lecture

Reachability probabilities = key to determine the probability of any ω-regular property. (This includes all linear temporal logic formulas.)

Major steps for Markov chain $\mathcal{D}$

1. Consider first a simple class of properties: regular safety properties.
2. All traces refuting such property P are recognized by a deterministic finite-state automaton $\mathcal{A}$.
3. Probability of P = reachability probability in a product of $\mathcal{D}$ and $\mathcal{A}$.
4. What are ω-regular properties?
5. All traces satisfying such property P are recognized by a deterministic Rabin automaton $\mathcal{A}$.

Aim of this lecture

Reachability probabilities = key to determine the probability of any ω-regular property. (This includes all linear temporal logic formulas.)

Major steps for Markov chain $\mathcal{D}$

1. Consider first a simple class of properties: regular safety properties.
2. All traces refuting such property P are recognized by a deterministic finite-state automaton $\mathcal{A}$.
3. Probability of P = reachability probability in a product of $\mathcal{D}$ and $\mathcal{A}$.
4. What are ω-regular properties?
5. All traces satisfying such property P are recognized by a deterministic Rabin automaton $\mathcal{A}$.
6. Probability of P = reachability probability in a product of $\mathcal{D}$ and $\mathcal{A}$.

Overview

Paths and traces

Paths

A *path* in DTMC $\mathcal{D}$ is an infinite sequence of states $s_0 s_1 s_2 \ldots\ldots$ with $\mathbf{P}(s_i, s_{i+1}) > 0$ for all i.

Paths and traces

Paths

A *path* in DTMC $\mathcal{D}$ is an infinite sequence of states $s_0 s_1 s_2 \ldots\ldots$ with $\mathbf{P}(s_i, s_{i+1}) > 0$ for all i.

Let $Paths(\mathcal{D})$ denote the set of paths in $\mathcal{D}$, and $Paths^*(\mathcal{D})$ the set of finite prefixes thereof.

Paths and traces

Paths

A *path* in DTMC $\mathcal{D}$ is an infinite sequence of states $s_0 s_1 s_2 \ldots\ldots$ with $\mathbf{P}(s_i, s_{i+1}) > 0$ for all i.

Let $Paths(\mathcal{D})$ denote the set of paths in $\mathcal{D}$, and $Paths^*(\mathcal{D})$ the set of finite prefixes thereof.

Traces

The *trace* of path $\pi = s_0\, s_1\, s_2 \ldots$ is $trace(\pi) = L(s_0)\, L(s_1)\, L(s_2) \ldots$.

Paths and traces

Paths

A *path* in DTMC $\mathcal{D}$ is an infinite sequence of states $s_0 s_1 s_2 \ldots \ldots$ with $\mathbf{P}(s_i, s_{i+1}) > 0$ for all i.

Let $Paths(\mathcal{D})$ denote the set of paths in $\mathcal{D}$, and $Paths^*(\mathcal{D})$ the set of finite prefixes thereof.

Traces

The *trace* of path $\pi = s_0\, s_1\, s_2 \ldots$ is $trace(\pi) = L(s_0)\, L(s_1)\, L(s_2) \ldots$.
The trace of finite path $\widehat{\pi} = s_0\, s_1 \ldots s_n$ is $trace(\widehat{\pi}) = L(s_0)\, L(s_1) \ldots L(s_n)$.

Paths and traces

Paths

A *path* in DTMC $\mathcal{D}$ is an infinite sequence of states $s_0 s_1 s_2 \ldots \ldots$ with $\mathbf{P}(s_i, s_{i+1}) > 0$ for all i.

Let $Paths(\mathcal{D})$ denote the set of paths in $\mathcal{D}$, and $Paths^*(\mathcal{D})$ the set of finite prefixes thereof.

Traces

The *trace* of path $\pi = s_0\, s_1\, s_2 \ldots$ is $trace(\pi) = L(s_0)\, L(s_1)\, L(s_2) \ldots$.
The trace of finite path $\widehat{\pi} = s_0\, s_1 \ldots s_n$ is $trace(\widehat{\pi}) = L(s_0)\, L(s_1) \ldots L(s_n)$.

The *set of traces* of a set Π of paths: $trace(\Pi) = \{\, trace(\pi) \mid \pi \in \Pi \,\}$.

Paths and traces

Paths

A *path* in DTMC $\mathcal{D}$ is an infinite sequence of states $s_0 s_1 s_2 \ldots \ldots$ with $\mathbf{P}(s_i, s_{i+1}) > 0$ for all i.

Let $Paths(\mathcal{D})$ denote the set of paths in $\mathcal{D}$, and $Paths^*(\mathcal{D})$ the set of finite prefixes thereof.

Traces

The *trace* of path $\pi = s_0\, s_1\, s_2 \ldots$ is $trace(\pi) = L(s_0)\, L(s_1)\, L(s_2) \ldots$.
The trace of finite path $\widehat{\pi} = s_0\, s_1 \ldots s_n$ is $trace(\widehat{\pi}) = L(s_0)\, L(s_1) \ldots L(s_n)$.

The *set of traces* of a set Π of paths: $trace(\Pi) = \{\, trace(\pi) \mid \pi \in \Pi \,\}$.

LT properties

Linear-time property

A *linear-time property* (LT property) over AP is a subset of $\left(2^{AP}\right)^{\omega}$.

LT properties

Linear-time property

A *linear-time property* (LT property) over AP is a subset of $\left(2^{AP}\right)^{\omega}$. An LT-property is thus a set of infinite traces over 2^{AP}.

LT properties

Linear-time property

A *linear-time property* (LT property) over AP is a subset of $\left(2^{AP}\right)^{\omega}$. An LT-property is thus a set of infinite traces over 2^{AP}.

Intuition

An LT-property gives the admissible behaviours of the DTMC at hand.

Probability of LT properties

Probability of LT properties

The *probability* for DTMC $\mathcal{D}$ to exhibit a trace in property P (over AP) is:

$$Pr^{\mathcal{D}}(P) \; = \; Pr^{\mathcal{D}}\{\, \pi \in Paths(\mathcal{D}) \mid trace(\pi) \in P \,\}.$$

Probability of LT properties

Probability of LT properties

The *probability* for DTMC $\mathcal{D}$ to exhibit a trace in property P (over AP) is:

$$Pr^{\mathcal{D}}(P) \; = \; Pr^{\mathcal{D}}\{ \, \pi \in Paths(\mathcal{D}) \mid trace(\pi) \in P \, \}.$$

For state s in $\mathcal{D}$, let $Pr(s \models P) \; = \; Pr_s\{ \, \pi \in Paths(s) \mid trace(\pi) \in P \, \}$.

Probability of LT properties

Probability of LT properties

The *probability* for DTMC $\mathcal{D}$ to exhibit a trace in property P (over AP) is:

$$Pr^{\mathcal{D}}(P) \; = \; Pr^{\mathcal{D}}\{\, \pi \in Paths(\mathcal{D}) \mid trace(\pi) \in P \,\}.$$

For state s in $\mathcal{D}$, let $Pr(s \models P) \; = \; Pr_s\{\, \pi \in Paths(s) \mid trace(\pi) \in P \,\}$.

We do not address measurability of P yet. We will later identify a rich set P of LT-properties—those that include all LTL formulas—for which the set of paths $\{\, \pi \in Paths(\mathcal{D}) \mid trace(\pi) \in P \,\}$ is measurable.

Safety properties

Safety properties

Safety property

LT property P_{safe} over AP is a *safety property* if for all $\sigma \in (2^{AP})^\omega \setminus P_{safe}$ there exists a finite prefix $\widehat{\sigma}$ of σ such that:

$$P_{safe} \cap \underbrace{\left\{\sigma' \in (2^{AP})^\omega \mid \widehat{\sigma} \text{ is a prefix of } \sigma'\right\}}_{\text{all possible extensions of } \widehat{\sigma}} = \varnothing.$$

Safety properties

Safety property

LT property P_{safe} over AP is a *safety property* if for all $\sigma \in \left(2^{AP}\right)^{\omega} \setminus P_{safe}$ there exists a finite prefix $\widehat{\sigma}$ of σ such that:

$$P_{safe} \cap \underbrace{\left\{ \sigma' \in \left(2^{AP}\right)^{\omega} \mid \widehat{\sigma} \text{ is a prefix of } \sigma' \right\}}_{\text{all possible extensions of } \widehat{\sigma}} \; = \; \varnothing.$$

Any such finite word $\widehat{\sigma}$ is called a *bad prefix* for P_{safe}.

Safety properties

Safety property

LT property P_{safe} over AP is a *safety property* if for all $\sigma \in (2^{AP})^\omega \setminus P_{safe}$ there exists a finite prefix $\widehat{\sigma}$ of σ such that:

$$P_{safe} \cap \underbrace{\left\{ \sigma' \in (2^{AP})^\omega \mid \widehat{\sigma} \text{ is a prefix of } \sigma' \right\}}_{\text{all possible extensions of } \widehat{\sigma}} = \varnothing.$$

Any such finite word $\widehat{\sigma}$ is called a *bad prefix* for P_{safe}.

Regular safety property

A safety property is *regular* if its set of bad prefixes constitutes a regular language (over the alphabet 2^{AP}).

Safety properties

Safety property

LT property P_{safe} over AP is a *safety property* if for all $\sigma \in (2^{AP})^\omega \setminus P_{safe}$ there exists a finite prefix $\widehat{\sigma}$ of σ such that:

$$P_{safe} \cap \underbrace{\left\{\sigma' \in (2^{AP})^\omega \mid \widehat{\sigma} \text{ is a prefix of } \sigma'\right\}}_{\text{all possible extensions of } \widehat{\sigma}} = \varnothing.$$

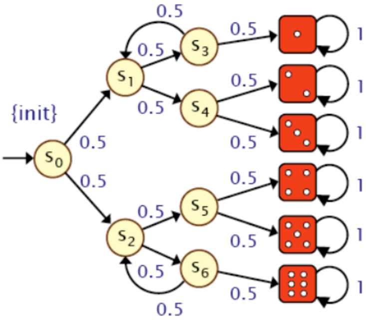
Any such finite word $\widehat{\sigma}$ is called a *bad prefix* for P_{safe}.

Regular safety property

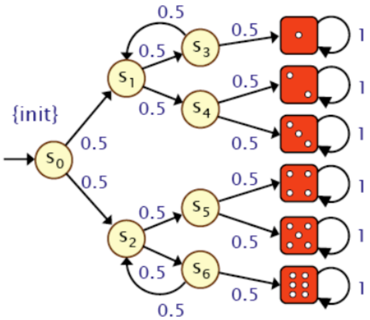
A safety property is *regular* if its set of bad prefixes constitutes a regular language (over the alphabet 2^{AP}). Thus, the set of all bad prefixes of a regular safety property can be represented by a finite-state automaton.

Property of Knuth's die

Property of Knuth's die



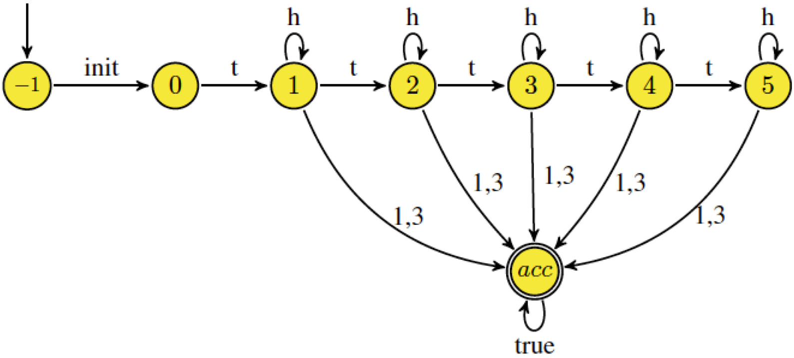
Property of Knuth's die



Property of Knuth's die

After initial tails, yield 1 or 3 but with maximally five time tails.

Property as an automaton



After initial tails, yield 1 or 3 but with at most five times tails in total

Overview

Probability of a regular safety property

Let $\mathcal{A} = (Q, 2^{AP}, \delta, q_0, F)$ be a deterministic finite-state automaton (DFA) for the bad prefixes of regular safety property P_{safe}:

$$P_{safe} \;=\; \{\, A_0\, A_1\, A_2 \ldots \in \left(2^{AP}\right)^{\omega} \mid \forall n \geqslant 0.\, A_0\, A_1 \ldots A_n \notin \mathcal{L}(\mathcal{A}) \,\}.$$

Probability of a regular safety property

Let $\mathcal{A} = (Q, 2^{AP}, \delta, q_0, F)$ be a deterministic finite-state automaton (DFA) for the bad prefixes of regular safety property P_{safe}:

$$P_{safe} = \{ A_0 A_1 A_2 \ldots \in \left(2^{AP}\right)^{\omega} \mid \forall n \geqslant 0. A_0 A_1 \ldots A_n \notin \mathcal{L}(\mathcal{A}) \}.$$

Let δ be total, i.e., $\delta(q, A)$ is defined for each $A \subseteq AP$ and state $q \in Q$.

Probability of a regular safety property

Let $\mathcal{A} = (Q, 2^{AP}, \delta, q_0, F)$ be a deterministic finite-state automaton (DFA) for the bad prefixes of regular safety property P_{safe}:

$$P_{safe} = \{ A_0 A_1 A_2 \ldots \in \left(2^{AP}\right)^{\omega} \mid \forall n \geqslant 0.\, A_0 A_1 \ldots A_n \notin \mathcal{L}(\mathcal{A}) \}.$$

Let δ be total, i.e., $\delta(q, A)$ is defined for each $A \subseteq AP$ and state $q \in Q$.
Furthermore, let $\mathcal{D} = (S, \mathbf{P}, \iota_{\mathrm{init}}, AP, L)$ be a finite DTMC.

Probability of a regular safety property

Let $\mathcal{A} = (Q, 2^{AP}, \delta, q_0, F)$ be a deterministic finite-state automaton (DFA) for the bad prefixes of regular safety property P_{safe}:

$$P_{safe} \ = \ \{ \, A_0 \, A_1 \, A_2 \ldots \in \left(2^{AP}\right)^{\omega} \mid \forall n \geqslant 0. \, A_0 \, A_1 \ldots A_n \notin \mathcal{L}(\mathcal{A}) \, \}.$$

Let δ be total, i.e., $\delta(q, A)$ is defined for each $A \subseteq AP$ and state $q \in Q$. Furthermore, let $\mathcal{D} = (S, \mathbf{P}, \iota_{\text{init}}, AP, L)$ be a finite DTMC. Our interest is to compute the probability

$$Pr^{\mathcal{D}}(P_{safe}) \ = \ 1 - \sum_{s \in S} \iota_{\text{init}}(s) \cdot Pr(s \models \mathcal{A}) \quad \text{where}$$

Probability of a regular safety property

Let $\mathcal{A} = (Q, 2^{AP}, \delta, q_0, F)$ be a deterministic finite-state automaton (DFA) for the bad prefixes of regular safety property P_{safe}:

$$P_{safe} \;=\; \{\, A_0\, A_1\, A_2 \ldots \in \left(2^{AP}\right)^{\omega} \mid \forall n \geqslant 0.\, A_0\, A_1 \ldots A_n \notin \mathcal{L}(\mathcal{A}) \,\}.$$

Let δ be total, i.e., $\delta(q, A)$ is defined for each $A \subseteq AP$ and state $q \in Q$. Furthermore, let $\mathcal{D} = (S, \mathbf{P}, \iota_{\text{init}}, AP, L)$ be a finite DTMC. Our interest is to compute the probability

$$Pr^{\mathcal{D}}(P_{safe}) \;=\; 1 \,-\, \sum_{s \in S} \iota_{\text{init}}(s) \cdot Pr(s \models \mathcal{A}) \quad \text{where}$$

$$Pr(s \models \mathcal{A}) \quad = \quad Pr_s^{\mathcal{D}}\{\, \pi \in Paths(s) \;\mid\; trace(\pi) \notin P_{safe} \,\}.$$

Probability of a regular safety property

Let $\mathcal{A} = (Q, 2^{AP}, \delta, q_0, F)$ be a deterministic finite-state automaton (DFA) for the bad prefixes of regular safety property P_{safe}:

$$P_{safe} = \{ A_0 A_1 A_2 \ldots \in \left(2^{AP}\right)^{\omega} \mid \forall n \geqslant 0.\, A_0 A_1 \ldots A_n \notin \mathcal{L}(\mathcal{A}) \}.$$

Let δ be total, i.e., $\delta(q, A)$ is defined for each $A \subseteq AP$ and state $q \in Q$. Furthermore, let $\mathcal{D} = (S, \mathbf{P}, \iota_{\mathrm{init}}, AP, L)$ be a finite DTMC. Our interest is to compute the probability

$$Pr^{\mathcal{D}}(P_{safe}) \;=\; 1 - \sum_{s \in S} \iota_{\mathrm{init}}(s) \cdot Pr(s \models \mathcal{A}) \quad \text{where}$$

$$Pr(s \models \mathcal{A}) \;=\; Pr_s^{\mathcal{D}}\{ \pi \in Paths(s) \mid trace(\pi) \notin P_{safe} \}.$$

These probabilities can be obtained by considering a product of DTMC $\mathcal{D}$ with DFA $\mathcal{A}$.

Probability of a regular safety property

$$Pr^{\mathcal{D}}(P_{safe}) \; = \; 1 - \sum_{s \in S} \iota_{\text{init}}(s) \cdot Pr(s \models \mathcal{A}) \quad \text{where}$$

Probability of a regular safety property

$$Pr^{\mathcal{D}}(P_{safe}) \; = \; 1 - \sum_{s \in S} \iota_{\text{init}}(s) \cdot Pr(s \models \mathcal{A}) \quad \text{where}$$

$$Pr(s \models \mathcal{A}) \quad = \quad Pr_s^{\mathcal{D}}\{ \, \pi \in Paths(s) \; | \; trace(\pi) \notin P_{safe} \, \}.$$

Probability of a regular safety property

$$Pr^{\mathcal{D}}(P_{safe}) \;=\; 1 - \sum_{s \in S} \iota_{\mathrm{init}}(s) \cdot Pr(s \models \mathcal{A}) \quad \text{where}$$

$$Pr(s \models \mathcal{A}) \;=\; Pr_s^{\mathcal{D}}\{\, \pi \in Paths(s) \mid trace(\pi) \notin P_{safe} \,\}.$$

Remark

The value $Pr(s \models \mathcal{A})$ can be written as the (possibly infinite) sum:

$$Pr(s \models \mathcal{A}) \;=\; \sum_{\widehat{\pi}} \mathbf{P}(\widehat{\pi})$$

where $\widehat{\pi}$ ranges over all finite path prefixes $s_0\, s_1 \ldots s_n$ with $s_0 = s$

Probability of a regular safety property

$$Pr^{\mathcal{D}}(P_{safe}) \;=\; 1 - \sum_{s \in S} \iota_{\text{init}}(s) \cdot Pr(s \models \mathcal{A}) \quad \text{where}$$

$$Pr(s \models \mathcal{A}) \;\;=\;\; Pr_s^{\mathcal{D}}\{\, \pi \in Paths(s) \mid trace(\pi) \notin P_{safe} \,\}.$$

Remark

The value $Pr(s \models \mathcal{A})$ can be written as the (possibly infinite) sum:

$$Pr(s \models \mathcal{A}) \;=\; \sum_{\widehat{\pi}} \mathbf{P}(\widehat{\pi})$$

where $\widehat{\pi}$ ranges over all finite path prefixes $s_0\, s_1 \ldots s_n$ with $s_0 = s$ and:

1. $trace(s_0\, s_1 \ldots s_n) \;=\; L(s_0)\, L(s_1) \ldots L(s_n) \;\in \mathcal{L}(\mathcal{A})$, and

Probability of a regular safety property

$$Pr^{\mathcal{D}}(P_{safe}) \;=\; 1 - \sum_{s \in S} \iota_{\text{init}}(s) \cdot Pr(s \models \mathcal{A}) \quad \text{where}$$

$$Pr(s \models \mathcal{A}) \;=\; Pr_s^{\mathcal{D}}\{\, \pi \in Paths(s) \mid trace(\pi) \notin P_{safe} \,\}.$$

Remark

The value $Pr(s \models \mathcal{A})$ can be written as the (possibly infinite) sum:

$$Pr(s \models \mathcal{A}) \;=\; \sum_{\widehat{\pi}} \mathbf{P}(\widehat{\pi})$$

where $\widehat{\pi}$ ranges over all finite path prefixes $s_0\, s_1 \ldots s_n$ with $s_0 = s$ and:

1. $trace(s_0\, s_1 \ldots s_n) \;=\; L(s_0)\, L(s_1) \ldots L(s_n) \;\in \mathcal{L}(\mathcal{A})$, and
2. the length of $\widehat{\pi}$ is minimal, i.e., $trace(s_0\, s_1 \ldots s_i) \notin \mathcal{L}(\mathcal{A})$ for all $0 \leqslant i < n$.

Product construction: intuition

DTMC $\mathcal{D}$
with state space S

DRA $\mathcal{A}$
with state space Q

$$
\begin{array}{ccc}
s_0 & \qquad L(s_0)=A_0 \\
\downarrow & \\
s_1 & \qquad L(s_1)=A_1 \\
\downarrow & \\
s_2 & \qquad L(s_2)=A_2 \\
\downarrow & \\
\vdots & \qquad \vdots \\
\downarrow & \\
s_n & \qquad L(s_n)=A_n
\end{array}
$$

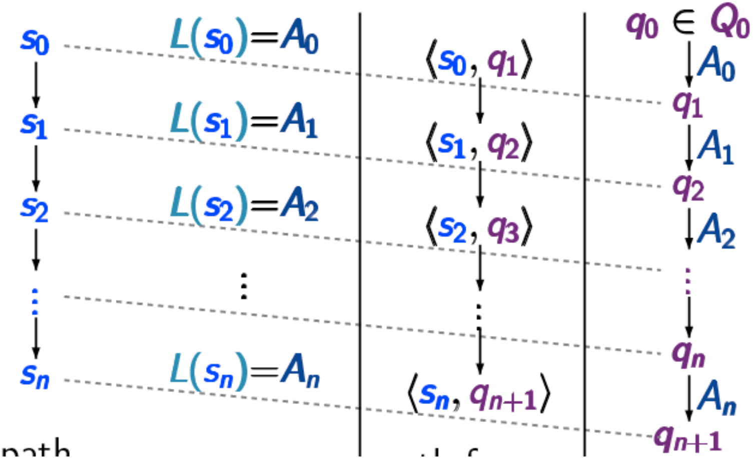
path

$$
\begin{array}{c}
q_0 \in Q_0 \\
\downarrow A_0 \\
q_1 \\
\downarrow A_1 \\
q_2 \\
\downarrow A_2 \\
\vdots \\
\downarrow \\
q_n \\
\downarrow A_n \\
q_{n+1}
\end{array}
$$

Product construction: intuition

DTMC $\mathcal{D}$ with state space S

DRA $\mathcal{A}$ with state space Q



product $\mathcal{D} \otimes \mathcal{A}$

Product Markov chain

Product Markov chain

Product Markov chain

Let $\mathcal{D} = (S, \mathbf{P}, \iota_{\mathrm{init}}, AP, L)$ be a DTMC and $\mathcal{A} = (Q, 2^{AP}, \delta, q_0, F)$ be a DFA.

Product Markov chain

Product Markov chain

Let $\mathcal{D} = (S, \mathbf{P}, \iota_{\text{init}}, AP, L)$ be a DTMC and $\mathcal{A} = (Q, 2^{AP}, \delta, q_0, F)$ be a DFA. The *product* $\mathcal{D} \otimes \mathcal{A}$ is the DTMC:

$$\mathcal{D} \otimes \mathcal{A} \;=\; (S \times Q, \mathbf{P}', \iota'_{\text{init}}, \{\, accept \,\}, L')$$

Product Markov chain

Product Markov chain

Let $\mathcal{D} = (S, \mathbf{P}, \iota_{\text{init}}, AP, L)$ be a DTMC and $\mathcal{A} = (Q, 2^{AP}, \delta, q_0, F)$ be a DFA. The *product* $\mathcal{D} \otimes \mathcal{A}$ is the DTMC:

$$\mathcal{D} \otimes \mathcal{A} \; = \; (S \times Q, \mathbf{P}', \iota'_{\text{init}}, \{ \, accept \, \}, L')$$

where $L'(\langle s, q \rangle) = \{ \, accept \, \}$ if $q \in F$ and $L'(\langle s, q \rangle) = \varnothing$ otherwise,

Product Markov chain

Product Markov chain

Let $\mathcal{D} = (S, \mathbf{P}, \iota_{\text{init}}, AP, L)$ be a DTMC and $\mathcal{A} = (Q, 2^{AP}, \delta, q_0, F)$ be a DFA. The *product* $\mathcal{D} \otimes \mathcal{A}$ is the DTMC:

$$\mathcal{D} \otimes \mathcal{A} \;=\; (S \times Q, \mathbf{P}', \iota'_{\text{init}}, \{\, accept \,\}, L')$$

where $L'(\langle s, q \rangle) = \{\, accept \,\}$ if $q \in F$ and $L'(\langle s, q \rangle) = \varnothing$ otherwise, and

$$\iota'_{\text{init}}(\langle s, q \rangle) \;=\; \begin{cases} \iota_{\text{init}}(s) & \text{if } q = \delta(q_0, L(s)) \\ 0 & \text{otherwise.} \end{cases}$$

Product Markov chain

Product Markov chain

Let $\mathcal{D} = (S, \mathbf{P}, \iota_{\mathrm{init}}, AP, L)$ be a DTMC and $\mathcal{A} = (Q, 2^{AP}, \delta, q_0, F)$ be a DFA. The *product* $\mathcal{D} \otimes \mathcal{A}$ is the DTMC:

$$\mathcal{D} \otimes \mathcal{A} \;=\; (S \times Q, \mathbf{P}', \iota'_{\mathrm{init}}, \{\, accept \,\}, L')$$

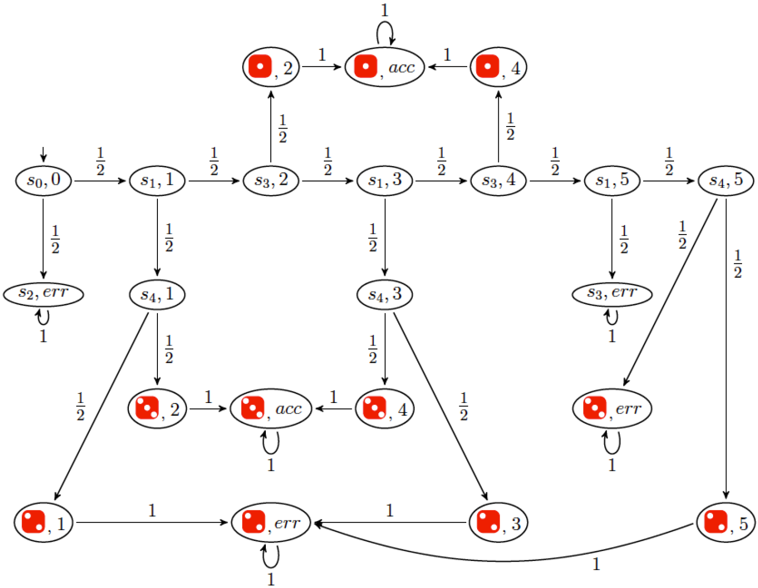
where $L'(\langle s, q \rangle) = \{\, accept \,\}$ if $q \in F$ and $L'(\langle s, q \rangle) = \varnothing$ otherwise, and

$$\iota'_{\mathrm{init}}(\langle s, q \rangle) \;=\; \begin{cases} \iota_{\mathrm{init}}(s) & \text{if } q = \delta(q_0, L(s)) \\ 0 & \text{otherwise.} \end{cases}$$

The transition probabilities in $\mathcal{D} \otimes \mathcal{A}$ are given by:

$$\mathbf{P}'(\langle s, q \rangle, \langle s', q' \rangle) \;=\; \begin{cases} \mathbf{P}(s, s') & \text{if } q' = \delta(q, L(s')) \\ 0 & \text{otherwise.} \end{cases}$$

Example product: Knuth-Yao's die



Product Markov chain

Product Markov chain

Some observations

- For each path $\pi = s_0\, s_1\, s_2 \ldots$ in DTMC $\mathcal{D}$ there exists a unique run $q_0\, q_1\, q_2 \ldots$ in DFA $\mathcal{A}$ for $trace(\pi) = L(s_0)\, L(s_1)\, L(s_2) \ldots$ and $\pi^+ = \langle s_0, q_1 \rangle \langle s_1, q_2 \rangle \langle s_2, q_3 \rangle \ldots$ is a path in $\mathcal{D} \otimes \mathcal{A}$.

Product Markov chain

Some observations

- For each path $\pi = s_0\, s_1\, s_2 \ldots$ in DTMC $\mathcal{D}$ there exists a unique run $q_0\, q_1\, q_2 \ldots$ in DFA $\mathcal{A}$ for $trace(\pi) = L(s_0)\, L(s_1)\, L(s_2) \ldots$ and $\pi^+ = \langle s_0, q_1 \rangle \langle s_1, q_2 \rangle \langle s_2, q_3 \rangle \ldots$ is a path in $\mathcal{D} \otimes \mathcal{A}$.
- The DFA $\mathcal{A}$ does not affect the probabilities, i.e., for each measurable set Π of paths in $\mathcal{D}$ and state s:

$$Pr_s^{\mathcal{D}}(\Pi) \;=\; Pr^{\mathcal{D}\otimes\mathcal{A}}_{\langle s, \delta(q_0, L(s)) \rangle} \underbrace{\{ \pi^+ \mid \pi \in \Pi \}}_{\Pi^+}$$

Product Markov chain

Some observations

- For each path $\pi = s_0\, s_1\, s_2 \ldots$ in DTMC $\mathcal{D}$ there exists a unique run $q_0\, q_1\, q_2 \ldots$ in DFA $\mathcal{A}$ for $\mathit{trace}(\pi) = L(s_0)\, L(s_1)\, L(s_2) \ldots$ and $\pi^+ = \langle s_0, q_1 \rangle \langle s_1, q_2 \rangle \langle s_2, q_3 \rangle \ldots$ is a path in $\mathcal{D} \otimes \mathcal{A}$.
- The DFA $\mathcal{A}$ does not affect the probabilities, i.e., for each measurable set Π of paths in $\mathcal{D}$ and state s:

$$Pr_s^{\mathcal{D}}(\Pi) = Pr^{\mathcal{D} \otimes \mathcal{A}}_{\langle s, \delta(q_0, L(s)) \rangle} \underbrace{\{ \pi^+ \mid \pi \in \Pi \}}_{\Pi^+}$$

- For $\Pi = \{ \pi \in \mathit{Paths}^{\mathcal{D}}(s) \mid \mathit{pref}(\mathit{trace}(\pi)) \cap \mathcal{L}(\mathcal{A}) \neq \varnothing \}$, the set Π^+ is given by:

$$\Pi^+ = \{ \pi^+ \in \mathit{Paths}^{\mathcal{D} \otimes \mathcal{A}}(\langle s, \delta(q_0, L(s)) \rangle) \mid \pi^+ \models \Diamond \mathit{accept} \}.$$

Quantitative analysis of regular safety properties

Quantitative analysis of regular safety properties

Theorem for analysing regular safety properties

Quantitative analysis of regular safety properties

Theorem for analysing regular safety properties

Let P_{safe} be a regular safety property, $\mathcal{A}$ a DFA for the set of bad prefixes of P_{safe}, $\mathcal{D}$ a DTMC, and s a state in $\mathcal{D}$. Then:

$$Pr^{\mathcal{D}}(s \models P_{safe}) \quad = \quad Pr^{\mathcal{D} \otimes \mathcal{A}}(\langle s, q_s \rangle \not\models \Diamond accept)$$

Quantitative analysis of regular safety properties

Theorem for analysing regular safety properties

Let P_{safe} be a regular safety property, $\mathcal{A}$ a DFA for the set of bad prefixes of P_{safe}, $\mathcal{D}$ a DTMC, and s a state in $\mathcal{D}$. Then:

$$\begin{aligned} Pr^{\mathcal{D}}(s \models P_{safe}) &= Pr^{\mathcal{D} \otimes \mathcal{A}}(\langle s, q_s \rangle \not\models \Diamond accept) \\ &= 1 - Pr^{\mathcal{D} \otimes \mathcal{A}}(\langle s, q_s \rangle \models \Diamond accept) \end{aligned}$$

where $q_s = \delta(q_0, L(s))$.

Quantitative analysis of regular safety properties

Theorem for analysing regular safety properties

Let P_{safe} be a regular safety property, $\mathcal{A}$ a DFA for the set of bad prefixes of P_{safe}, $\mathcal{D}$ a DTMC, and s a state in $\mathcal{D}$. Then:

$$\begin{aligned} Pr^{\mathcal{D}}(s \models P_{safe}) &= Pr^{\mathcal{D}\otimes\mathcal{A}}(\langle s, q_s\rangle \not\models \Diamond accept) \\ &= 1 - Pr^{\mathcal{D}\otimes\mathcal{A}}(\langle s, q_s\rangle \models \Diamond accept) \end{aligned}$$

where $q_s = \delta(q_0, L(s))$.

Remarks

Quantitative analysis of regular safety properties

Theorem for analysing regular safety properties

Let P_{safe} be a regular safety property, $\mathcal{A}$ a DFA for the set of bad prefixes of P_{safe}, $\mathcal{D}$ a DTMC, and s a state in $\mathcal{D}$. Then:

$$\begin{aligned} Pr^{\mathcal{D}}(s \models P_{safe}) &= Pr^{\mathcal{D} \otimes \mathcal{A}}(\langle s, q_s \rangle \not\models \Diamond accept) \\ &= 1 - Pr^{\mathcal{D} \otimes \mathcal{A}}(\langle s, q_s \rangle \models \Diamond accept) \end{aligned}$$

where $q_s = \delta(q_0, L(s))$.

Remarks

1. For finite DTMCs, $Pr^{\mathcal{D}}(s \models P_{safe})$ can thus be computed by determining reachability probabilities of *accept* states in $\mathcal{D} \otimes \mathcal{A}$.

Quantitative analysis of regular safety properties

Theorem for analysing regular safety properties

Let P_{safe} be a regular safety property, $\mathcal{A}$ a DFA for the set of bad prefixes of P_{safe}, $\mathcal{D}$ a DTMC, and s a state in $\mathcal{D}$. Then:

$$\begin{aligned} Pr^{\mathcal{D}}(s \models P_{safe}) &= Pr^{\mathcal{D}\otimes\mathcal{A}}(\langle s, q_s \rangle \not\models \Diamond accept) \\ &= 1 - Pr^{\mathcal{D}\otimes\mathcal{A}}(\langle s, q_s \rangle \models \Diamond accept) \end{aligned}$$

where $q_s = \delta(q_0, L(s))$.

Remarks

1. For finite DTMCs, $Pr^{\mathcal{D}}(s \models P_{safe})$ can thus be computed by determining reachability probabilities of *accept* states in $\mathcal{D} \otimes \mathcal{A}$. This amounts to solving a linear equation system.

Quantitative analysis of regular safety properties

Theorem for analysing regular safety properties

Let P_{safe} be a regular safety property, $\mathcal{A}$ a DFA for the set of bad prefixes of P_{safe}, $\mathcal{D}$ a DTMC, and s a state in $\mathcal{D}$. Then:

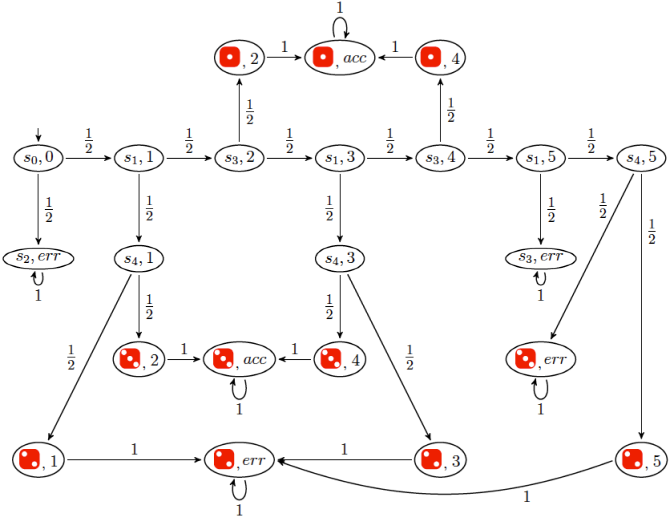
$$\begin{aligned} Pr^{\mathcal{D}}(s \models P_{safe}) &= Pr^{\mathcal{D}\otimes\mathcal{A}}(\langle s, q_s \rangle \not\models \Diamond accept) \\ &= 1 - Pr^{\mathcal{D}\otimes\mathcal{A}}(\langle s, q_s \rangle \models \Diamond accept) \end{aligned}$$

where $q_s = \delta(q_0, L(s))$.

Remarks

1. For finite DTMCs, $Pr^{\mathcal{D}}(s \models P_{safe})$ can thus be computed by determining reachability probabilities of *accept* states in $\mathcal{D} \otimes \mathcal{A}$. This amounts to solving a linear equation system.
2. For qualitative regular safety properties, i.e., $Pr^{\mathcal{D}}(s \models P_{safe}) > 0$ and $Pr^{\mathcal{D}}(s \models P_{safe}) = 1$, a graph analysis of $\mathcal{D} \otimes \mathcal{A}$ suffices.

Determining the property's probability



$Pr^{\mathcal{D} \otimes \mathcal{A}}(\langle s, q_s \rangle \models \Diamond accept)$ equals $\frac{1}{8} + \frac{1}{8} + \frac{1}{32} + \frac{1}{32} = \frac{5}{16}$.

Overview

ω-regular languages

ω-regular languages

Infinite repetition of languages

ω-regular languages

Infinite repetition of languages

Let Σ be a finite alphabet.

ω-regular languages

Infinite repetition of languages

Let Σ be a finite alphabet. For language $\mathcal{L} \subseteq \Sigma^*$, let $\mathcal{L}^\omega$ be the set of words in $\Sigma^* \cup \Sigma^\omega$ that arise from the infinite concatenation of (arbitrary) words in Σ,

ω-regular languages

Infinite repetition of languages

Let Σ be a finite alphabet. For language $\mathcal{L} \subseteq \Sigma^*$, let $\mathcal{L}^\omega$ be the set of words in $\Sigma^* \cup \Sigma^\omega$ that arise from the infinite concatenation of (arbitrary) words in Σ, i.e.,

$$\mathcal{L}^\omega \;=\; \left\{ w_1 w_2 w_3 \ldots \mid w_i \in \mathcal{L}, i \geqslant 1 \right\}.$$

ω-regular languages

Infinite repetition of languages

Let Σ be a finite alphabet. For language $\mathcal{L} \subseteq \Sigma^*$, let $\mathcal{L}^\omega$ be the set of words in $\Sigma^* \cup \Sigma^\omega$ that arise from the infinite concatenation of (arbitrary) words in Σ, i.e.,

$$\mathcal{L}^\omega = \left\{ w_1 w_2 w_3 \ldots \mid w_i \in \mathcal{L}, i \geqslant 1 \right\}.$$

The result is an *ω-language*, i.e., $\mathcal{L} \subseteq \Sigma^*$, provided that $\mathcal{L} \subseteq \Sigma^+$, i.e., $\varepsilon \notin \mathcal{L}$.

ω-regular languages

Infinite repetition of languages

Let Σ be a finite alphabet. For language $\mathcal{L} \subseteq \Sigma^*$, let $\mathcal{L}^\omega$ be the set of words in $\Sigma^* \cup \Sigma^\omega$ that arise from the infinite concatenation of (arbitrary) words in Σ, i.e.,

$$\mathcal{L}^\omega \; = \; \left\{ w_1 w_2 w_3 \ldots \mid w_i \in \mathcal{L}, i \geqslant 1 \right\}.$$

The result is an *ω-language*, i.e., $\mathcal{L} \subseteq \Sigma^*$, provided that $\mathcal{L} \subseteq \Sigma^+$, i.e., $\varepsilon \notin \mathcal{L}$.

ω-regular expression

An *ω-regular expression* G over the Σ has the form:

ω-regular languages

Infinite repetition of languages

Let Σ be a finite alphabet. For language $\mathcal{L} \subseteq \Sigma^*$, let $\mathcal{L}^\omega$ be the set of words in $\Sigma^* \cup \Sigma^\omega$ that arise from the infinite concatenation of (arbitrary) words in Σ, i.e.,

$$\mathcal{L}^\omega \ = \ \left\{ w_1 w_2 w_3 \ldots \mid w_i \in \mathcal{L}, i \geqslant 1 \right\}.$$

The result is an *ω-language*, i.e., $\mathcal{L} \subseteq \Sigma^*$, provided that $\mathcal{L} \subseteq \Sigma^+$, i.e., $\varepsilon \notin \mathcal{L}$.

ω-regular expression

An *ω-regular expression* G over the Σ has the form: $\mathsf{G} = \mathsf{E}_1.\mathsf{F}_1^\omega + \ldots + \mathsf{E}_n.\mathsf{F}_n^\omega$ where $n \geqslant 1$ and $\mathsf{E}_1, \ldots, \mathsf{E}_n, \mathsf{F}_1, \ldots, \mathsf{F}_n$ are regular expressions over Σ such that $\varepsilon \notin \mathcal{L}(\mathsf{F}_i)$, for all $1 \leqslant i \leqslant n$.

ω-regular languages

Infinite repetition of languages

Let Σ be a finite alphabet. For language $\mathcal{L} \subseteq \Sigma^*$, let $\mathcal{L}^\omega$ be the set of words in $\Sigma^* \cup \Sigma^\omega$ that arise from the infinite concatenation of (arbitrary) words in Σ, i.e.,

$$\mathcal{L}^\omega \;=\; \left\{ w_1 w_2 w_3 \ldots \mid w_i \in \mathcal{L}, i \geqslant 1 \right\}.$$

The result is an *ω-language*, i.e., $\mathcal{L} \subseteq \Sigma^*$, provided that $\mathcal{L} \subseteq \Sigma^+$, i.e., $\varepsilon \notin \mathcal{L}$.

ω-regular expression

An *ω-regular expression* G over the Σ has the form: $\mathsf{G} = \mathsf{E}_1.\mathsf{F}_1^\omega + \ldots + \mathsf{E}_n.\mathsf{F}_n^\omega$ where $n \geqslant 1$ and $\mathsf{E}_1, \ldots, \mathsf{E}_n, \mathsf{F}_1, \ldots, \mathsf{F}_n$ are regular expressions over Σ such that $\varepsilon \notin \mathcal{L}(\mathsf{F}_i)$, for all $1 \leqslant i \leqslant n$.

The *semantics* of G is defined by $\mathcal{L}_\omega(\mathsf{G}) \;=\; \mathcal{L}(\mathsf{E}_1).\mathcal{L}(\mathsf{F}_1)^\omega \cup \ldots \cup \mathcal{L}(\mathsf{E}_n).\mathcal{L}(\mathsf{F}_n)^\omega$ where $\mathcal{L}(\mathsf{E}) \subseteq \Sigma^*$ denotes the language (of finite words) induced by the regular expression E.

ω-regular expressions

ω-regular expression

An *ω-regular expression* G over the Σ has the form: $G = E_1.F_1^\omega + \ldots + E_n.F_n^\omega$ where $n \geqslant 1$ and $E_1, \ldots, E_n, F_1, \ldots, F_n$ are regular expressions over Σ such that $\varepsilon \notin \mathcal{L}(F_i)$, for all $1 \leqslant i \leqslant n$.

The semantics of G is defined by $\mathcal{L}_\omega(G) \; = \; \mathcal{L}(E_1).\mathcal{L}(F_1)^\omega \cup \ldots \cup \mathcal{L}(E_n).\mathcal{L}(F_n)^\omega$ where $\mathcal{L}(E) \subseteq \Sigma^*$ denotes the language (of finite words) induced by the regular expression E.

ω-regular expressions

ω-regular expression

An *ω-regular expression* G over the Σ has the form: $\mathsf{G} = \mathsf{E}_1.\mathsf{F}_1^\omega + \ldots + \mathsf{E}_n.\mathsf{F}_n^\omega$ where $n \geqslant 1$ and $\mathsf{E}_1, \ldots, \mathsf{E}_n, \mathsf{F}_1, \ldots, \mathsf{F}_n$ are regular expressions over Σ such that $\varepsilon \notin \mathcal{L}(\mathsf{F}_i)$, for all $1 \leqslant i \leqslant n$.

The semantics of G is defined by $\mathcal{L}_\omega(\mathsf{G}) \ = \ \mathcal{L}(\mathsf{E}_1).\mathcal{L}(\mathsf{F}_1)^\omega \cup \ldots \cup \mathcal{L}(\mathsf{E}_n).\mathcal{L}(\mathsf{F}_n)^\omega$ where $\mathcal{L}(\mathsf{E}) \subseteq \Sigma^*$ denotes the language (of finite words) induced by the regular expression E.

Example

Examples for ω-regular expressions over the alphabet $\Sigma = \{ A, B, C \}$ are

$$(A + B)^* A(AAB + C)^\omega \quad \text{or} \quad A(B + C)^* A^\omega + B(A + C)^\omega.$$

ω-regular properties

ω-regular properties

ω-regular property

LT property P over AP is called *ω-regular* if $P = \mathcal{L}_\omega(\mathrm{G})$ for some ω-regular expression G over the alphabet 2^{AP}.

ω-regular properties

ω-regular property

LT property P over AP is called *ω-regular* if $P = \mathcal{L}_\omega(\mathrm{G})$ for some ω-regular expression G over the alphabet 2^{AP}.

Example

Let $AP = \{ a, b \}$. Then some ω-regular properties over AP are:

ω-regular properties

ω-regular property

LT property P over AP is called *ω-regular* if $P = \mathcal{L}_\omega(\mathrm{G})$ for some ω-regular expression G over the alphabet 2^{AP}.

Example

Let $AP = \{ a, b \}$. Then some ω-regular properties over AP are:

- always a, i.e., $(\{ a \} + \{ a, b \})^\omega$.

ω-regular properties

ω-regular property

LT property P over AP is called *ω-regular* if $P = \mathcal{L}_\omega(\mathrm{G})$ for some ω-regular expression G over the alphabet 2^{AP}.

Example

Let $AP = \{ a, b \}$. Then some ω-regular properties over AP are:

- always a, i.e., $(\{ a \} + \{ a, b \})^\omega$.
- eventually a, i.e., $(\varnothing + \{ b \})^*.(\{ a \} + \{ a, b \}).(2^{AP})^\omega$.

ω-regular properties

ω-regular property

LT property P over AP is called *ω-regular* if $P = \mathcal{L}_\omega(\mathrm{G})$ for some ω-regular expression G over the alphabet 2^{AP}.

Example

Let $AP = \{ a, b \}$. Then some ω-regular properties over AP are:

- always a, i.e., $(\{ a \} + \{ a, b \})^\omega$.
- eventually a, i.e., $(\varnothing + \{ b \})^*.(\{ a \} + \{ a, b \}).(2^{AP})^\omega$.
- infinitely often a, i.e., $((\varnothing + \{ b \})^*.(\{ a \} + \{ a, b \}))^\omega$.

ω-regular properties

ω-regular property

LT property P over AP is called *ω-regular* if $P = \mathcal{L}_\omega(\mathrm{G})$ for some ω-regular expression G over the alphabet 2^{AP}.

Example

Let $AP = \{ a, b \}$. Then some ω-regular properties over AP are:

- always a, i.e., $(\{ a \} + \{ a, b \})^\omega$.
- eventually a, i.e., $(\varnothing + \{ b \})^*.(\{ a \} + \{ a, b \}).(2^{AP})^\omega$.
- infinitely often a, i.e., $((\varnothing + \{ b \})^*.(\{ a \} + \{ a, b \}))^\omega$.
- from some moment on, always a, i.e., $(2^{AP})^*.(\{ a \} + \{ a, b \})^\omega$.

ω-regular properties

ω-regular property

LT property P over AP is called *ω-regular* if $P = \mathcal{L}_\omega(\mathrm{G})$ for some ω-regular expression G over the alphabet 2^{AP}.

Example

Any regular safety property P_{safe} is an ω-regular property.

ω-regular properties

ω-regular property

LT property P over AP is called *ω-regular* if $P = \mathcal{L}_\omega(\mathrm{G})$ for some ω-regular expression G over the alphabet 2^{AP}.

Example

Any regular safety property P_{safe} is an ω-regular property. This follows from the fact that the complement language

$$\left(2^{AP}\right)^\omega \setminus P_{safe} \;=\; \underbrace{BadPref(P_{safe})}_{\text{regular}} . \left(2^{AP}\right)^\omega$$

is an ω-regular language, and ω-regular languages are closed under complement.

ω-regular properties

ω-regular property

LT property P over AP is called *ω-regular* if $P = \mathcal{L}_\omega(\mathrm{G})$ for some ω-regular expression G over the alphabet 2^{AP}.

ω-regular properties

ω-regular property

LT property P over AP is called *ω-regular* if $P = \mathcal{L}_\omega(\mathrm{G})$ for some ω-regular expression G over the alphabet 2^{AP}.

Example

Starvation freedom in the sense of "whenever process $\mathcal{P}$ is waiting then it will enter its critical section eventually" is an ω-regular property

ω-regular properties

ω-regular property

LT property P over AP is called *ω-regular* if $P = \mathcal{L}_\omega(\mathrm{G})$ for some ω-regular expression G over the alphabet 2^{AP}.

Example

Starvation freedom in the sense of "whenever process $\mathcal{P}$ is waiting then it will enter its critical section eventually" is an ω-regular property as it can be described by

$$\big((\neg \mathit{wait})^*.\mathit{wait}.\mathrm{true}^*.\mathit{crit}\big)^\omega \; + \; \big((\neg \mathit{wait})^*.\mathit{wait}.\mathrm{true}^*.\mathit{crit}\big)^*.(\neg \mathit{wait})^\omega$$

ω-regular properties

ω-regular property

LT property P over AP is called *ω-regular* if $P = \mathcal{L}_\omega(\mathrm{G})$ for some ω-regular expression G over the alphabet 2^{AP}.

Example

Starvation freedom in the sense of "whenever process $\mathcal{P}$ is waiting then it will enter its critical section eventually" is an ω-regular property as it can be described by

$$\left((\neg wait)^*.wait.\mathrm{true}^*.crit\right)^\omega \;+\; \left((\neg wait)^*.wait.\mathrm{true}^*.crit\right)^*.(\neg wait)^\omega$$

Intuitively, the first summand stands for the case where $\mathcal{P}$ requests and enters its critical section infinitely often,

ω-regular properties

ω-regular property

LT property P over AP is called *ω-regular* if $P = \mathcal{L}_\omega(\mathrm{G})$ for some ω-regular expression G over the alphabet 2^{AP}.

Example

Starvation freedom in the sense of "whenever process $\mathcal{P}$ is waiting then it will enter its critical section eventually" is an ω-regular property as it can be described by

$$\big((\neg wait)^*.wait.\mathrm{true}^*.crit\big)^\omega \; + \; \big((\neg wait)^*.wait.\mathrm{true}^*.crit\big)^*.(\neg wait)^\omega$$

Intuitively, the first summand stands for the case where $\mathcal{P}$ requests and enters its critical section infinitely often, while the second summand stands for the case where $\mathcal{P}$ is in its waiting phase only finitely many times.

Overview

Deterministic Büchi automata

Deterministic Büchi automata

Deterministic Büchi Automaton (DBA)

A *deterministic Büchi automaton* (DBA) $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ with

Deterministic Büchi automata

Deterministic Büchi Automaton (DBA)

A *deterministic Büchi automaton* (DBA) $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ with

- Q is a finite set of states with initial state $q_0 \in Q_0$,

Deterministic Büchi automata

Deterministic Büchi Automaton (DBA)

A *deterministic Büchi automaton* (DBA) $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ with

- Q is a finite set of states with initial state $q_0 \in Q_0$,
- Σ is an alphabet,

Deterministic Büchi automata

Deterministic Büchi Automaton (DBA)

A *deterministic Büchi automaton* (DBA) $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ with

- Q is a finite set of states with initial state $q_0 \in Q_0$,
- Σ is an alphabet,
- $\delta : Q \times \Sigma \rightarrow Q$ is a transition function,

Deterministic Büchi automata

Deterministic Büchi Automaton (DBA)

A *deterministic Büchi automaton* (DBA) $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ with

- Q is a finite set of states with initial state $q_0 \in Q_0$,
- Σ is an alphabet,
- $\delta : Q \times \Sigma \rightarrow Q$ is a transition function,
- $F \subseteq Q$ is a set of *accept* (or: final) states.

Deterministic Büchi automata

Deterministic Büchi Automaton (DBA)

A *deterministic Büchi automaton* (DBA) $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ with

- Q is a finite set of states with initial state $q_0 \in Q_0$,
- Σ is an alphabet,
- $\delta : Q \times \Sigma \rightarrow Q$ is a transition function,
- $F \subseteq Q$ is a set of *accept* (or: final) states.

A *run* for $\sigma = A_0 A_1 A_2 \ldots \in \Sigma^\omega$ denotes an infinite sequence $q_0\, q_1\, q_2 \ldots$ of states in $\mathcal{A}$ such that $q_0 \in Q_0$ and $q_i \xrightarrow{A_i} q_{i+1}$ for $i \geqslant 0$.

Deterministic Büchi automata

Deterministic Büchi Automaton (DBA)

A *deterministic Büchi automaton* (DBA) $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ with

- Q is a finite set of states with initial state $q_0 \in Q_0$,
- Σ is an alphabet,
- $\delta : Q \times \Sigma \rightarrow Q$ is a transition function,
- $F \subseteq Q$ is a set of *accept* (or: final) states.

A *run* for $\sigma = A_0 A_1 A_2 \ldots \in \Sigma^\omega$ denotes an infinite sequence $q_0\, q_1\, q_2 \ldots$ of states in $\mathcal{A}$ such that $q_0 \in Q_0$ and $q_i \xrightarrow{A_i} q_{i+1}$ for $i \geqslant 0$.

Run $q_0\, q_1\, q_2 \ldots$ is *accepting* if $q_i \in F$ for infinitely many indices $i \in \mathbb{N}$.

Deterministic Büchi automata

Deterministic Büchi Automaton (DBA)

A *deterministic Büchi automaton* (DBA) $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ with

- Q is a finite set of states with initial state $q_0 \in Q_0$,
- Σ is an alphabet,
- $\delta : Q \times \Sigma \rightarrow Q$ is a transition function,
- $F \subseteq Q$ is a set of *accept* (or: final) states.

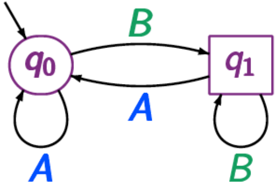
A *run* for $\sigma = A_0 A_1 A_2 \ldots \in \Sigma^\omega$ denotes an infinite sequence $q_0\, q_1\, q_2 \ldots$ of states in $\mathcal{A}$ such that $q_0 \in Q_0$ and $q_i \xrightarrow{A_i} q_{i+1}$ for $i \geqslant 0$.

Run $q_0\, q_1\, q_2 \ldots$ is *accepting* if $q_i \in F$ for infinitely many indices $i \in \mathbb{N}$.

The infinite *language* of $\mathcal{A}$ is

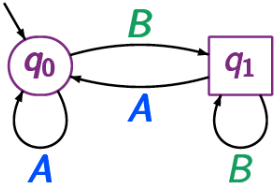
$$\mathcal{L}_\omega(\mathcal{A}) \ = \ \{\, \sigma \in \Sigma^\omega \mid \text{there exists an accepting run for } \sigma \text{ in } \mathcal{A} \,\}.$$

Deterministic Büchi automata for LT properties



DBA over $\{ A, B \}$ with $F = \{ q_1 \}$ and initial state q_0

Deterministic Büchi automata for LT properties



DBA over $\{ A, B \}$ with $F = \{ q_1 \}$ and initial state q_0 accepting the LT property "infinitely often B".

Some facts about DBA

Some facts about DBA

Expressiveness of DBA

For any DBA $\mathcal{A}$, the language $\mathcal{L}_\omega(\mathcal{A})$ is ω-regular.

Some facts about DBA

Expressiveness of DBA

For any DBA $\mathcal{A}$, the language $\mathcal{L}_\omega(\mathcal{A})$ is ω-regular.

There does not exist a DBA over the alphabet $\Sigma = \{ a, b \}$ for the ω-regular expression $(a+b)^*.a^\omega$.

Some facts about DBA

Expressiveness of DBA

For any DBA $\mathcal{A}$, the language $\mathcal{L}_\omega(\mathcal{A})$ is ω-regular.

There does not exist a DBA over the alphabet $\Sigma = \{ a, b \}$ for the ω-regular expression $(a + b)^*.a^\omega$.

The class of DBA-recognizable languages is a proper subclass of the class of ω-regular languages

Some facts about DBA

Expressiveness of DBA

For any DBA $\mathcal{A}$, the language $\mathcal{L}_\omega(\mathcal{A})$ is ω-regular.

There does not exist a DBA over the alphabet $\Sigma = \{ a, b \}$ for the ω-regular expression $(a + b)^*.a^\omega$.

The class of DBA-recognizable languages is a proper subclass of the class of ω-regular languages and is not closed under complementation.

Some facts about DBA

Expressiveness of DBA

For any DBA $\mathcal{A}$, the language $\mathcal{L}_\omega(\mathcal{A})$ is ω-regular.

There does not exist a DBA over the alphabet $\Sigma = \{ a, b \}$ for the ω-regular expression $(a + b)^*.a^\omega$.

The class of DBA-recognizable languages is a proper subclass of the class of ω-regular languages and is not closed under complementation.

An ω-language is recognizable by a DBA iff it is the limit language of a regular language. (Details: see lecture Applications of Automata Theory.)

Quantitative analysis of DBA properties

Quantitative analysis of DBA properties

Quantitative Analysis for DBA-Definable Properties

Let $\mathcal{A}$ be a DBA and $\mathcal{D}$ a DTMC. Then, for all states s in $\mathcal{D}$:

$$Pr^{\mathcal{D}}(s \models \mathcal{A}) \;=\; Pr^{\mathcal{D} \otimes \mathcal{A}}(\langle s, q_s \rangle \models \Box\Diamond accept)$$

where $q_s = \delta(q_0, L(s))$.

Quantitative analysis of DBA properties

Quantitative Analysis for DBA-Definable Properties

Let $\mathcal{A}$ be a DBA and $\mathcal{D}$ a DTMC. Then, for all states s in $\mathcal{D}$:

$$Pr^{\mathcal{D}}(s \models \mathcal{A}) \;=\; Pr^{\mathcal{D}\otimes\mathcal{A}}(\langle s, q_s \rangle \models \Box\Diamond accept)$$

where $q_s = \delta(q_0, L(s))$.

Algorithm

Recall that for finite DTMCs, the probability of $\Box\Diamond accept$ can be obtained in polynomial time by first determining the BSCCs of $\mathcal{D} \otimes \mathcal{A}$.

Quantitative analysis of DBA properties

Quantitative Analysis for DBA-Definable Properties

Let $\mathcal{A}$ be a DBA and $\mathcal{D}$ a DTMC. Then, for all states s in $\mathcal{D}$:

$$Pr^{\mathcal{D}}(s \models \mathcal{A}) \;=\; Pr^{\mathcal{D} \otimes \mathcal{A}}(\langle s, q_s \rangle \models \Box\Diamond accept)$$

where $q_s = \delta(q_0, L(s))$.

Algorithm

Recall that for finite DTMCs, the probability of $\Box\Diamond accept$ can be obtained in polynomial time by first determining the BSCCs of $\mathcal{D} \otimes \mathcal{A}$. For each BSCC B that contains a state $\langle s, q \rangle$ with $q \in F$, determine the probability of eventually reaching B.

Quantitative analysis of DBA properties

Quantitative Analysis for DBA-Definable Properties

Let $\mathcal{A}$ be a DBA and $\mathcal{D}$ a DTMC. Then, for all states s in $\mathcal{D}$:

$$Pr^{\mathcal{D}}(s \models \mathcal{A}) \;=\; Pr^{\mathcal{D} \otimes \mathcal{A}}(\langle s, q_s \rangle \models \Box\Diamond accept)$$

where $q_s = \delta(q_0, L(s))$.

Algorithm

Recall that for finite DTMCs, the probability of $\Box\Diamond accept$ can be obtained in polynomial time by first determining the BSCCs of $\mathcal{D} \otimes \mathcal{A}$. For each BSCC B that contains a state $\langle s, q \rangle$ with $q \in F$, determine the probability of eventually reaching B. Its sum is the required probability.

Quantitative analysis of DBA properties

Quantitative Analysis for DBA-Definable Properties

Let $\mathcal{A}$ be a DBA and $\mathcal{D}$ a DTMC. Then, for all states s in $\mathcal{D}$:

$$Pr^{\mathcal{D}}(s \models \mathcal{A}) \;=\; Pr^{\mathcal{D}\otimes\mathcal{A}}(\langle s, q_s \rangle \models \Box\Diamond accept)$$

where $q_s = \delta(q_0, L(s))$.

Algorithm

Recall that for finite DTMCs, the probability of $\Box\Diamond accept$ can be obtained in polynomial time by first determining the BSCCs of $\mathcal{D} \otimes \mathcal{A}$. For each BSCC B that contains a state $\langle s, q \rangle$ with $q \in F$, determine the probability of eventually reaching B. Its sum is the required probability. Thus this amounts to solve a linear equation system for each accepting BSCC in $\mathcal{D}$.

Overview

Beyond DBA properties

Beyond DBA properties

Remarks

- Since DBAs do not have the full power of ω-regular languages, this approach is not capable of handling arbitrary ω-regular properties.

Beyond DBA properties

Remarks

- Since DBAs do not have the full power of ω-regular languages, this approach is not capable of handling arbitrary ω-regular properties.
- To overcome this deficiency, Büchi automata will be replaced by an alternative automaton model for which their deterministic counterparts are as expressive as ω-regular languages.

Beyond DBA properties

Remarks

- Since DBAs do not have the full power of ω-regular languages, this approach is not capable of handling arbitrary ω-regular properties.
- To overcome this deficiency, Büchi automata will be replaced by an alternative automaton model for which their deterministic counterparts are as expressive as ω-regular languages.
- Such automata have the same components as DBA (finite set of states, and so on) except for the acceptance sets. We consider *deterministic Rabin automata*.

Beyond DBA properties

Remarks

- Since DBAs do not have the full power of ω-regular languages, this approach is not capable of handling arbitrary ω-regular properties.
- To overcome this deficiency, Büchi automata will be replaced by an alternative automaton model for which their deterministic counterparts are as expressive as ω-regular languages.
- Such automata have the same components as DBA (finite set of states, and so on) except for the acceptance sets. We consider *deterministic Rabin automata*. There are alternatives, e.g., Muller automata.
- Determinism is important to stay within the realm of Markov chains; a product of an MC with a deterministic automaton yields a MC.

Deterministic Rabin automata

Deterministic Rabin automata

Deterministic Rabin automaton

A *deterministic Rabin automaton* (DRA) $\mathcal{A} = (Q, \Sigma, \delta, q_0, \mathcal{F})$ with

- Q, $q_0 \in Q_0$, Σ is an alphabet, and $\delta : Q \times \Sigma \rightarrow Q$ as before

Deterministic Rabin automata

Deterministic Rabin automaton

A *deterministic Rabin automaton* (DRA) $\mathcal{A} = (Q, \Sigma, \delta, q_0, \mathcal{F})$ with

- Q, $q_0 \in Q_0$, Σ is an alphabet, and $\delta : Q \times \Sigma \to Q$ as before
- $\mathcal{F} = \{ (L_i, K_i) \mid 0 < i \leqslant k \}$ with $L_i, K_i \subseteq Q$, is a set of *accept pairs*

Deterministic Rabin automata

Deterministic Rabin automaton

A *deterministic Rabin automaton* (DRA) $\mathcal{A} = (Q, \Sigma, \delta, q_0, \mathcal{F})$ with

- Q, $q_0 \in Q_0$, Σ is an alphabet, and $\delta : Q \times \Sigma \rightarrow Q$ as before
- $\mathcal{F} = \{ (L_i, K_i) \mid 0 < i \leqslant k \}$ with $L_i, K_i \subseteq Q$, is a set of *accept pairs*

A *run* for $\sigma = A_0 A_1 A_2 \ldots \in \Sigma^\omega$ denotes an infinite sequence $q_0\, q_1\, q_2 \ldots$ of states in $\mathcal{A}$ such that $q_0 \in Q_0$ and $q_i \xrightarrow{A_i} q_{i+1}$ for $i \geqslant 0$.

Deterministic Rabin automata

Deterministic Rabin automaton

A *deterministic Rabin automaton* (DRA) $\mathcal{A} = (Q, \Sigma, \delta, q_0, \mathcal{F})$ with

- Q, $q_0 \in Q_0$, Σ is an alphabet, and $\delta : Q \times \Sigma \to Q$ as before
- $\mathcal{F} = \{ (L_i, K_i) \mid 0 < i \leqslant k \}$ with $L_i, K_i \subseteq Q$, is a set of *accept pairs*

A *run* for $\sigma = A_0 A_1 A_2 \ldots \in \Sigma^\omega$ denotes an infinite sequence $q_0\, q_1\, q_2 \ldots$ of states in $\mathcal{A}$ such that $q_0 \in Q_0$ and $q_i \xrightarrow{A_i} q_{i+1}$ for $i \geqslant 0$.

Run $q_0\, q_1\, q_2 \ldots$ is *accepting* if for some pair (L_i, K_i), the states in L_i are visited finitely often and the states in K_i infinitely often.

Deterministic Rabin automata

Deterministic Rabin automaton

A *deterministic Rabin automaton* (DRA) $\mathcal{A} = (Q, \Sigma, \delta, q_0, \mathcal{F})$ with

- Q, $q_0 \in Q_0$, Σ is an alphabet, and $\delta : Q \times \Sigma \rightarrow Q$ as before
- $\mathcal{F} = \{ (L_i, K_i) \mid 0 < i \leqslant k \}$ with $L_i, K_i \subseteq Q$, is a set of *accept pairs*

A *run* for $\sigma = A_0 A_1 A_2 \ldots \in \Sigma^\omega$ denotes an infinite sequence $q_0\, q_1\, q_2 \ldots$ of states in $\mathcal{A}$ such that $q_0 \in Q_0$ and $q_i \xrightarrow{A_i} q_{i+1}$ for $i \geqslant 0$.

Run $q_0\, q_1\, q_2 \ldots$ is *accepting* if for some pair (L_i, K_i), the states in L_i are visited finitely often and the states in K_i infinitely often. That is, an accepting run should satisfy

$$\bigvee_{0<i\leqslant k} (\Diamond\Box\neg L_i \wedge \Box\Diamond K_i).$$

When does a DRA accept an infinite word?

Acceptance condition

A run of a word in Σ^ω on a DRA is accepting if and only if:

for some $(L_i, K_i) \in \mathcal{F}$, the states in L_i are visited finitely often
and (some of) the states in K_i are visited infinitely often

When does a DRA accept an infinite word?

Acceptance condition

A run of a word in Σ^ω on a DRA is accepting if and only if:

for some $(L_i, K_i) \in \mathcal{F}$, the states in L_i are visited finitely often

and (some of) the states in K_i are visited infinitely often

Stated in terms of an LTL formula:

$$\bigvee_{0 < i \leqslant k} (\Diamond \Box \neg L_i \wedge \Box \Diamond K_i)$$

When does a DRA accept an infinite word?

Acceptance condition

A run of a word in Σ^ω on a DRA is accepting if and only if:

for some $(L_i, K_i) \in \mathcal{F}$, the states in L_i are visited finitely often

and (some of) the states in K_i are visited infinitely often

Stated in terms of an LTL formula:

$$\bigvee_{0 < i \leqslant k} (\Diamond \Box \neg L_i \wedge \Box \Diamond K_i)$$

A deterministic Büchi automaton is a DRA with acceptance condition $\{ (\varnothing, F) \}$.

Deterministic Rabin automaton: Example

Deterministic Rabin automaton: Example

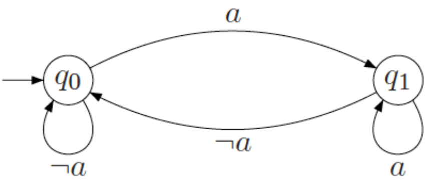
Acceptance condition

A run of a word in Σ^ω on a DRA is accepting iff $\bigvee_{0<i\leqslant k} (\Diamond \Box \neg L_i \wedge \Box\Diamond K_i)$.

Deterministic Rabin automaton: Example

Acceptance condition

A run of a word in Σ^ω on a DRA is accepting iff $\bigvee_{0<i\leqslant k} (\Diamond\Box\neg L_i \wedge \Box\Diamond K_i)$.

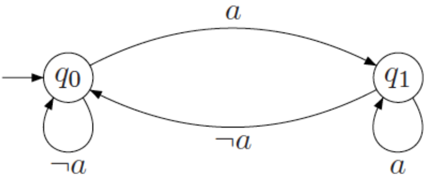


For $\mathcal{F} = \{ (L, K) \}$ with $L = \{ q_0 \}$ and $K = \{ q_1 \}$, this DRA accepts $\Diamond\Box a$

Deterministic Rabin automaton: Example

Acceptance condition

A run of a word in Σ^ω on a DRA is accepting iff $\bigvee_{0<i\leqslant k} (\Diamond \Box \neg L_i \wedge \Box \Diamond K_i)$.



For $\mathcal{F} = \{ (L, K) \}$ with $L = \{ q_0 \}$ and $K = \{ q_1 \}$, this DRA accepts $\Diamond \Box a$

Recall that there does not exist a deterministic Büchi automaton for $\Diamond \Box a$.

Deterministic Rabin automata

Deterministic Rabin automata

DRA are ω-regular

A language on infinite words is ω-regular iff there exists a DRA that generates it.

Deterministic Rabin automata

DRA are ω-regular

A language on infinite words is ω-regular iff there exists a DRA that generates it.

- DRA are thus equally expressive as nondeterministic Büchi automata.
- They are more expressive than deterministic Büchi automata.
- Any nondeterministic Büchi automata of n states can be converted to a DRA of size $2^{\mathcal{O}(n \cdot \log n)}$. (Details omitted.)

Verifying DRA properties

Verifying DRA properties

Product of a Markov chain and a DRA

The product of DTMC $\mathcal{D}$ and DRA $\mathcal{A}$ is defined as the product of a Markov chain and a DFA, except that the labeling is defined differently.

Verifying DRA properties

Product of a Markov chain and a DRA

The product of DTMC $\mathcal{D}$ and DRA $\mathcal{A}$ is defined as the product of a Markov chain and a DFA, except that the labeling is defined differently.

Let the acceptance condition of $\mathcal{A}$ is $\mathcal{F} = \{ (L_1, K_1), \ldots, (L_k, K_k) \}$.

Verifying DRA properties

Product of a Markov chain and a DRA

The product of DTMC $\mathcal{D}$ and DRA $\mathcal{A}$ is defined as the product of a Markov chain and a DFA, except that the labeling is defined differently.

Let the acceptance condition of $\mathcal{A}$ is $\mathcal{F} = \{ (L_1, K_1), \ldots, (L_k, K_k) \}$. Then the sets L_i, K_i serve as atomic propositions in $\mathcal{D} \otimes \mathcal{A}$.

Verifying DRA properties

Product of a Markov chain and a DRA

The product of DTMC $\mathcal{D}$ and DRA $\mathcal{A}$ is defined as the product of a Markov chain and a DFA, except that the labeling is defined differently.

Let the acceptance condition of $\mathcal{A}$ is $\mathcal{F} = \{ (L_1, K_1), \ldots, (L_k, K_k) \}$. Then the sets L_i, K_i serve as atomic propositions in $\mathcal{D} \otimes \mathcal{A}$. The labeling function L' in $\mathcal{D} \otimes \mathcal{A}$ is the obvious one: if $H \in \{ L_1, \ldots, L_k, K_1, \ldots, K_k \}$, then $H \in L'(\langle s, q \rangle)$ iff $q \in H$.

Verifying DRA properties

Product of a Markov chain and a DRA

The product of DTMC $\mathcal{D}$ and DRA $\mathcal{A}$ is defined as the product of a Markov chain and a DFA, except that the labeling is defined differently.

Let the acceptance condition of $\mathcal{A}$ is $\mathcal{F} = \{ (L_1, K_1), \ldots, (L_k, K_k) \}$. Then the sets L_i, K_i serve as atomic propositions in $\mathcal{D} \otimes \mathcal{A}$. The labeling function L' in $\mathcal{D} \otimes \mathcal{A}$ is the obvious one: if $H \in \{ L_1, \ldots, L_k, K_1, \ldots, K_k \}$, then $H \in L'(\langle s, q \rangle)$ iff $q \in H$.

Accepting BSCC

Verifying DRA properties

Product of a Markov chain and a DRA

The product of DTMC $\mathcal{D}$ and DRA $\mathcal{A}$ is defined as the product of a Markov chain and a DFA, except that the labeling is defined differently.

Let the acceptance condition of $\mathcal{A}$ is $\mathcal{F} = \{ (L_1, K_1), \ldots, (L_k, K_k) \}$. Then the sets L_i, K_i serve as atomic propositions in $\mathcal{D} \otimes \mathcal{A}$. The labeling function L' in $\mathcal{D} \otimes \mathcal{A}$ is the obvious one: if $H \in \{ L_1, \ldots, L_k, K_1, \ldots, K_k \}$, then $H \in L'(\langle s, q \rangle)$ iff $q \in H$.

Accepting BSCC

A BSCC T in $\mathcal{D} \otimes \mathcal{A}$ is *accepting* iff for some index $i \in \{ 1, \ldots, k \}$ we have:

Verifying DRA properties

Product of a Markov chain and a DRA

The product of DTMC $\mathcal{D}$ and DRA $\mathcal{A}$ is defined as the product of a Markov chain and a DFA, except that the labeling is defined differently.

Let the acceptance condition of $\mathcal{A}$ is $\mathcal{F} = \{ (L_1, K_1), \ldots, (L_k, K_k) \}$. Then the sets L_i, K_i serve as atomic propositions in $\mathcal{D} \otimes \mathcal{A}$. The labeling function L' in $\mathcal{D} \otimes \mathcal{A}$ is the obvious one: if $H \in \{ L_1, \ldots, L_k, K_1, \ldots, K_k \}$, then $H \in L'(\langle s, q \rangle)$ iff $q \in H$.

Accepting BSCC

A BSCC T in $\mathcal{D} \otimes \mathcal{A}$ is *accepting* iff for some index $i \in \{ 1, \ldots, k \}$ we have:

$$T \cap (S \times L_i) = \varnothing \quad \text{and} \quad T \cap (S \times K_i) \neq \varnothing.$$

Verifying DRA properties

Product of a Markov chain and a DRA

The product of DTMC $\mathcal{D}$ and DRA $\mathcal{A}$ is defined as the product of a Markov chain and a DFA, except that the labeling is defined differently.

Let the acceptance condition of $\mathcal{A}$ is $\mathcal{F} = \{ (L_1, K_1), \ldots, (L_k, K_k) \}$. Then the sets L_i, K_i serve as atomic propositions in $\mathcal{D} \otimes \mathcal{A}$. The labeling function L' in $\mathcal{D} \otimes \mathcal{A}$ is the obvious one: if $H \in \{ L_1, \ldots, L_k, K_1, \ldots, K_k \}$, then $H \in L'(\langle s, q \rangle)$ iff $q \in H$.

Accepting BSCC

A BSCC T in $\mathcal{D} \otimes \mathcal{A}$ is *accepting* iff for some index $i \in \{ 1, \ldots, k \}$ we have:

$$T \cap (S \times L_i) = \varnothing \quad \text{and} \quad T \cap (S \times K_i) \neq \varnothing.$$

Thus, once such an accepting BSCC T is reached in $\mathcal{D} \otimes \mathcal{A}$, the acceptance criterion for the DRA $\mathcal{A}$ is fulfilled almost surely.

Verifying DRA properties

Verifying DRA properties

Accepting BSCC

Verifying DRA properties

Accepting BSCC

A BSCC T in $\mathcal{D} \otimes \mathcal{A}$ is *accepting* iff for some index $i \in \{ 1, \ldots, k \}$ we have:

Verifying DRA properties

Accepting BSCC

A BSCC T in $\mathcal{D} \otimes \mathcal{A}$ is *accepting* iff for some index $i \in \{1, \ldots, k\}$ we have:

$$T \cap (S \times L_i) = \varnothing \quad \text{and} \quad T \cap (S \times K_i) \neq \varnothing.$$

Verifying DRA properties

Accepting BSCC

A BSCC T in $\mathcal{D} \otimes \mathcal{A}$ is *accepting* iff for some index $i \in \{1, \ldots, k\}$ we have:

$$T \cap (S \times L_i) = \varnothing \quad \text{and} \quad T \cap (S \times K_i) \neq \varnothing.$$

Thus, once such an accepting BSCC T is reached in $\mathcal{D} \otimes \mathcal{A}$, the acceptance criterion for the DRA $\mathcal{A}$ is fulfilled almost surely.

DRA probabilities = reachability probabilities

Let $\mathcal{D}$ be a finite DTMC, s a state in $\mathcal{D}$, $\mathcal{A}$ a DRA, and let U be the union of all accepting BSCCs in $\mathcal{D} \otimes \mathcal{A}$.

Verifying DRA properties

Accepting BSCC

A BSCC T in $\mathcal{D} \otimes \mathcal{A}$ is *accepting* iff for some index $i \in \{1, \ldots, k\}$ we have:

$$T \cap (S \times L_i) = \varnothing \quad \text{and} \quad T \cap (S \times K_i) \neq \varnothing.$$

Thus, once such an accepting BSCC T is reached in $\mathcal{D} \otimes \mathcal{A}$, the acceptance criterion for the DRA $\mathcal{A}$ is fulfilled almost surely.

DRA probabilities = reachability probabilities

Let $\mathcal{D}$ be a finite DTMC, s a state in $\mathcal{D}$, $\mathcal{A}$ a DRA, and let U be the union of all accepting BSCCs in $\mathcal{D} \otimes \mathcal{A}$. Then:

$$Pr^{\mathcal{D}}(s \models \mathcal{A}) = Pr^{\mathcal{D} \otimes \mathcal{A}}(\langle s, q_s \rangle \models \Diamond U) \quad \text{where} \quad q_s = \delta(q_0, L(s)).$$

Verifying DRA properties

Accepting BSCC

A BSCC T in $\mathcal{D} \otimes \mathcal{A}$ is *accepting* iff for some index $i \in \{1, \ldots, k\}$ we have:

$$T \cap (S \times L_i) = \varnothing \quad \text{and} \quad T \cap (S \times K_i) \neq \varnothing.$$

Thus, once such an accepting BSCC T is reached in $\mathcal{D} \otimes \mathcal{A}$, the acceptance criterion for the DRA $\mathcal{A}$ is fulfilled almost surely.

DRA probabilities = reachability probabilities

Let $\mathcal{D}$ be a finite DTMC, s a state in $\mathcal{D}$, $\mathcal{A}$ a DRA, and let U be the union of all accepting BSCCs in $\mathcal{D} \otimes \mathcal{A}$. Then:

$$Pr^{\mathcal{D}}(s \models \mathcal{A}) = Pr^{\mathcal{D} \otimes \mathcal{A}}(\langle s, q_s \rangle \models \Diamond U) \quad \text{where} \quad q_s = \delta(q_0, L(s)).$$

Proof

On the blackboard (if time permits).

Verifying DRA properties

Accepting BSCC

A BSCC T in $\mathcal{D} \otimes \mathcal{A}$ is *accepting* iff for some index $i \in \{1, \ldots, k\}$ we have:

$$T \cap (S \times L_i) = \varnothing \quad \text{and} \quad T \cap (S \times K_i) \neq \varnothing.$$

Thus, once such an accepting BSCC T is reached in $\mathcal{D} \otimes \mathcal{A}$, the acceptance criterion for the DRA $\mathcal{A}$ is fulfilled almost surely.

DRA probabilities = reachability probabilities

Let $\mathcal{D}$ be a finite DTMC, s a state in $\mathcal{D}$, $\mathcal{A}$ a DRA, and let U be the union of all accepting BSCCs in $\mathcal{D} \otimes \mathcal{A}$. Then:

$$Pr^{\mathcal{D}}(s \models \mathcal{A}) \;=\; Pr^{\mathcal{D} \otimes \mathcal{A}}(\langle s, q_s \rangle \models \Diamond U) \quad \text{where} \quad q_s = \delta(q_0, L(s)).$$

Proof

On the blackboard (if time permits).

Verifying DRA objectives

Verifying DRA objectives

DRA probabilities = reachability probabilities

Let $\mathcal{D}$ be a finite DTMC, s a state in $\mathcal{D}$, $\mathcal{A}$ a DRA, and let U be the union of all accepting BSCCs in $\mathcal{D} \otimes \mathcal{A}$.

Verifying DRA objectives

DRA probabilities = reachability probabilities

Let $\mathcal{D}$ be a finite DTMC, s a state in $\mathcal{D}$, $\mathcal{A}$ a DRA, and let U be the union of all accepting BSCCs in $\mathcal{D} \otimes \mathcal{A}$. Then:

$$Pr^{\mathcal{D}}(s \models \mathcal{A}) \;=\; Pr^{\mathcal{D}\otimes\mathcal{A}}(\, \langle s, q_s \rangle \models \Diamond U \,) \quad \text{where} \quad q_s = \delta(q_0, L(s)).$$

Verifying DRA objectives

DRA probabilities = reachability probabilities

Let $\mathcal{D}$ be a finite DTMC, s a state in $\mathcal{D}$, $\mathcal{A}$ a DRA, and let U be the union of all accepting BSCCs in $\mathcal{D} \otimes \mathcal{A}$. Then:

$$Pr^{\mathcal{D}}(s \models \mathcal{A}) \;=\; Pr^{\mathcal{D}\otimes\mathcal{A}}(\,\langle s, q_s \rangle \,\models\, \Diamond U\,) \quad \text{where} \quad q_s = \delta(q_0, L(s)).$$

Probabilities for satisfying ω-regular properties are obtained by computing the reachability probabilities for accepting BSCCs in $\mathcal{D} \otimes \mathcal{A}$.

Verifying DRA objectives

DRA probabilities = reachability probabilities

Let $\mathcal{D}$ be a finite DTMC, s a state in $\mathcal{D}$, $\mathcal{A}$ a DRA, and let U be the union of all accepting BSCCs in $\mathcal{D} \otimes \mathcal{A}$. Then:

$$Pr^{\mathcal{D}}(s \models \mathcal{A}) \;=\; Pr^{\mathcal{D}\otimes\mathcal{A}}(\,\langle s, q_s\rangle \models \Diamond U\,) \quad \text{where} \quad q_s = \delta(q_0, L(s)).$$

Probabilities for satisfying ω-regular properties are obtained by computing the reachability probabilities for accepting BSCCs in $\mathcal{D} \otimes \mathcal{A}$. Again, a graph analysis and solving systems of linear equations suffice.

Verifying DRA objectives

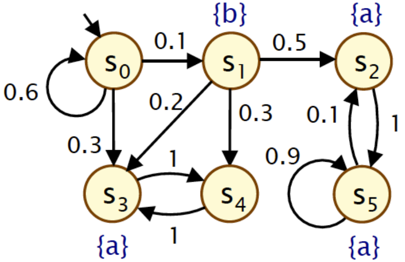
DRA probabilities = reachability probabilities

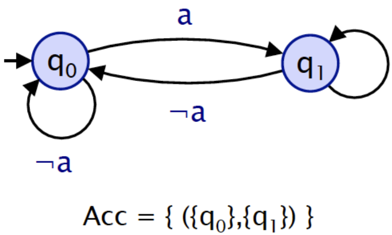
Let $\mathcal{D}$ be a finite DTMC, s a state in $\mathcal{D}$, $\mathcal{A}$ a DRA, and let U be the union of all accepting BSCCs in $\mathcal{D} \otimes \mathcal{A}$. Then:

$$Pr^{\mathcal{D}}(s \models \mathcal{A}) \;=\; Pr^{\mathcal{D}\otimes\mathcal{A}}(\,\langle s, q_s\rangle \,\models\, \Diamond U\,) \quad \text{where} \quad q_s = \delta(q_0, L(s)).$$

Probabilities for satisfying ω-regular properties are obtained by computing the reachability probabilities for accepting BSCCs in $\mathcal{D} \otimes \mathcal{A}$. Again, a graph analysis and solving systems of linear equations suffice. The time complexity is polynomial in the size of $\mathcal{D}$ and $\mathcal{A}$.

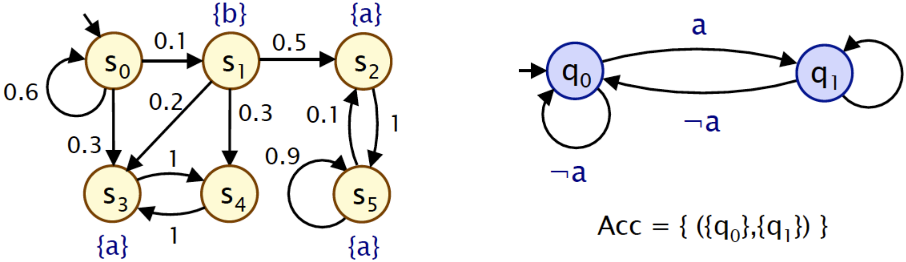
Example: verifying a DTMC versus a DRA





Single accepting BSCC: $\{\,\langle s_2, q_1\rangle, \langle s_5, q_1\rangle\,\}$.

Example: verifying a DTMC versus a DRA



Single accepting BSCC: $\{ \langle s_2, q_1 \rangle, \langle s_5, q_1 \rangle \}$.

Reachability probability is $\frac{1}{2} \cdot \frac{1}{10} \cdot \sum_{k=0}^{\infty} \left(\frac{3}{5} \right)^k \; = \; \frac{1}{8}$.

Measurability

Measurability

Measurability theorem for ω-regular properties **[Vardi 1985]**

For any DTMC $\mathcal{D}$ and DRA $\mathcal{A}$ the set

$$\{\, \pi \in \mathit{Paths}(\mathcal{D}) \mid \mathit{trace}(\pi) \in \mathcal{L}_\omega(\mathcal{A}) \,\}$$

is measurable.

Measurability

Measurability theorem for ω-regular properties **[Vardi 1985]**

For any DTMC $\mathcal{D}$ and DRA $\mathcal{A}$ the set

$$\{\, \pi \in \mathit{Paths}(\mathcal{D}) \mid \mathit{trace}(\pi) \in \mathcal{L}_\omega(\mathcal{A}) \,\}$$

is measurable.

Proof (sketch)

Measurability

Measurability theorem for ω-regular properties [Vardi 1985]

For any DTMC $\mathcal{D}$ and DRA $\mathcal{A}$ the set

$$\{\, \pi \in Paths(\mathcal{D}) \mid trace(\pi) \in \mathcal{L}_\omega(\mathcal{A}) \,\}$$

is measurable.

Proof (sketch)

Let DRA $\mathcal{A}$ with accept sets $\{\, (L_1, K_1), \ldots, (L_m, K_m) \,\}$.

Measurability

Measurability theorem for ω-regular properties [Vardi 1985]

For any DTMC $\mathcal{D}$ and DRA $\mathcal{A}$ the set

$$\{ \pi \in Paths(\mathcal{D}) \mid trace(\pi) \in \mathcal{L}_\omega(\mathcal{A}) \}$$

is measurable.

Proof (sketch)

Let DRA $\mathcal{A}$ with accept sets $\{ (L_1, K_1), \ldots, (L_m, K_m) \}$. Let $\varphi_i = \Diamond \Box \neg L_i \wedge \Box \Diamond K_i$ and Π_i the set of paths satisfying φ_i.

Measurability

Measurability theorem for ω-regular properties [Vardi 1985]

For any DTMC $\mathcal{D}$ and DRA $\mathcal{A}$ the set

$$\{ \pi \in Paths(\mathcal{D}) \mid trace(\pi) \in \mathcal{L}_\omega(\mathcal{A}) \}$$

is measurable.

Proof (sketch)

Let DRA $\mathcal{A}$ with accept sets $\{ (L_1, K_1), \ldots, (L_m, K_m) \}$. Let $\varphi_i = \Diamond \Box \neg L_i \wedge \Box \Diamond K_i$ and Π_i the set of paths satisfying φ_i. Then $\Pi = \Pi_1 \cup \ldots \cup \Pi_k$.

Measurability

Measurability theorem for ω-regular properties **[Vardi 1985]**

For any DTMC $\mathcal{D}$ and DRA $\mathcal{A}$ the set

$$\{ \pi \in Paths(\mathcal{D}) \mid trace(\pi) \in \mathcal{L}_\omega(\mathcal{A}) \}$$

is measurable.

Proof (sketch)

Let DRA $\mathcal{A}$ with accept sets $\{ (L_1, K_1), \ldots, (L_m, K_m) \}$. Let $\varphi_i = \Diamond \Box \neg L_i \wedge \Box \Diamond K_i$ and Π_i the set of paths satisfying φ_i. Then $\Pi = \Pi_1 \cup \ldots \cup \Pi_k$. In addition, $\Pi_i = \Pi_i^{\Diamond\Box} \cap \Pi_i^{\Box\Diamond}$ where $\Pi_i^{\Diamond\Box}$ is the set of paths π in $\mathcal{D}$ such that $\pi^+ \models \Diamond\Box\neg L_i$, and $\Pi_i^{\Box\Diamond}$ is the set of paths π in $\mathcal{D}$ such that $\pi^+ \models \Box\Diamond K_i$.

Measurability

Measurability theorem for ω-regular properties [Vardi 1985]

For any DTMC $\mathcal{D}$ and DRA $\mathcal{A}$ the set

$$\{ \pi \in \mathit{Paths}(\mathcal{D}) \mid \mathit{trace}(\pi) \in \mathcal{L}_\omega(\mathcal{A}) \}$$

is measurable.

Proof (sketch)

Let DRA $\mathcal{A}$ with accept sets $\{ (L_1, K_1), \ldots, (L_m, K_m) \}$. Let $\varphi_i = \Diamond \Box \neg L_i \wedge \Box \Diamond K_i$ and Π_i the set of paths satisfying φ_i. Then $\Pi = \Pi_1 \cup \ldots \cup \Pi_k$. In addition, $\Pi_i = \Pi_i^{\Diamond\Box} \cap \Pi_i^{\Box\Diamond}$ where $\Pi_i^{\Diamond\Box}$ is the set of paths π in $\mathcal{D}$ such that $\pi^+ \models \Diamond\Box\neg L_i$, and $\Pi_i^{\Box\Diamond}$ is the set of paths π in $\mathcal{D}$ such that $\pi^+ \models \Box\Diamond K_i$. It remains to show that $\Pi_i^{\Diamond\Box}$ and $\Pi_i^{\Box\Diamond}$ are measurable.

Measurability

Measurability theorem for ω-regular properties **[Vardi 1985]**

For any DTMC $\mathcal{D}$ and DRA $\mathcal{A}$ the set

$$\{\, \pi \in Paths(\mathcal{D}) \mid trace(\pi) \in \mathcal{L}_\omega(\mathcal{A}) \,\}$$

is measurable.

Proof (sketch)

Let DRA $\mathcal{A}$ with accept sets $\{\, (L_1, K_1), \ldots, (L_m, K_m) \,\}$. Let $\varphi_i = \Diamond \Box \neg L_i \wedge \Box \Diamond K_i$ and Π_i the set of paths satisfying φ_i. Then $\Pi = \Pi_1 \cup \ldots \cup \Pi_k$. In addition, $\Pi_i = \Pi_i^{\Diamond\Box} \cap \Pi_i^{\Box\Diamond}$ where $\Pi_i^{\Diamond\Box}$ is the set of paths π in $\mathcal{D}$ such that $\pi^+ \models \Diamond\Box\neg L_i$, and $\Pi_i^{\Box\Diamond}$ is the set of paths π in $\mathcal{D}$ such that $\pi^+ \models \Box\Diamond K_i$. It remains to show that $\Pi_i^{\Diamond\Box}$ and $\Pi_i^{\Box\Diamond}$ are measurable. This goes along the same lines as proving that $\Diamond\Box\, G$ and $\Box\Diamond\, G$ are measurable.

Linear temporal logic

Linear temporal logic

Linear Temporal Logic: Syntax [Pnueli 1977]

LTL formulas over the set AP obey the grammar:

$$\varphi \ ::= \ a \ \Big| \ \neg\varphi \ \Big| \ \varphi_1 \wedge \varphi_2 \ \Big| \ \bigcirc \varphi \ \Big| \ \varphi_1 \, \mathsf{U} \, \varphi_2$$

where $a \in AP$ and φ, φ_1, and φ_2 are LTL formulas.

Linear temporal logic

Linear Temporal Logic: Syntax [Pnueli 1977]

LTL formulas over the set AP obey the grammar:

$$\varphi \ ::= \ a \ \Big| \ \neg\varphi \ \Big| \ \varphi_1 \wedge \varphi_2 \ \Big| \ \bigcirc \varphi \ \Big| \ \varphi_1 \, \mathsf{U} \, \varphi_2$$

where $a \in AP$ and φ, φ_1, and φ_2 are LTL formulas.

Example

On the blackboard.

LTL semantics

LTL semantics

The LT-property induced by LTL formula φ over AP is:

$Words(\varphi) = \left\{ \sigma \in \left(2^{AP}\right)^{\omega} \mid \sigma \models \varphi \right\}$, where $\models$ is the smallest relation satisfying

LTL semantics

LTL semantics

The LT-property induced by LTL formula φ over AP is:

$$Words(\varphi) = \left\{\sigma \in \left(2^{AP}\right)^{\omega} \mid \sigma \models \varphi\right\},$$ where $\models$ is the smallest relation satisfying

$$\begin{array}{lcll}
\sigma & \models & \text{true} & \\
\sigma & \models & a & \text{iff} \quad a \in A_0 \quad (\text{i.e., } A_0 \models a) \\
\sigma & \models & \varphi_1 \wedge \varphi_2 & \text{iff} \quad \sigma \models \varphi_1 \text{ and } \sigma \models \varphi_2 \\
\sigma & \models & \neg\varphi & \text{iff} \quad \sigma \not\models \varphi \\
\sigma & \models & \bigcirc\varphi & \text{iff} \quad \sigma^1 = A_1 A_2 A_3 \ldots \models \varphi \\
\sigma & \models & \varphi_1 \, \mathsf{U} \, \varphi_2 & \text{iff} \quad \exists j \geqslant 0.\ \sigma^j \models \varphi_2 \text{ and } \sigma^i \models \varphi_1,\ 0 \leqslant i < j
\end{array}$$

for $\sigma = A_0 A_1 A_2 \ldots$ we have $\sigma^i = A_i A_{i+1} A_{i+2} \ldots$ is the suffix of σ from index i on.

Some facts about LTL

Some facts about LTL

LTL is ω-regular

For any LTL formula φ, the set $\mathit{Words}(\varphi)$ is an ω-regular language.

LTL are DRA-definable

For any LTL formula φ, there exists a DRA $\mathcal{A}$ such that $\mathcal{L}_\omega = \mathit{Words}(\varphi)$

Some facts about LTL

LTL is ω-regular

For any LTL formula φ, the set *Words*(φ) is an ω-regular language.

LTL are DRA-definable

For any LTL formula φ, there exists a DRA $\mathcal{A}$ such that $\mathcal{L}_\omega = \mathit{Words}(\varphi)$ where the number of states in $\mathcal{A}$ lies in $2^{2^{|\varphi|}}$.

Verifying a DTMC against LTL formulas

Complexity of LTL model checking **[Vardi 1985]**

The qualitative model-checking problem for finite DTMCs against LTL formula φ is PSPACE-complete, i.e., verifying whether $Pr(s \models \varphi) > 0$ or $Pr(s \models \varphi) = 1$ is PSPACE-complete.

Recall that the LTL model-checking problem for finite transition systems is PSPACE-complete.

Overview

1. Introduction
2. Preliminaries
3. Verifying regular safety properties
4. ω-regular properties
5. Verifying DBA objectives
6. Verifying ω-regular properties
7. **Summary**

Summary

Summary

- Verifying a DTMC $\mathcal{D}$ against a DFA $\mathcal{A}$, i.e., determining $Pr(\mathcal{D} \models \mathcal{A})$, amounts to computing reachability probabilities of accept states in $\mathcal{D} \otimes \mathcal{A}$.

Summary

Summary

- Verifying a DTMC $\mathcal{D}$ against a DFA $\mathcal{A}$, i.e., determining $Pr(\mathcal{D} \models \mathcal{A})$, amounts to computing reachability probabilities of accept states in $\mathcal{D} \otimes \mathcal{A}$.
- For DBA objectives, the probability of infinitely often visiting an accept state in $\mathcal{D} \otimes \mathcal{A}$.

Summary

Summary

- Verifying a DTMC $\mathcal{D}$ against a DFA $\mathcal{A}$, i.e., determining $Pr(\mathcal{D} \models \mathcal{A})$, amounts to computing reachability probabilities of accept states in $\mathcal{D} \otimes \mathcal{A}$.
- For DBA objectives, the probability of infinitely often visiting an accept state in $\mathcal{D} \otimes \mathcal{A}$.
- DBA are strictly less powerful than ω-regular languages.

Summary

Summary

- Verifying a DTMC $\mathcal{D}$ against a DFA $\mathcal{A}$, i.e., determining $Pr(\mathcal{D} \models \mathcal{A})$, amounts to computing reachability probabilities of accept states in $\mathcal{D} \otimes \mathcal{A}$.
- For DBA objectives, the probability of infinitely often visiting an accept state in $\mathcal{D} \otimes \mathcal{A}$.
- DBA are strictly less powerful than ω-regular languages.
- Deterministic Rabin automata are as expressive as ω-regular languages.

Summary

Summary

- Verifying a DTMC $\mathcal{D}$ against a DFA $\mathcal{A}$, i.e., determining $Pr(\mathcal{D} \models \mathcal{A})$, amounts to computing reachability probabilities of accept states in $\mathcal{D} \otimes \mathcal{A}$.
- For DBA objectives, the probability of infinitely often visiting an accept state in $\mathcal{D} \otimes \mathcal{A}$.
- DBA are strictly less powerful than ω-regular languages.
- Deterministic Rabin automata are as expressive as ω-regular languages.
- Verifying DTMC $\mathcal{D}$ agains DRA $\mathcal{A}$ amounts to computing reachability probabilities of accepting BSCCs in $\mathcal{D} \otimes \mathcal{A}$.

Summary

Summary

- Verifying a DTMC $\mathcal{D}$ against a DFA $\mathcal{A}$, i.e., determining $Pr(\mathcal{D} \models \mathcal{A})$, amounts to computing reachability probabilities of accept states in $\mathcal{D} \otimes \mathcal{A}$.
- For DBA objectives, the probability of infinitely often visiting an accept state in $\mathcal{D} \otimes \mathcal{A}$.
- DBA are strictly less powerful than ω-regular languages.
- Deterministic Rabin automata are as expressive as ω-regular languages.
- Verifying DTMC $\mathcal{D}$ agains DRA $\mathcal{A}$ amounts to computing reachability probabilities of accepting BSCCs in $\mathcal{D} \otimes \mathcal{A}$.

Take-home message

Model checking a DTMC against various automata models reduces to computing reachability probabilities in a product.