Efficient Analysis of Unambiguous Automata Using Matrix Semigroup Techniques

S. Kiefer[1] C. Widdershoven[1]

[1]Department of Computer Science
University of Oxford

MFCS 2019

Outline

1. Introduction
   - What is model checking
   - Finite word model checking
   - Infinite word model checking
2. Cuts
3. Pseudo-cuts

Outline

1. Introduction
   - What is model checking
   - Finite word model checking
   - Infinite word model checking
2. Cuts
3. Pseudo-cuts

Model checking

- Calculating possibility of a scenario happening
- Example: “It’s always darkest before the dawn”
- Example: $\mathbf{G}(\mathit{darkest}(x) \rightarrow \mathbf{X}\mathit{dawn}(x))$
- Example:

Model checking

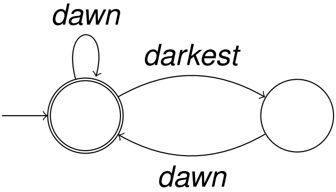
- Calculating possibility of a scenario happening
- Example: “It’s always darkest before the dawn”
- Example: $\mathbf{G}(\mathit{darkest}(x) \to \mathbf{X}\mathit{dawn}(x))$
- Example:

Model checking

- Calculating possibility of a scenario happening
- Example: “It’s always darkest before the dawn”
- Example: $\mathbf{G}(darkest(x) \rightarrow \mathbf{X}dawn(x))$
- Example:

Model checking

- Calculating possibility of a scenario happening
- Example: “It’s always darkest before the dawn”
- Example: $\mathbf{G}(\mathit{darkest}(x) \rightarrow \mathbf{X}\mathit{dawn}(x))$
- Example:



Stochastic models

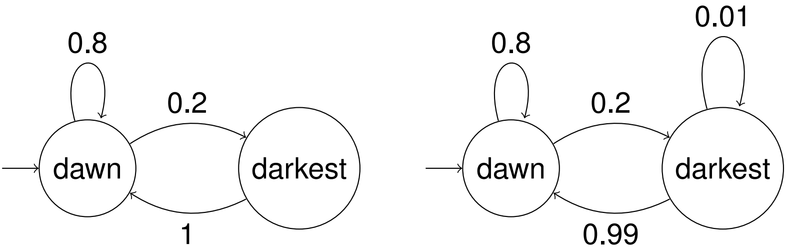
- Probability distribution over sequences of events
- Generally: Markov chains
- Example:

Stochastic models

- Probability distribution over sequences of events
- Generally: Markov chains
- Example:

Stochastic models

- Probability distribution over sequences of events
- Generally: Markov chains
- Example:



Outline

1. Introduction
   - What is model checking
   - Finite word model checking
   - Infinite word model checking

2. Cuts

3. Pseudo-cuts

Finite word model checking

- In general, PSPACE-complete
- For deterministic automata, in P

Finite word model checking

- In general, PSPACE-complete
- For deterministic automata, in P

DFA model checking

- Given a DFA (S, M, δ, q_0, F),
  a Markov chain (P, M), with $P_{m,m'}$ denoting the probability of going to $m' \in M$ from $m \in M$,
  an initial vector $i \in M$ stating the initial distribution of the Markov chain,
  and a vector $\omega \in [0, 1]^M$ denoting the probability of stopping the word after reading a character in M.
- Denote by $\zeta_{s,m}$ the probability of accepting a word starting with m in the DFA starting in s.
- Then the probability of generating an accepted word is $\sum_{m \in M} i_m \zeta_{q_0,m}$.

DFA model checking

- Given a DFA (S, M, δ, q_0, F),
  a Markov chain (P, M), with $P_{m,m'}$ denoting the probability of going to $m' \in M$ from $m \in M$,
  an initial vector $i \in M$ stating the initial distribution of the Markov chain,
  and a vector $\omega \in [0,1]^M$ denoting the probability of stopping the word after reading a character in M.
- Denote by $\zeta_{s,m}$ the probability of accepting a word starting with m in the DFA starting in s.
- Then the probability of generating an accepted word is $\sum_{m \in M} i_m \zeta_{q_0,m}$.

DFA model checking

- Given a DFA (S, M, δ, q_0, F),
  a Markov chain (P, M), with $P_{m,m'}$ denoting the probability of going to $m' \in M$ from $m \in M$,
  an initial vector $i \in M$ stating the initial distribution of the Markov chain,
  and a vector $\omega \in [0, 1]^M$ denoting the probability of stopping the word after reading a character in M.
- Denote by $\zeta_{s,m}$ the probability of accepting a word starting with m in the DFA starting in s.
- Then the probability of generating an accepted word is $\sum_{m \in M} i_m \zeta_{q_0,m}$.

DFA model checking

- Given a DFA (S, M, δ, q_0, F),
  a Markov chain (P, M), with $P_{m,m'}$ denoting the probability of going to $m' \in M$ from $m \in M$,
  an initial vector $i \in M$ stating the initial distribution of the Markov chain,
  and a vector $\omega \in [0, 1]^M$ denoting the probability of stopping the word after reading a character in M.
- Denote by $\zeta_{s,m}$ the probability of accepting a word starting with m in the DFA starting in s.
- Then the probability of generating an accepted word is $\sum_{m \in M} i_m \zeta_{q_0,m}$.

DFA model checking

- Given a DFA (S, M, δ, q_0, F),
  a Markov chain (P, M), with $P_{m,m'}$ denoting the probability of going to $m' \in M$ from $m \in M$,
  an initial vector $i \in M$ stating the initial distribution of the Markov chain,
  and a vector $\omega \in [0, 1]^M$ denoting the probability of stopping the word after reading a character in M.
- Denote by $\zeta_{s,m}$ the probability of accepting a word starting with m in the DFA starting in s.
- Then the probability of generating an accepted word is $\sum_{m \in M} i_m \zeta_{q_0,m}$.

DFA model checking

- Given a DFA (S, M, δ, q_0, F),
  a Markov chain (P, M), with $P_{m,m'}$ denoting the probability of going to $m' \in M$ from $m \in M$,
  an initial vector $i \in M$ stating the initial distribution of the Markov chain,
  and a vector $\omega \in [0, 1]^M$ denoting the probability of stopping the word after reading a character in M.
- Denote by $\zeta_{s,m}$ the probability of accepting a word starting with m in the DFA starting in s.
- Then the probability of generating an accepted word is $\sum_{m \in M} i_m \zeta_{q_0,m}$.

DFA model checking

- Notice that $\zeta_{q_0,m} = \omega(m) + (1-\omega(m)) \sum_{m' \in M} P_{m,m'} \zeta_{\delta(q_0,m),m'}$ if $\delta(q_0, m) \in F$ and $\zeta_{q_0,m} = (1-\omega(m)) \sum_{m' \in M} P_{m,m'} \zeta_{\delta(q_0,m),m'}$ otherwise.
- This leads to the following characterisation for ζ:

$$\zeta_{q,m} = \begin{cases} \omega(m) + (1-\omega(m)) \sum_{m' \in M} P_{m,m'} \zeta_{\delta(q,m),m'} & \text{if } \delta(q,m) \in F \\ (1-\omega(m)) \sum_{m' \in M} P_{m,m'} \zeta_{\delta(q,m),m'} & \text{otherwise.} \end{cases}$$

- Solving this system of equations and calculating $\sum_{m \in M} i_m \zeta_{q_0,m}$ gives us a polynomial algorithm for model checking DFAs.

DFA model checking

- Notice that $\zeta_{q_0,m} = \omega(m) + (1 - \omega(m)) \sum_{m' \in M} P_{m,m'} \zeta_{\delta(q_0,m),m'}$ if $\delta(q_0, m) \in F$ and $\zeta_{q_0,m} = (1 - \omega(m)) \sum_{m' \in M} P_{m,m'} \zeta_{\delta(q_0,m),m'}$ otherwise.
- This leads to the following characterisation for ζ:

$$\zeta_{q,m} = \begin{cases} \omega(m) + (1 - \omega(m)) \sum_{m' \in M} P_{m,m'} \zeta_{\delta(q,m),m'} & \text{if } \delta(q, m) \in F \\ (1 - \omega(m)) \sum_{m' \in M} P_{m,m'} \zeta_{\delta(q,m),m'} & \text{otherwise.} \end{cases}$$

- Solving this system of equations and calculating $\sum_{m \in M} i_m \zeta_{q_0,m}$ gives us a polynomial algorithm for model checking DFAs.

DFA model checking

- Notice that $\zeta_{q_0,m} = \omega(m) + (1 - \omega(m)) \sum_{m' \in M} P_{m,m'} \zeta_{\delta(q_0,m),m'}$ if $\delta(q_0, m) \in F$ and $\zeta_{q_0,m} = (1 - \omega(m)) \sum_{m' \in M} P_{m,m'} \zeta_{\delta(q_0,m),m'}$ otherwise.
- This leads to the following characterisation for ζ:

$$\zeta_{q,m} = \begin{cases} \omega(m) + (1 - \omega(m)) \sum_{m' \in M} P_{m,m'} \zeta_{\delta(q,m),m'} & \text{if } \delta(q,m) \in F \\ (1 - \omega(m)) \sum_{m' \in M} P_{m,m'} \zeta_{\delta(q,m),m'} & \text{otherwise.} \end{cases}$$

- Solving this system of equations and calculating $\sum_{m \in M} i_m \zeta_{q_0,m}$ gives us a polynomial algorithm for model checking DFAs.

Finite word model checking

- In general, PSPACE-complete
- For deterministic automata, in P
- For unambiguous automata, also in P
  The same algorithm works!

Finite word model checking

- In general, PSPACE-complete
- For deterministic automata, in P
- For unambiguous automata, also in P
  The same algorithm works!

- For DFAs:

$$\zeta_{q,m} = \begin{cases} \omega(m) + (1-\omega(m)) \sum_{m' \in M} P_{m,m'} \zeta_{\delta(q,m),m'} & \text{if } \delta(q,m) \in F \\ (1-\omega(m)) \sum_{m' \in M} P_{m,m'} \zeta_{\delta(q,m),m'} & \text{otherwise.} \end{cases}$$

- For UFAs:

$$\zeta_{q,m} = |\delta(q,m) \cap F| \omega(m) + (1-\omega(m)) \sum_{m' \in M} \sum_{q' \in \delta(q,m)} P_{m,m'} \zeta_{q',m'}$$

Outline

1. Introduction
   - What is model checking
   - Finite word model checking
   - Infinite word model checking
2. Cuts
3. Pseudo-cuts

Infinite word model checking

- In general, PSPACE-complete
- For deterministic automata, in P
- For unambiguous automata, in P (Baier et. al., 2016)

Infinite word model checking

- In general, PSPACE-complete
- For deterministic automata, in P
- For unambiguous automata, in P (Baier et. al., 2016)

Infinite word model checking

- In general, PSPACE-complete
- For deterministic automata, in P
- For unambiguous automata, in P (Baier et. al., 2016)

UBA model checking

- Let ζ be the vector of probabilities as defined before.
- Then ζ satisfies

$$\zeta_{q,m} = \sum_{m' \in M} \sum_{q' \in \delta(q,m)} P_{m,m'} \zeta_{q',m'} .$$

- In matrix terms, ζ satisfies $\zeta = B\zeta$, where

$$B_{\langle q,m \rangle, \langle q',m' \rangle} = \begin{cases} P_{m,m'} & \text{if } q' \in \delta(q,m) \\ 0 & \text{otherwise.} \end{cases}$$

UBA model checking

- Let ζ be the vector of probabilities as defined before.
- Then ζ satisfies

$$\zeta_{q,m} = \sum_{m' \in M} \sum_{q' \in \delta(q,m)} P_{m,m'} \zeta_{q',m'} \,.$$

- In matrix terms, ζ satisfies $\zeta = B\zeta$, where

$$B_{\langle q,m \rangle, \langle q',m' \rangle} = \begin{cases} P_{m,m'} & \text{if } q' \in \delta(q,m) \\ 0 & \text{otherwise.} \end{cases}$$

UBA model checking

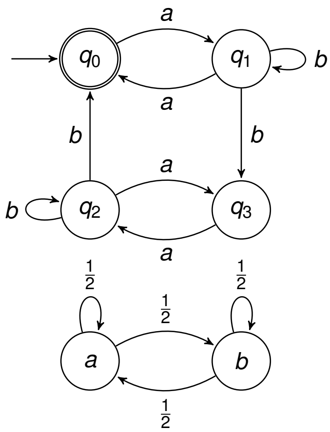
- Let ζ be the vector of probabilities as defined before.
- Then ζ satisfies

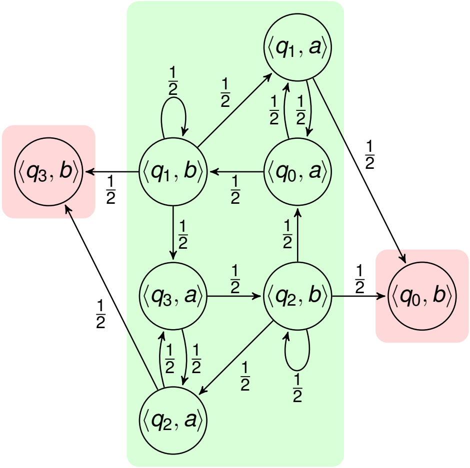
$$\zeta_{q,m} = \sum_{m' \in M} \sum_{q' \in \delta(q,m)} P_{m,m'} \zeta_{q',m'} .$$

- In matrix terms, ζ satisfies $\zeta = B\zeta$, where

$$B_{\langle q,m \rangle, \langle q',m' \rangle} = \begin{cases} P_{m,m'} & \text{if } q' \in \delta(q,m) \\ 0 & \text{otherwise.} \end{cases}$$

Running example





Cuts

- Notice that if the probability of generating an accepted word is greater than zero, then there exist prefixes such that almost every word beginning with that prefix is accepted
- Given such a prefix p for a UBA (S, M, δ, Q_0, F), we see that for any $m \in M$, $\sum_{q_0 \in Q_0} \sum_{s \in \delta(q_0, p)} \zeta_{s,m} \geq 1$. $\delta(q_0, p)$ is called a *cut*.
- But since no two co-reachable states can accept the same words, we see that $\sum_{q_0 \in Q_0} \sum_{s \in \delta(q_0, p)} \zeta_{s,m} = 1$. We call the characteristic vector of a cut a *normaliser*.
- Baier et. al. proved that it suffices to add one such equation for every recurrent SCC in B, equal to 1 if the SCC is accepting.

Cuts

- Notice that if the probability of generating an accepted word is greater than zero, then there exist prefixes such that almost every word beginning with that prefix is accepted
- Given such a prefix p for a UBA (S, M, δ, Q_0, F), we see that for any $m \in M$, $\sum_{q_0 \in Q_0} \sum_{s \in \delta(q_0,p)} \zeta_{s,m} \geq 1$. $\delta(q_0, p)$ is called a *cut*.
- But since no two co-reachable states can accept the same words, we see that $\sum_{q_0 \in Q_0} \sum_{s \in \delta(q_0,p)} \zeta_{s,m} = 1$. We call the characteristic vector of a cut a *normaliser*.
- Baier et. al. proved that it suffices to add one such equation for every recurrent SCC in B, equal to 1 if the SCC is accepting.

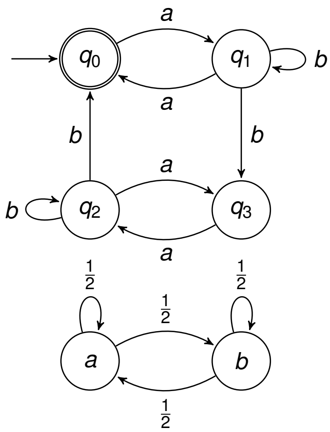
Cuts

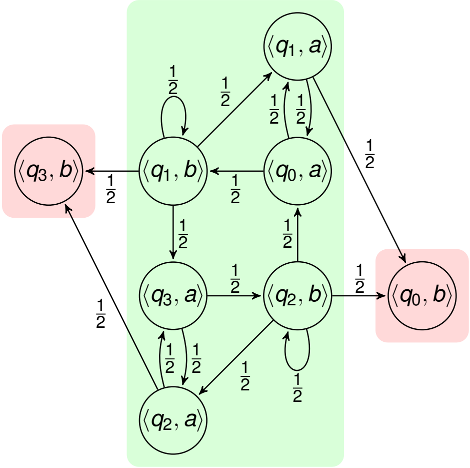
- Notice that if the probability of generating an accepted word is greater than zero, then there exist prefixes such that almost every word beginning with that prefix is accepted
- Given such a prefix p for a UBA (S, M, δ, Q_0, F), we see that for any $m \in M$, $\sum_{q_0 \in Q_0} \sum_{s \in \delta(q_0, p)} \zeta_{s,m} \geq 1$. $\delta(q_0, p)$ is called a *cut*.
- But since no two co-reachable states can accept the same words, we see that $\sum_{q_0 \in Q_0} \sum_{s \in \delta(q_0, p)} \zeta_{s,m} = 1$. We call the characteristic vector of a cut a *normaliser*.
- Baier et. al. proved that it suffices to add one such equation for every recurrent SCC in B, equal to 1 if the SCC is accepting.

Cuts

- Notice that if the probability of generating an accepted word is greater than zero, then there exist prefixes such that almost every word beginning with that prefix is accepted
- Given such a prefix p for a UBA (S, M, δ, Q_0, F), we see that for any $m \in M$, $\sum_{q_0 \in Q_0} \sum_{s \in \delta(q_0, p)} \zeta_{s,m} \geq 1$. $\delta(q_0, p)$ is called a *cut*.
- But since no two co-reachable states can accept the same words, we see that $\sum_{q_0 \in Q_0} \sum_{s \in \delta(q_0, p)} \zeta_{s,m} = 1$. We call the characteristic vector of a cut a *normaliser*.
- Baier et. al. proved that it suffices to add one such equation for every recurrent SCC in B, equal to 1 if the SCC is accepting.

Running example





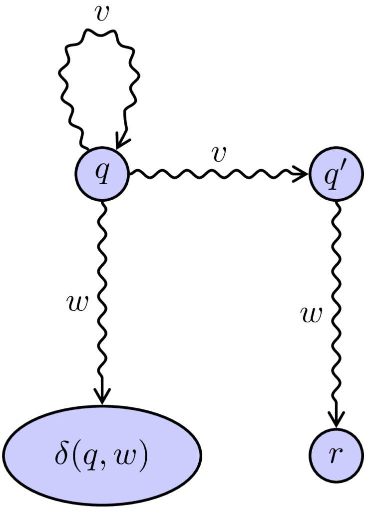
Calculating cuts

- Cuts are calculated using extenders
- Given a state q and a word w such that $\delta(q, w)$ is not a cut, an extender is a word v such that $\delta(q, v) \supseteq \{q, q'\}$, $q \neq q'$, and $\delta(q', w) \neq \emptyset$. Then $\delta(q, vw) \supsetneq \delta(q, w)$.

Calculating cuts

- Cuts are calculated using extenders
- Given a state q and a word w such that $\delta(q, w)$ is not a cut, an extender is a word v such that $\delta(q, v) \supseteq \{q, q'\}$, $q \neq q'$, and $\delta(q', w) \neq \emptyset$. Then $\delta(q, vw) \supsetneq \delta(q, w)$.

Extenders increase set size



Calculating cuts

- Cuts are calculated using extenders
- Given a state q and a word w such that $\delta(q, w)$ is not a cut, an extender is a word v such that $\delta(q, v) \supseteq \{q, q'\}$, $q \neq q'$, and $\delta(q', w) \neq \emptyset$. Then $\delta(q, vw) \supsetneq \delta(q, w)$.
- By repeatedly calculating extenders we get a word p such that almost any infinite word prefixed by p is accepted

Pseudo-cuts

- Cuts remain cuts, cut vectors remain cut vectors
- Sums of cut vectors also invariant under transition matrices
- Matrix semigroups with constant spectral radius have invariant affine plane (Protasov, Voynov, 2017)

Pseudo-cuts

- Cuts remain cuts, cut vectors remain cut vectors
- Sums of cut vectors also invariant under transition matrices
- Matrix semigroups with constant spectral radius have invariant affine plane (Protasov, Voynov, 2017)

Pseudo-cuts

- Cuts remain cuts, cut vectors remain cut vectors
- Sums of cut vectors also invariant under transition matrices
- Matrix semigroups with constant spectral radius have invariant affine plane (Protasov, Voynov, 2017)

Pseudo-cuts

- A *pseudo-cut* is a vector that's invariant under transition matrices
- However, if a is a pseudo-cut, then so are linear multiples of a. Finding a pseudo-cut is not sufficient for finding a normaliser.
- Solution: $Co(d)$*-pseudo-cuts*.

Pseudo-cuts

- A *pseudo-cut* is a vector that's invariant under transition matrices
- However, if a is a pseudo-cut, then so are linear multiples of a. Finding a pseudo-cut is not sufficient for finding a normaliser.
- Solution: *Co*(d)*-pseudo-cuts*.

Pseudo-cuts

- A *pseudo-cut* is a vector that's invariant under transition matrices
- However, if a is a pseudo-cut, then so are linear multiples of a. Finding a pseudo-cut is not sufficient for finding a normaliser.
- Solution: $Co(d)$*-pseudo-cuts*.

Deriving normalisers from $Co(d)$-pseudo-cuts

- $Co(d)$-pseudo-cuts are cuts that are 0 outside of $Co(d)$
- No state in $Co(d)$ apart from d accepts words accepted by d
- Therefore, for a $Co(d)$-pseudo-cut v, $\frac{1}{d_v}d$ is a normaliser

Deriving normalisers from $Co(d)$-pseudo-cuts

- $Co(d)$-pseudo-cuts are cuts that are 0 outside of $Co(d)$
- No state in $Co(d)$ apart from d accepts words accepted by d
- Therefore, for a $Co(d)$-pseudo-cut v, $\frac{1}{d_v} d$ is a normaliser

Deriving normalisers from $Co(d)$-pseudo-cuts

- $Co(d)$-pseudo-cuts are cuts that are 0 outside of $Co(d)$
- No state in $Co(d)$ apart from d accepts words accepted by d
- Therefore, for a $Co(d)$-pseudo-cut v, $\frac{1}{d_v} d$ is a normaliser

Calculating $Co(d)$-pseudo-cuts

- Matrix semigroups with constant spectral radius have invariant affine plane
- Find the affine plane with BFS on M^*
- $Co(d)$-pseudo-cuts are the vectors in the space orthogonal to the plane, intersected with $Co(d)$

Calculating $Co(d)$-pseudo-cuts

- Matrix semigroups with constant spectral radius have invariant affine plane
- Find the affine plane with BFS on M^*
- $Co(d)$-pseudo-cuts are the vectors in the space orthogonal to the plane, intersected with $Co(d)$

Calculating $Co(d)$-pseudo-cuts

- Matrix semigroups with constant spectral radius have invariant affine plane
- Find the affine plane with BFS on M^*
- $Co(d)$-pseudo-cuts are the vectors in the space orthogonal to the plane, intersected with $Co(d)$

Further work

- Finitely ambiguous automata
- (Almost) complementation
- ...