Recent Advances in Model Checking
Practical Course

Jan Křetínský

Technical University of Munich

Summer 2023
Motivation
Example I: Simulation of a die by coins

Knuth & Yao die

Quiz
Is the probability of obtaining 3 equal to \( \frac{1}{6} \)?

Zhang (Saarland University, Germany) Quantitative Model Checking August 24th, 2009
Example 1: Simulation of a die by coins

Knuth & Yao die

Question:
- What is the probability of obtaining 2?
Example II: Zero Configuration Networking (Zeroconf)

- Previously: Manual assignment of IP addresses
- Zeroconf: Dynamic configuration of local IPv4 addresses
- Advantage: Simple devices able to communicate automatically

Automatic Private IP Addressing (APIPA) – RFC 3927

- Used when DHCP is configured but unavailable
- Pick randomly an address from 169.254.1.0 – 169.254.254.255
- Find out whether anybody else uses this address (by sending several ARP requests)

Model:

- Randomly pick an address among the $K$ (65024) addresses.
- With $m$ hosts in the network, collision probability is $q = \frac{m}{K}$.
- Send 4 ARP requests.
- In case of collision, the probability of no answer to the ARP request is $p$ (due to the lossy channel)
Example II: Zero Configuration Networking (Zeroconf)

For 100 hosts and $p = 0.001$, the probability of error is $\approx 1.55 \cdot 10^{-15}$. 
Verification of non-deterministic systems

Controller synthesis for under-specified systems

Given a model $S$ of a system and formula $\phi$, the **model checking problem** is to decide whether $K \models \phi$ (for all/some resolutions of choices).
Verification of non-deterministic systems

Controller synthesis for under-specified systems

Given a model $S$ of a system and formula $\phi$, the model checking problem is to decide whether $K \models \phi$ (for all/some resolutions of choices).

Solution: Combine $K$ and $\phi$ into a “product game graph” $K \times \phi$ with a “winning condition” such that

$$K \models \phi$$

iff

from a designated vertex of $K \times \phi$ player 0 has “winning strategy”.

Aplication II – Synthesis

Alonzo Church, 1957

“Given a requirement which a circuit is to satisfy, we may suppose the requirement expressed in some suitable logistic system which is an extension of restricted recursive arithmetic. The synthesis problem is then to find recursion equivalences representing a circuit that satisfies the given requirement (or alternatively, to determine that there is no such circuit).”
“Given a requirement which a circuit is to satisfy, we may suppose the requirement expressed in some suitable logistic system which is an extension of restricted recursive arithmetic. The synthesis problem is then to find recursion equivalences representing a circuit that satisfies the given requirement (or alternatively, to determine that there is no such circuit).”

Given a requirement on a bit stream transformation

```
input
...1011
```

```
<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
```

```
output
...0100
```

fill the box by a machine with output, satisfying the requirement (or state that the requirement is not satisfiable).
Discrete-time
Markov Decision Processes
MDP
Markov chains – purely probabilistic
Possible successor states are chosen based on probabilities but not on decisions.

We want decisions to model both

▶ controllable setting (game theory, operations theory, control theory);
▶ uncontrollable setting (interleaving in concurrent systems, abstractions of models, open systems)
Definition:
A (labelled) Markov Decision Process (MDP) is a tuple

\[ \mathcal{M} = (S, \text{Act}, P, \pi_0) \]

where
- \( S \) is a countable set of states,
- \( \text{Act} \) is a finite set of actions,
- \( P : S \times \text{Act} \times S \rightarrow [0, 1] \) is the transition probability function, such that for each state \( s \) and action \( \alpha \),
  - \( \sum_{s' \in S} P(s, \alpha, s') = 1 \), then we say that \( \alpha \) is enabled in \( s \); or
  - \( P(s, \alpha, s') = 0 \) for all \( s' \), then we say that \( \alpha \) is not enabled in \( s \).
- \( \pi_0 \) is the initial distribution.

The set of actions enabled in \( s \) is denoted by \( \text{Act}(s) \). We assume that for each \( s \), we have \( \text{Act}(s) \neq \emptyset \).
Problem:
How is the non-determinism resolved?
(Possibly allowing also for memory and randomness)

Definition (Scheduler):
A scheduler (also called strategy or policy) on an MDP $\mathcal{M} = (S, Act, P, \pi_0)$ is a function $\Theta$ assigning to each state $s \in S$ an action $\alpha$ that is enabled in $s$.

Definition (Induced DTMC):
Let $\mathcal{M} = (S, Act, P, \pi_0)$ be a MDP and scheduler $\Theta$ on $\mathcal{M}$. The induced DTMC is given by

$$\mathcal{M}^\Theta = (S, P^\Theta, \pi_0),$$

where

$$P^\Theta(s, s') = P(s, \Theta(s), s')$$
Definition (Scheduler):
A scheduler (also called strategy or policy) on an MDP $\mathcal{M} = (S, \text{Act}, P, \pi_0)$ is a function $\Theta$ assigning to each history $s_0 \cdots s_n \in S^+$ a probability distribution over $\text{Act}$ such that $\alpha$ is enabled in $s_n$ whenever $\Theta(s_0 \cdots s_n)(\alpha) > 0$.

Definition (Induced DTMC):
Let $\mathcal{M} = (S, \text{Act}, P, \pi_0)$ be a MDP and scheduler $\Theta$ on $\mathcal{M}$. The induced DTMC is given by

$$\mathcal{M}^\Theta = (S^+, P^\Theta, \pi_0),$$

where for any $h = s_0 s_1 \cdots s_n$, we define

$$P^\Theta(h, hs_{n+1}) = \sum_{\alpha \in \text{Act}} \Theta(h)(\alpha) \cdot P(s_n, \alpha, s_{n+1})$$
MDP – Reachability
**Min**
When playing “Mensch Ärgere dich nicht” against a fixed opponent strategy, what is the minimal probability of having all pieces kicked out into the outside area?

**Max**
What is the maximal probability of winning the game?
MDP - Reachability

**Min**
- **Best case for reaching undesirable states when controlled**
- **Worst case for reaching desirable states when not controlled**

The **minimum probability to reach** a set of states $B$ from a state $s$ (within $n$ steps) is

\[
\inf_{\Theta} P_s^{\Theta}(\Diamond B), \quad \inf_{\Theta} P_s^{\Theta}(\Diamond \leq n B)
\]

**Max**
- **Best case for reaching desirable states when controlled**
- **Worst case for reaching undesirable states when not controlled**

The **maximum probability to reach** a set of states $B$ from a state $s$ (within $n$ steps) is

\[
\sup_{\Theta} P_s^{\Theta}(\Diamond B), \quad \sup_{\Theta} P_s^{\Theta}(\Diamond \leq n B)
\]

Focus on maximum; minimum is similar
Recall for Markov chains
Let \((S, P, \pi_0)\) be a finite DTMC and \(B \subseteq S\). The vector \(x\) with \(x(s) = P_s(\Diamond B)\) is the unique solution of the equation system:

\[
x(s) = \begin{cases} 
1 & \text{if } s \in B, \\
0 & \text{if } s \in S_0 = \{s \mid P_s(\Diamond B) = 0\}, \\
\sum_{s' \in S} P(s, s') \cdot x(s') & \text{otherwise}.
\end{cases}
\]
Recall for Markov chains

Let \((S, P, \pi_0)\) be a finite DTMC and \(B \subseteq S\). The vector \(x\) with \(x(s) = P_s(\diamond B)\) is the unique solution of the equation system

\[
x(s) = \begin{cases} 
1 & \text{if } s \in B, \\
0 & \text{if } s \in S_0 = \{s \mid P_s(\diamond B) = 0\}, \\
\sum_{s' \in S} P(s, s') \cdot x(s') & \text{otherwise}.
\end{cases}
\]

Theorem (Maximum Reachability Probability):

Let \((S, Act, P, \pi_0)\) be a finite MDP and \(B \subseteq S\). The vector \(x\) with \(x(s) = \sup_{\Theta} P_s^\Theta(\diamond B)\) is the least solution of the equation system

\[
x(s) = \begin{cases} 
1 & \text{if } s \in B, \\
0 & \text{if } s \in S_0^{\max} = \{s \mid \sup_{\Theta} P_s^\Theta(\diamond B) = 0\}, \\
\max_{\alpha \in Act(s)} \sum_{s' \in S} P(s, \alpha, s') \cdot x(s') & \text{otherwise}.
\end{cases}
\]
Let \((S, \text{Act}, P, \pi_0)\) be a finite MDP and \(B \subseteq S\). The vector \(x\) with \(x(s) = \max_\Theta P(\Theta s(\Diamond B))\) is the unique solution of the linear program

\[
\begin{align*}
\text{minimize} & \quad \sum_{s \in S} x(s) \\
\text{satisfying} & \quad x(s) = 1 \quad \forall s \in B, \\
& \quad x(s) = 0 \quad \forall s \in S \setminus (B \cup S) \\
& \quad x(s) \geq \sum_{u \in S} P(s, \alpha, u) \cdot x(u) \quad \forall s \in S \setminus (B \cup S), \quad \forall \alpha \in \text{Act}.
\end{align*}
\]
Linear Program:

Let \((S, \text{Act}, P, \pi_0)\) be a finite MDP and \(B \subseteq S\). The vector \(x\) with \(x(s) = \max_{\Theta} P_s^\Theta(\Diamond B)\) is the unique solution of the linear program satisfying:

\[
x(s) = 1 \quad \forall s \in B,
\]
\[
x(s) = 0 \quad \forall s \in S_{0}^{\text{max}},
\]
\[
x(s) \geq \sum_{u \in S} P(s, \alpha, u) \cdot x(u) \quad \forall s \in S \setminus (B \cup S_{0}^{\text{max}}), \forall \alpha \in \text{Act}.
\]
Linear Program:

Let \((S, \text{Act}, P, \pi_0)\) be a finite MDP and \(B \subseteq S\). The vector \(x\) with \(x(s) = \max_\Theta P_s^\Theta(\Diamond B)\) is the unique solution of the linear program

\[
\begin{align*}
\text{minimize} & \quad \sum_{s \in S} x(s) \\
\text{satisfying} & \quad x(s) = 1 \quad \forall s \in B, \\
& \quad x(s) = 0 \quad \forall s \in S_{0\text{max}}, \\
& \quad x(s) \geq \sum_{u \in S} P(s, \alpha, u) \cdot x(u) \quad \forall s \in S \setminus (B \cup S_{0\text{max}}), \forall \alpha \in \text{Act}.
\end{align*}
\]
Value Iteration Algorithm:
Let $\mathcal{M}$ be a finite MDP with state space $S$, and $B \subseteq S$.
- Initialize $x_0(s)$ to 1 if $s \in B$ and to 0, otherwise.
- Iterate

$$x_{n+1}(s) = \begin{cases} 1 & \text{if } s \in B, \\ 0 & \text{if } s \in S_0^{\max}, \\ \max_{\alpha \in \text{Act}(s)} \sum_{s' \in S} P(s, \alpha, s') \cdot x_n(s') & \text{otherwise} \end{cases}$$

until convergence.
I.e., until $\max_{s \in S} |x_{n+1}(s) - x_n(s)| < \epsilon$ for a small $\epsilon > 0$?
Value Iteration Algorithm:
Let $\mathcal{M}$ be a finite MDP with state space $S$, and $B \subseteq S$.

- Initialize $x_0(s)$ to 1 if $s \in B$ and to 0, otherwise.
- Iterate

$$
\begin{align*}
x_{n+1}(s) &= \begin{cases} 
1 & \text{if } s \in B, \\
0 & \text{if } s \in S_0^{\text{max}}, \\
\max_{\alpha \in \text{Act}(s)} \sum_{s' \in S} P(s, \alpha, s') \cdot x_n(s') & \text{otherwise}
\end{cases}
\end{align*}
$$

until convergence.
I.e., until $\max_{s \in S} |x_{n+1}(s) - x_n(s)| < \epsilon$ for a small $\epsilon > 0$?

Theorem

- $x_n(s) = \sup_{\Theta} P_s^\Theta (\Diamond \leq^n B)$.
- $x_{n+1} \geq x_n$.
- $\lim_{n \to \infty} x_n(s) = \sup_{\Theta} P_s^\Theta (\Diamond B)$. 
We rather compute the set

\[ S_{>0}^{\text{max}} = \{ s \mid \sup_{\Theta} P_s^{\Theta}(\diamond B) > 0 \} \]

and return

\[ S_{0}^{\text{max}} = S \setminus S_{>0}^{\text{max}} \]
We rather compute the set

$$S_{>0}^{\text{max}} = \{ s \mid \sup_{\Theta} P_{s}^{\Theta}(\Box B) > 0 \}$$

and return

$$S_{0}^{\text{max}} = S \setminus S_{>0}^{\text{max}}$$

**$S_{>0}^{\text{max}}$**:

Initialize the set to $B$ and in every iteration add states that reach the set in one step with positive probability for some enabled action. Repeat until fix-point is reached.
We rather compute the set

\[ S_{>0}^{max} = \{ s \mid \sup_{\Theta} P^\Theta_s(\Diamond B) > 0 \} \]

and return

\[ S_0^{max} = S \setminus S_{>0}^{max} \]

\[ S_{>0}^{max} \] .

Initialize the set to \( B \) and in every iteration add states that reach the set in one step with positive probability for some enabled action. Repeat until fix-point is reached.

(Similarly for \( S_{>0}^{min} \):)
We rather compute the set

\[ S_{>0}^{\max} = \{ s \mid \sup_{\Theta} P_s^\Theta (\Diamond B) > 0 \} \]

and return

\[ S_0^{\max} = S \setminus S_{>0}^{\max} \]

\( S_{>0}^{\max} \cdot \)

Initialize the set to \( B \) and in every iteration add states that reach the set in one step with positive probability for some enabled action. Repeat until fix-point is reached.

(Similarly for \( S_{>0}^{\min} \): replace “some” by “every”)
General MDP with end components
General MDP with end components