Antichains: A New Algorithm for Checking Universality of Finite Automata

Laurent Doyen
Université Libre de Bruxelles

Joint work with
Martin De Wulf, Tom Henzinger, Jean-François Raskin

CAV, Seattle, 17th August, 2006
Outline of the talk

- Motivation
- Universality - A Game Approach
- Example
- Experimental Results
- Conclusion
Finite State Automaton

Finite automaton: $A = \langle \text{Loc}, \ell_I, \Sigma, \delta, F \rangle$

with $\delta : \text{Loc} \times \Sigma \rightarrow 2^{\text{Loc}}$ (non-deterministic)

For $w \in \Sigma^*$, we have

\[
\begin{cases} 
    w \in L(A) \text{ iff some path on } w \text{ accepts.} \\
    w \notin L(A) \text{ iff all paths on } w \text{ reject.}
\end{cases}
\]
Language Inclusion and Universality

An implementation $A$ of a program is correct with regard to its specification $B$ if:

$L(A) \subseteq L(B)$

- non-deterministic
- deterministic
Language Inclusion and Universality

\[ L(A) \subseteq L(B) \]

iff \( L(A \cap B^c) \) is empty

- Computing \( B^c \): hard (via determinization)
- Checking emptiness: easy

iff \( L(A^c \cup B) \) is universal

- Computing \( A^c \): easy
- Checking universality: hard
Language Inclusion and Universality

\[ L(A) \subseteq L(B) \]

iff \( L(A \cap B^c) \) is empty

- Computing \( B^c \): hard (via determinization)
- Checking emptiness: easy

iff \( L(A^c \cup B) \) is universal

- Computing \( A^c \): easy
- Checking universality: hard
  
  not so hard in practice with antichains.
Universality - Experimental results

![Graph showing execution time versus number of states for Antichains and dk.bricts.automaton.]
Universality - Experimental results

![Graph showing execution time vs number of states]

- **Execution Time (s)**
- **Number of states**
  - 3000
  - 3500
  - 4000
  - 2500
  - 2000

**Antichains**

**dk.brics.automaton**
Universality - Execution times (in milliseconds)

<table>
<thead>
<tr>
<th>Number of states</th>
<th>20</th>
<th>40</th>
<th>60</th>
<th>80</th>
<th>100</th>
<th>175</th>
<th>500</th>
</tr>
</thead>
<tbody>
<tr>
<td>Determinization</td>
<td>23</td>
<td>50</td>
<td>141</td>
<td>309</td>
<td>583</td>
<td>2257</td>
<td>-</td>
</tr>
<tr>
<td>Antichains</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>14</td>
<td>76</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Number of states</th>
<th>1000</th>
<th>1500</th>
<th>2000</th>
<th>2500</th>
<th>3000</th>
<th>3500</th>
<th>4000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Determinization</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Antichains</td>
<td>400</td>
<td>973</td>
<td>1741</td>
<td>2886</td>
<td>5341</td>
<td>9063</td>
<td>13160</td>
</tr>
</tbody>
</table>
Outline of the talk

- Motivation
- Universality - A Game Approach
- Example
- Experimental Results
- Conclusion
Universality - A game approach

Consider a game played by a protagonist and an antagonist.

The protagonist wants to establish that \( A \) is not universal.

The protagonist has to provide a finite word \( w \) such that no matter how the antagonist reads it using \( A \), the automaton ends up in a rejecting location.

\[ \implies \text{This is a one-shot game.} \]
Universality - A game approach

Consider a game played by a protagonist and an antagonist. The protagonist wants to establish that $A$ is not universal. The protagonist has to provide a finite word $w$ such that no matter how the antagonist reads it using $A$, the automaton ends up in a rejecting location.

Example: Protagonist: $w = 101$
Antagonist: $\pi = \ell_0 \xrightarrow{1} \ell_0 \xrightarrow{0} \ell_2 \xrightarrow{1} \ell_2$

Antagonist wins the play since $\ell_2$ is accepting.
Universality - A game approach

Consider a game played by a protagonist and an antagonist

The protagonist wants to establish that $\mathcal{A}$ is not universal.

The protagonist has to provide a finite word $w$ such that no matter how the antagonist reads it using $\mathcal{A}$, the automaton ends up in a rejecting location.

$\implies$ This is a one-shot game.

Protagonist has a strategy to win this game iff
$\mathcal{A}$ is not universal
Universality - A game approach

Consider a game played by a protagonist and an antagonist.

The protagonist wants to establish that $A$ is not universal.

The game is turn-based:

- **Protagonist** provides a word $w$ one letter at a time;
- **Antagonist** updates the state of $A$ accordingly.
Universality - A game approach

Consider a game played by a protagonist and an antagonist. The protagonist wants to establish that $A$ is not universal. The game is turn-based: the protagonist provides a word $w$ one letter at a time; the antagonist updates the state of $A$ accordingly.

Example: Protagonist: $w = 1$
Antagonist: $\pi = \ell_0 \rightarrow^1 \ell_0$
Universality - A game approach

Consider a game played by a protagonist and an antagonist. The protagonist wants to establish that A is not universal.

The game is turn-based: the protagonist provides a word $w$ one letter at a time; the antagonist updates the state of $A$ accordingly.

**Example:**

Protagonist: $w = 10$

Antagonist: $\pi = \ell_0 \overset{1}{\rightarrow} \ell_0 \overset{0}{\rightarrow} \ell_2$
Universality - A game approach

Consider a game played by a protagonist and an antagonist. The protagonist wants to establish that A is not universal. The game is turn-based: the protagonist provides a word \( w \) one letter at a time; the antagonist updates the state of A accordingly.

**Example:** Protagonist: \( w = 10 \)

Antagonist: \( \pi = \begin{array}{c}
\{l_0\} & \{l_0\} & \{l_1, l_2\}
\end{array} \)
Universality - A game approach

Consider a game played by a protagonist and an antagonist. The protagonist wants to establish that $A$ is not universal. The game is turn-based: Protagonist provides a word $w$ one letter at a time; Antagonist updates the state of $A$ accordingly.

Example: Protagonist: $w = 101$
Antagonist: $\pi = ? \rightarrow ? \rightarrow ? \rightarrow 1 \rightarrow l_2$

Antagonist wins the play since $l_2$ is accepting.
Universality - A game approach

Consider a game played by a protagonist and an antagonist.

The protagonist wants to establish that $A$ is not universal.

The game is turn-based:

- **Protagonist** provides a word $w$ one letter at a time;
- **Antagonist** updates the state of $A$ accordingly.

The protagonist cannot observe the state chosen by the antagonist.

$\implies$ This is a blind game (or game of null information).
Universality - A game approach

Let $\mathcal{A} = \langle \text{Loc}, \ell_I, \Sigma, \delta_A, F \rangle$.

Checking universality of $\mathcal{A}$ is equivalent to solving a blind reachability game $G_T$ with target $T = \text{Loc}/F$. 
Universality - A game approach

Let $A = \langle \text{Loc}, \ell_I, \Sigma, \delta_A, F \rangle$.

Checking universality of $A$ is equivalent to solving a blind reachability game $G_T$ with target $T = \text{Loc} \setminus F$.

Recipe for solving classical reachability games
Universality - A game approach

Let $\mathcal{A} = \langle \text{Loc}, \ell_I, \Sigma, \delta_A, F \rangle$.

Checking universality of $\mathcal{A}$ is equivalent to solving a blind reachability game $G_T$ with target $T = \text{Loc} \setminus F$.

Recipe for solving classical reachability games
Universality - A game approach

Let $\mathcal{A} = \langle \text{Loc}, \ell_I, \Sigma, \delta_A, F' \rangle$.

Checking universality of $\mathcal{A}$ is equivalent to solving a blind reachability game $G_T$ with target $T = \text{Loc}\backslash F$.

Recipe for solving classical reachability games
Universality - A game approach

Let $\mathcal{A} = \langle \text{Loc}, \ell_I, \Sigma, \delta_A, F \rangle$.

Checking universality of $\mathcal{A}$ is equivalent to solving a blind reachability game $G_T$ with target $T = \text{Loc}\backslash F$.

Recipe for solving classical reachability games
Universality - A game approach

Let $\mathcal{A} = \langle \text{Loc}, \ell_I, \Sigma, \delta_A, F \rangle$.

Checking universality of $\mathcal{A}$ is equivalent to solving a blind reachability game $G_T$ with target $T = \text{Loc}\setminus F$.

Recipe for solving classical reachability games

\[
C\text{Pre}(x_0) \quad \text{with} \quad x_0 = T
\]
Universality - A game approach

Let $\mathcal{A} = \langle \text{Loc}, \ell_I, \Sigma, \delta_{\mathcal{A}}, F \rangle$.

Checking universality of $\mathcal{A}$ is equivalent to solving a blind reachability game $G_T$ with target $T = \text{Loc}\backslash F$.

Recipe for solving classical reachability games

\[
x_1 = \text{CPre}(x_0) \cup x_0
\]
Universality - A game approach

Let $\mathcal{A} = \langle \text{Loc}, \ell_I, \Sigma, \delta_A, F \rangle$.

Checking universality of $\mathcal{A}$ is equivalent to solving a blind reachability game $G_T$ with target $T = \text{Loc}\setminus F$.

Recipe for solving classical reachability games
Universality - A game approach

Let $\mathcal{A} = \langle \text{Loc}, \ell_I, \Sigma, \delta_A, F \rangle$.

Checking universality of $\mathcal{A}$ is equivalent to solving a blind reachability game $G_T$ with target $T = \text{Loc} \setminus F$.

Recipe for solving classical reachability games

\[
x_2 = \text{CPre}(x_1) \cup x_1
\]
Universality - A game approach

Let $A = \langle \text{Loc}, \ell_I, \Sigma, \delta_A, F \rangle$.

Checking universality of $A$ is equivalent to solving a blind reachability game $G_T$ with target $T = \text{Loc} \setminus F$.

Recipe for solving classical reachability games

\begin{align*}
CPre(x_{i-1}) & \quad \ldots \quad x_{i-1} \\
\end{align*}
Universality - A game approach

Let $A = \langle \text{Loc}, \ell_I, \Sigma, \delta_A, F \rangle$.

Checking universality of $A$ is equivalent to solving a blind reachability game $G_T$ with target $T = \text{Loc} \setminus F$.

Recipe for solving classical reachability games

Winning states

$W = \mu x. (\text{CPre}(x) \cup T)$
Universality - A game approach

Let $A = \langle \text{Loc}, \ell_I, \Sigma, \delta_A, F \rangle$.

Universality of $A$ is equivalent to a blind reachability game $G_T$ with target $T = \text{Loc}\setminus F$.

Recipe for solving classical reachability games

1. Compute the set of states that are winning in one move: $\text{CPre}(T)$

2. Iterate $\text{CPre}()$: compute $\mathcal{W} = \mu x.(\text{CPre}(x) \cup T)$

3. Check whether $\ell_I \in \mathcal{W}$
Universality - Controllable predecessor operator

Let $\mathcal{A} = \langle \text{Loc}, \ell_I, \Sigma, \delta_A, F \rangle$.

- $\text{CPre}(\cdot)$ should encode the blindness of the game:
  
  “The knowledge of the protagonist is a set of states.”

- $\text{CPre}(T)$ contains all the set of states $s$ such that:
  
  there exists $\sigma \in \Sigma$ such that:
  if protagonist plays $\sigma$ from $s$, then the set $T$ is reached no matter the antagonist's move.

\[
\exists \sigma \in \Sigma \cdot \forall \ell \in s : \delta_A(\ell, \sigma) \subseteq T \quad \text{post}_\sigma(s) \subseteq T
\]
Universality - Controllable predecessor operator

Let $\mathcal{A} = \langle \text{Loc}, \ell_I, \Sigma, \delta_A, F \rangle$.

Consider the following controllable predecessor operator over sets of sets of locations. For $q \subseteq 2^{\text{Loc}}$, let:

$$\text{CPre}(q) = \left\{ s \mid \exists s' \in q \cdot \exists \sigma \in \Sigma : \text{post}_\sigma(s) \subseteq s' \right\}$$

So $s \in \text{CPre}(q)$ if there is a set $s' \in q$ that is reached from any location in $s$, reading input letter $\sigma$.

$\implies$ CPre encodes the blindness of the game.
Universality - A game approach

Let $A = \langle \text{Loc}, \ell_I, \Sigma, \delta_A, F \rangle$.

**Theorem:**

\[
\{\ell_I\} \in \mu x.(\text{CPre}(x) \cup \{T\})
\]

iff

Protagonist has a strategy to win $G_T$

iff

$A$ is not universal

**Claim:** For $s_1 \subseteq s_2$, if $\text{post}_\sigma(s_2) \subseteq s'$ then $\text{post}_\sigma(s_1) \subseteq s'$

\[
s_2 \in \text{CPre}(\cdot) \quad \quad \quad \quad \quad s_1 \in \text{CPre}(\cdot)
\]

Hence, we compute $\subseteq$-downward-closed sets of state sets.

**Idea:** Keep in $\text{CPre}(x)$ only the maximal elements.
Universality - A game approach

Let $\mathcal{A} = \langle \text{Loc}, \ell_I, \Sigma, \delta_{\mathcal{A}}, F \rangle$.

**Definition:**

For $q \subseteq 2^{\text{Loc}}$, let:

$$C_{\text{Pre}}(q) = \text{MaximalSets}\left(\{s \mid \exists s' \in q \cdot \exists \sigma \in \Sigma : \text{post}_\sigma(s) \subseteq s'\}\right)$$

$$= \left[\{s \mid \exists s' \in q \cdot \exists \sigma \in \Sigma : \text{post}_\sigma(s) \subseteq s'\}\right]$$

where $[q] = \{s \in q \mid \exists s' \in q : s \subseteq s'\}$ is an antichain of sets of locations (containing only pairwise $\subseteq$-incomparable elements).
Outline of the talk

- Motivation
- Universality - A Game Approach
- Example
- Experimental Results
- Conclusion
Universality - Example

\[ x_0 = T = \{6, 7\} \]
Universality - Example

\[ x_0 = T = \{6, 7\} \]

\[ x_1 = \text{CPre}(x_0) \cup T = \{4\}_0, \]
Universality - Example

\[ A \]

\[ x_0 = T = \{6, 7\} \]
\[ x_1 = \text{CPre}(x_0) \cup \{T\} = \left[ \{4\}_{0, 1}, \{4, 5\}_1, \{5\}_1, \emptyset \right] \cup \{6, 7\} \]
Universality - Example

\[ x_0 = \quad T = \quad \{6, 7\} \]
\[ x_1 = \quad \text{CPre}(x_0) \cup \{T\} = \quad \{6, 7\}, \{4, 5\} \]
Universality - Example

\[ x_0 = T = \{6, 7\} \]
\[ x_1 = \text{CPre}(x_0) \cup \{T\} = \{6, 7\}, \{4, 5\} \]
\[ x_2 = \text{CPre}(x_1) \cup \{T\} = \{4, 5\}, \{2\}_0 \]
Universality - Example

\[ x_0 = T = \{6, 7\} \]
\[ x_1 = \text{CPre}(x_0) \cup \{T\} = \{6, 7\}, \{4, 5\} \]
\[ x_2 = \text{CPre}(x_1) \cup \{T\} = \left[ \{4, 5\}, \{2\}_{0, 1}, \{2, 3\}_1, \{3\}_1, \emptyset \right] \cup \{6, 7\} \]
Universality - Example

\[ x_0 = T = \{6, 7\} \]
\[ x_1 = \text{CPre}(x_0) \cup \{T\} = \{6, 7\}, \{4, 5\} \]
\[ x_2 = \text{CPre}(x_1) \cup \{T\} = \{6, 7\}, \{4, 5\}, \{2, 3\} \]
Universality - Example

\[ x_0 = T = \{6, 7\} \]
\[ x_1 = \text{CPre}(x_0) \cup \{T\} = \{6, 7\} \cup \{4, 5\} \]
\[ x_2 = \text{CPre}(x_1) \cup \{T\} = \{6, 7\} \cup \{4, 5\} \cup \{2, 3\} \]
\[ x_3 = \text{CPre}(x_2) \cup \{T\} = \left(\{4, 5\}, \{2, 3\}, \{1\}, \emptyset\right) \cup \{6, 7\} \]
Universality - Example

\[
x_0 = T = \{6, 7\}
\]
\[
x_1 = \text{CPre}(x_0) \cup \{T\} = \{6, 7\}, \{4, 5\}
\]
\[
x_2 = \text{CPre}(x_1) \cup \{T\} = \{6, 7\}, \{4, 5\}, \{2, 3\}
\]
\[
x_3 = \text{CPre}(x_2) \cup \{T\} = \{6, 7\}, \{4, 5\}, \{2, 3\}, \{1\}
\]
Universality - Example

\[ x_0 = \{6, 7\} \]
\[ x_1 = \text{CPre}(x_0) \cup \{T\} = \{6, 7, \{4, 5\}\} \]
\[ x_2 = \text{CPre}(x_1) \cup \{T\} = \{6, 7, \{4, 5\}, \{2, 3\}\} \]
\[ x_3 = \text{CPre}(x_2) \cup \{T\} = \{6, 7, \{4, 5\}, \{2, 3\}, \{1\}\} \]
\[ x_4 = \text{CPre}(x_3) \cup \{T\} = x_3 \]
Universality - Example

A

\[ x_0 = T = \{6, 7\} \]
\[ x_1 = \text{CPre}(x_0) \cup \{T\} = \{6, 7\}, \{4, 5\} \]
\[ x_2 = \text{CPre}(x_1) \cup \{T\} = \{6, 7\}, \{4, 5\}, \{2, 3\} \]
\[ x_3 = \text{CPre}(x_2) \cup \{T\} = \{6, 7\}, \{4, 5\}, \{2, 3\}, \{1\} \]
\[ x_4 = \text{CPre}(x_3) \cup \{T\} = x_3 \]

Protagonist has a strategy to win \( G_T \) (e.g.: \( w = 111 \)) \iff A \text{ is not universal}
Universality - Example

We have explored/constructed

instead of
Universality - Determinization
Outline of the talk

- Motivation
- Universality - A Game Approach
- Example
- Experimental Results
- Conclusion
Universality - Experimental results (1)

- We compare our algorithm Antichains with the best\(^{(1)}\) known algorithm dk.brics.automaton by Anders Møller.

\(^{(1)}\) According to "D. Tabakov, M. Y. Vardi. Experimental Evaluation of Classical Automata Constructions. LPAR 2005".

- We use a randomized model to generate the instances (automata of 175 locations). Two parameters:
  - Transition density: \( r \geq 0 \)
  - Density of accepting states: \( 0 \leq f \leq 1 \)
Each sample point: 100 automata with \(|\text{Loc}| = 175, \Sigma = \{0, 1\}\).
Universality - Experimental results (3)

- Transition density: $r = 2$.
- Density of accepting states: $f = 1$. 
## Determinization - Average Number of sets (100 instances)

<table>
<thead>
<tr>
<th># states</th>
<th>20</th>
<th>40</th>
<th>60</th>
<th>80</th>
<th>100</th>
<th>120</th>
<th>140</th>
<th>160</th>
</tr>
</thead>
<tbody>
<tr>
<td>All instances</td>
<td>71</td>
<td>176</td>
<td>415</td>
<td>713</td>
<td>1120</td>
<td>1404</td>
<td>1750</td>
<td>2084</td>
</tr>
<tr>
<td>Univ. inst.</td>
<td>116</td>
<td>388</td>
<td>826</td>
<td>1563</td>
<td>2364</td>
<td>2805</td>
<td>3850</td>
<td>4758</td>
</tr>
<tr>
<td>¬Univ. inst.</td>
<td>11</td>
<td>28</td>
<td>64</td>
<td>98</td>
<td>61</td>
<td>162</td>
<td>32</td>
<td>67</td>
</tr>
</tbody>
</table>

## Antichains - Average Number of sets (same 100 instances)

<table>
<thead>
<tr>
<th># states</th>
<th>20</th>
<th>40</th>
<th>60</th>
<th>80</th>
<th>100</th>
<th>120</th>
<th>140</th>
<th>160</th>
</tr>
</thead>
<tbody>
<tr>
<td>All instances</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td>7</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>12</td>
</tr>
<tr>
<td>Univ. inst.</td>
<td>3</td>
<td>6</td>
<td>7</td>
<td>9</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>19</td>
</tr>
<tr>
<td>¬Univ. inst.</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>5</td>
<td>7</td>
</tr>
</tbody>
</table>
Outline of the talk

- Motivation
- Universality - A Game Approach
- Example
- Experimental Results
- Conclusion
Beyond Universality

- **Universality** ($L(\mathcal{A}) = \Sigma^*$): antichains over $2^{|\text{Loc}_A|}$.

  \[ \text{CPre}(q) = \left\{ s \mid \exists s' \in q \cdot \exists \sigma \in \Sigma : \text{post}_\sigma(s) \subseteq s' \right\} \]

- **Language inclusion** ($L(\mathcal{A}) \subseteq L(\mathcal{B})$): antichains over $\text{Loc}_A \times 2^{|\text{Loc}_B|}$.

  \[ \text{CPre}(q) = \left\{ (\ell, s) \mid \exists (\ell', s') \in q \cdot \exists \sigma \in \Sigma : \ell' \in \delta^A(\ell, \sigma) \wedge \text{post}_\sigma^B(s) \subseteq s' \right\} \]

- **Emptiness of AFA** ($L(\mathcal{A}) = \emptyset$): antichains over $2^{|\text{Loc}_A|}$.

  \[ \text{CPre}(q) = \left\{ s \mid \exists s' \in q \cdot \exists \sigma \in \Sigma : \forall \ell \in s : s' \models \delta(\ell, \sigma) \right\} \]
Conclusion and perspectives

The antichains algorithms apply to:

- Universality of FSA,
- Language inclusion of FSA,
- Emptiness of finite alternating automata.

- ... and soon to automata over infinite words (Büchi)?
  (work in progress)
Thank you

Questions ???