Antichains: An Alternative Approach for Automata-Based Model Checking

Guoqiang Li

School of Software, Shanghai Jiao Tong University

June 4, 2009
- Antichains: A New Algorithm for Checking Universality of Finite Automata. M. De Wulf et al. CAV’06
- Improved Algorithms for the Automata-Based Approach to Model Checking. Laurent Doyen TACAS’07
The Usual Solution for Inclusion Problem

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

- **Complementation:** \( B^c \)
  - Cost: deterministic: low
  - Cost: nondeterministic \( \Rightarrow \) determination: high \( (O(2^n)) \)

- **Intersection:** \( A \cap B^c \)
  - Cost: low

- **Emptiness:** \( L(A \cap B^c) = \emptyset \)
  - Cost: low
  - Determination is the major fact to consume the execution time.
The Usual Solution for Inclusion Problem

\[ L(\mathcal{A}) \subseteq L(\mathcal{B}) \]

- **Complementation**: \( \mathcal{B}^c \)
  - Cost: deterministic: low
  - Cost: nondeterministic \( \implies \) determination: high (\( O(2^n) \))

- **Intersection**: \( \mathcal{A} \cap \mathcal{B}^c \)
  - Cost: low

- **Emptiness**: \( L(\mathcal{A} \cap \mathcal{B}^c) = \emptyset \)
  - Cost: low

- Determination is the major fact to consume the execution time.
The Usual Solution for Inclusion Problem

\[ L(\mathcal{A}) \subseteq L(\mathcal{B}) \]

- **Complementation:** \( \mathcal{B}^c \)
  - Cost: deterministic: low
  - Cost: nondeterministic \( \implies \) determination: high \( (O(2^n)) \)

- **Intersection:** \( \mathcal{A} \cap \mathcal{B}^c \)
  - Cost: low

- **Emptiness:** \( L(\mathcal{A} \cap \mathcal{B}^c) = \emptyset \)
  - Cost: low

- **Determination** is the major fact to consume the execution time.
Determination for NFA: Subset Construction

- **NFA** $A = (Q, \Sigma, \delta, q_0, F)$

- **DFA** $A' = (Q', \Sigma, \delta', q'_0, F')$
  - $Q' = 2^Q$
  - $\delta'(P, a) = \{q \in Q \mid \exists p \in P_1, q \in \delta(p, a)\}$
  - $q'_0 = \{q_0\}$
  - $F' = \{P \in Q' \mid P \cap F \neq \emptyset\}$
Determination for NFA: Subset Construction

We now briefly recall the notion of antichains that were proposed in [77] for solving universality of finite automata. Let $Q$ be a finite set (in [77], $Q$ is a set of states of some finite automaton).

Definition 5.2. An antichain over $Q$ is a set $p \subseteq 2^Q$ such that $\forall s, s' \in p: s \not\subseteq s'$ and $s' \not\subseteq s$. Intuitively, $p$ is a set of pairwise incomparable subsets of $Q$.

Let $L$ be the set of antichains over $Q$. Based on the subset-inclusion relation, we define the partial orders over $L$ as follows:

Definition 5.3. For two antichains $p, p' \in L$, let $p \sqsubseteq p'$ iff $\forall s' \in p': \exists s \in p: s \subseteq s'$. Given a set $p \subseteq 2^Q$ (not necessarily an antichain), a set $s \in p$ is maximal in $p$ iff $\forall s' \in p$: $s \not\subseteq s'$. Similarly, $s \in p$ is minimal in $p$ iff $\forall s' \in p$: $s' \not\subseteq s$. Denote $\lceil p \rceil$ (resp. $\lfloor p \rfloor$) for the set of maximal (resp. minimal) elements of $p$. We are now ready to define the operations over a set of antichains.

Definition 5.4. Given two antichains $p, p' \in L$, the $\sqsubseteq$-lub (least upper bound) is the antichain $p \sqcup p' = \lfloor \{s \cup s' | s \in p \wedge s' \in p' \} \rfloor$ and the $\sqsubseteq$-glb (greatest lower bound) is the antichain $p \sqcap p' = \lfloor \{s | s \in p \lor s \in p' \} \rfloor$. 

49
Determination for NFA: Subset Construction

For Büchi automata, $\text{DBA} \subset \text{NBA}$. Thus, this technique does not work. ($O(2^{n \log n})$)
An Alternative Solution for Inclusion Problem

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

- **Complementation:** \( A^c \)
  - Cost: deterministic: low
- **Union:** \( A^c \cup B \)
  - Cost: low
- **Universality:** \( L(A^c \cup B) = \Sigma^* \)
  - Cost: high
  - The usual way to solve the universality is determination.
  - A new algorithm without determination. Antichain
- Determination VS. Antichain
An Alternative Solution for Inclusion Problem

$L(\mathcal{A}) \subseteq L(\mathcal{B})$

- Complementation: $\mathcal{A}^c$
  - Cost: deterministic: low
- Union: $\mathcal{A}^c \cup \mathcal{B}$
  - Cost: low
- Universality: $L(\mathcal{A}^c \cup \mathcal{B}) = \Sigma^*$
  - Cost: high
  - The usual way to solve the universality is determination.
  - A new algorithm without determination. Antichain

Determinations VS. Antichain
An Alternative Solution for Inclusion Problem

\[ L(\mathcal{A}) \subseteq L(\mathcal{B}) \]

- Complementation: \( \mathcal{A}^c \)
  - Cost: deterministic: low
- Union: \( \mathcal{A}^c \cup \mathcal{B} \)
  - Cost: low
- Universality: \( L(\mathcal{A}^c \cup \mathcal{B}) = \Sigma^* \)
  - Cost: high
  - The usual way to solve the universality is determination.
  - A new algorithm without determination. Antichain
- Determination VS. Antichain
Universality: A Game Approach

- Consider two players: protagonist, antagonist
- **Protagonist** wants to show that $\mathcal{A}$ is not universal.
- **Protagonist** has to provide a finite word $\omega$ such that no matter how the **antagonist** reads it on $\mathcal{A}$, the automaton ends up in a rejecting location.

**Example:**
- **Protagonist:** 101
- **Antagonist:** $l_0 \xrightarrow{1} l_0 \xrightarrow{0} l_2 \xrightarrow{1} l_2$
- Antagonist wins.
- One-shot game.
Universality: A Game Approach

protagonist has a strategy to win the game

iff

\(A\) is not universal
Universality: A Game Approach

- Consider two players: protagonist, antagonist
- Protagonist provides a letter from $\omega$ a time.
- Antagonist updates control locations based on $A$ accordingly.

Example:
- Protagonist: $w = 1$
- Antagonist: $\pi = \ell_0 \rightarrow \ell_0$

Example:
- Protagonist: $1$
- Antagonist: $\ell_0 \xrightarrow{1} \ell_0$
Consider two players: **protagonist**, **antagonist**

- **Protagonist** provides a letter from $\omega$ a time.
- **Antagonist** updates control locations based on $A$ accordingly.

**Example:**
- **protagonist**: 10
- **antagonist**: $l_0 \xrightarrow{1} l_0 \xrightarrow{0} l_2$
Universality: A Game Approach

- Consider two players: protagonist, antagonist
- Protagonist provides a letter from $\omega$ a time.
- Antagonist updates control locations based on $A$ accordingly.

Example:
- protagonist: 10
- antagonist: $l_0 \xrightarrow{1} \{l_0\} \xrightarrow{0} \{l_1, l_2\}$
Universality: A Game Approach

Consider two players: protagonist, antagonist

- Protagonist provides a letter from \( \omega \) a time.
- Antagonist updates control locations based on \( A \) accordingly.

Example:

- Protagonist: 101
- Antagonist:

  \[
  \begin{align*}
  l_0 \overset{1}{\to} \{l_0\} \overset{0}{\to} \{l_1, l_2\} \overset{1}{\to} \{l_2\}
  \end{align*}
  \]

- Turn-based blind game (game with null information).
Solution of the Game

- let $A = (Q, \Sigma, \delta, q_0, F)$
- to solve a blind reachability game $G_T$ with the target $T = Q/F$
Solution of the Game

- let $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$
- to solve a blind reachability game $G_T$ with the target $T = Q/F$
Solution of the Game

- let $A = (Q, \Sigma, \delta, q_0, F)$
- to solve a blind reachability game $G_T$ with the target $T = Q/F$
Solution of the Game

- let $A = (Q, \Sigma, \delta, q_0, F)$
- to solve a blind reachability game $G_T$ with the target $T = Q/F$

$x_0 = T$

$CPre(x_0)$
Solution of the Game

- let $A = (Q, \Sigma, \delta, q_0, F)$
- to solve a blind reachability game $G_T$ with the target $T = Q/F$

$x_1 = \text{CPre}(x_0) \cup x_0$
Solution of the Game

- let $A = (Q, \Sigma, \delta, q_0, F)$
- to solve a blind reachability game $G_T$ with the target $T = Q/F$
Solution of the Game

- let $A = (Q, \Sigma, \delta, q_0, F)$
- to solve a blind reachability game $G_T$ with the target $T = Q/F$

\[ x_2 = \text{CPre}(x_1) \cup x_1 \]
Solution of the Game

▶ let $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$
▶ to solve a blind reachability game $G_T$ with the target $T = Q/F$
Solution of the Game

- let $A = (Q, \Sigma, \delta, q_0, F)$
- to solve a blind reachability game $G_T$ with the target $T = Q/F$

Winning states

$\mathcal{W} = \mu x. (\text{CPre}(x) \cup T)$
Solution of the Game

- let $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$
- to solve a blind reachability game $G_T$ with the target $T = Q/F$

Recipe for solving reachability games

- Compute the set of control locations that are winning in one move $C\text{Pre}^A(T)$
- Iterate $C\text{pre}^A(.)$, compute $\mathcal{W} = \mu x. (C\text{Pre}^A(x) \cup T)$
- Check whether $q_0 \in \mathcal{W}$
Small Formalization: Antichains

- let $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$
- an antichain over $Q$ is a set $q \subseteq 2^Q$, such that $\forall s, s' \in q. s \not\subset s'$. That is, a finite set of pairwise incomparable elements.
- define a monotone function:

  $$CPre^A(q) = \left\lceil \left( \{ s \mid \exists s' \in q, \exists \sigma \in \Sigma, s = cpre^A_{\sigma}(s') \} \right) \right\rceil,$$

  where
  - $cpre^A_{\sigma}(s) = \{ q \in Q \mid \forall q' \in Q, \delta(q, \sigma, q') \rightarrow q' \in s \}$
  - $\left\lceil p \right\rceil$ denotes the maximal elements in $p$
  - $\left\lfloor p \right\rfloor$ denotes the minimal elements in $p$

- To check whether the initial state contained in the greatest fixed point of a lattice of the antichain.
\[ q_0 \subseteq \mu x. (CPre^A(x) \cup T) \]
iff
\[ A \text{ is not universal} \]
Examples
The Lattice of Antichains

- For two antichains $p, p' \in L$, let $p \subseteq p'$ iff $\forall s' \in p', \exists s \in p : s \subseteq s'$
- Given two antichains $p, p' \in L$, the $\sqcup_{\text{lub}}$ is the antichain $p \sqcup p' = \{s \cup s' \mid s \in p \land s' \in p'\}$
- Given two antichains $p, p' \in L$, the $\sqcap_{\text{glb}}$ is the antichain $p \sqcap p' = \{s \mid s \in p \lor s \in p'\}$
- The partial order $\subseteq$ yields a complete lattice on the set $L$ of antichains: $\text{Latt} = \langle L, \subseteq, \sqcup, \sqcap, \emptyset, \{Q\} \rangle$
- It is possible to use least fixed point.
  - It is easy to get a counterexample.
  - we need not compute the final fixed point.
Least Fixed Point and Forward Search

- let $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$
- define a monotone function:
  $\text{CPost}^\mathcal{A}(q) = \lfloor (\{s \mid \exists s' \in q, \exists \sigma \in \Sigma, s = \text{cpost}_\sigma^\mathcal{A}(s')\}) \rfloor$, where
  - $\text{cpost}_\sigma^\mathcal{A}(s) = \{q \in Q \mid \exists q' \in Q, \delta(q', \sigma, q)\}$
  - $\lfloor p \rfloor$ denotes the minimal elements in $p$

  $\exists s \in \mu x. (\text{CPost}^\mathcal{A}(x) \cap \{q_0\}), s \cap F = \emptyset$

  iff

  $\mathcal{A}$ is not universal

- When automaton is not universal, the computation can be stopped as soon as one of the sets does not intersect with $F$. 
Examples

- \( X_0 = \{\{1\}\} \)
- \( X_1 = \text{cpost}(X_0) \cap p_0 = \{\{1\}, \{2\}, \{1, 3\}\} \cap \{\{1\}\} = \emptyset \)
- \( X_2 = \text{cpost}(X_1) \cap p_0 = \{\{1\}, \{2\}, \{1, 3\}, \{5\}, \{4, 5\}\} \cap \{\{1\}\} = \emptyset \)
- \( X_3 = \text{cpost}(X_2) \cap p_0 = \{\{1\}, \{2\}, \{1, 3\}, \{5\}, \{4, 5\}, \{7, 8\}, \{7\}\} \cap \{\{1\}\} = \emptyset \)
- \( \{7\} \cap F = \emptyset \)
Examples

- $X_0 = \{\{1\}\}$
- $X_1 = \text{cpost}(X_0) \cap p_0 = \{\{1\}, \{2\}, \{1, 3\}\} \cap \{\{1\}\} = [\{\{1\}, \{2\}, \{1, 3\}\}] = \{\{1\}, \{2\}\}$
- $X_2 = \text{cpost}(X_1) \cap p_0 = \{\{1\}, \{2\}, \{1, 3\}, \{5\}, \{4, 5\}\} \cap \{\{1\}\} = [\{\{1\}, \{2\}, \{1, 3\}, \{5\}, \{4, 5\}\}] = \{\{1\}, \{2\}, \{5\}\}$
- $X_3 = \text{cpost}(X_2) \cap p_0 = \{\{1\}, \{2\}, \{1, 3\}, \{5\}, \{4, 5\}, \{7, 8\}, \{7\}\} \cap \{\{1\}\} = [\{\{1\}, \{2\}, \{1, 3\}, \{5\}, \{4, 5\}, \{7, 8\}, \{7\}\}] = \{\{1\}, \{2\}, \{5\}, \{7\}\}$
- $\{7\} \cap F = \emptyset$
Examples

- $X_0 = \{\{1\}\}$
- $X_1 = \text{cpost}(X_0) \cap p_0 = \{\{1\}, \{2\}, \{1, 3\}\} \cap \{\{1\}\} = \{\{1\}, \{2\}, \{1, 3\}\} = \{\{1\}, \{2\}\}$
- $X_2 = \text{cpost}(X_1) \cap p_0 = \{\{1\}, \{2\}, \{1, 3\}, \{5\}, \{4, 5\}\} \cap \{\{1\}\} = \{\{1\}, \{2\}, \{1, 3\}, \{5\}, \{4, 5\}\} = \{\{1\}, \{2\}, \{5\}\}$
- $X_3 = \text{cpost}(X_2) \cap p_0 = \{\{1\}, \{2\}, \{1, 3\}, \{5\}, \{4, 5\}, \{7, 8\}, \{7\}\} \cap \{\{1\}\} = \{\{1\}, \{2\}, \{1, 3\}, \{5\}, \{4, 5\}, \{7, 8\}, \{7\}\} = \{\{1\}, \{2\}, \{5\}, \{7\}\}$
- $\{7\} \cap F = \emptyset$
Examples

- \( X_0 = \{\{1\}\} \)
- \( X_1 = \text{cpost}(X_0) \cap p_0 = \{\{1\}, \{2\}, \{1, 3\}\} \cap \{\{1\}\} = [\{\{1\}, \{2\}, \{1, 3\}\}] = \{\{1\}, \{2\}\} \)
- \( X_2 = \text{cpost}(X_1) \cap p_0 = \{\{1\}, \{2\}, \{1, 3\}, \{5\}, \{4, 5\}\} \cap \{\{1\}\} = [\{\{1\}, \{2\}, \{1, 3\}, \{5\}, \{4, 5\}\}] = \{\{1\}, \{2\}, \{5\}\} \)
- \( X_3 = \text{cpost}(X_2) \cap p_0 = \{\{1\}, \{2\}, \{1, 3\}, \{5\}, \{4, 5\}, \{7, 8\}, \{7\}\} \cap \{\{1\}\} = [\{\{1\}, \{2\}, \{1, 3\}, \{5\}, \{4, 5\}, \{7, 8\}, \{7\}\}] = \{\{1\}, \{2\}, \{5\}, \{7\}\} \)
- \( \{7\} \cap F = \emptyset \)
Examples

- $X_0 = \{\{1\}\}$
- $X_1 = \text{cpost}(X_0) \cap p_0 = \{\{1\}, \{2\}, \{1, 3\}\} \cap \{\{1\}\} = [\{\{1\}, \{2\}, \{1, 3\}\}] = \{\{1\}, \{2\}\}$
- $X_2 = \text{cpost}(X_1) \cap p_0 = \{\{1\}, \{2\}, \{1, 3\}, \{5\}, \{4, 5\}\} \cap \{\{1\}\} = [\{\{1\}, \{2\}, \{1, 3\}, \{5\}, \{4, 5\}\}] = \{\{1\}, \{2\}, \{5\}\}$
- $X_3 = \text{cpost}(X_2) \cap p_0 = \{\{1\}, \{2\}, \{1, 3\}, \{5\}, \{4, 5\}, \{7, 8\}, \{7\}\} \cap \{\{1\}\} = [\{\{1\}, \{2\}, \{1, 3\}, \{5\}, \{4, 5\}, \{7, 8\}, \{7\}\}] = \{\{1\}, \{2\}, \{5\}, \{7\}\}$
- $\{7\} \cap F = \emptyset$
We now briefly recall the notion of antichains that were proposed in \cite{77} for solving universality of finite automata. Let $Q$ be a finite set (in \cite{77}, $Q$ is a set of states of some finite automaton).

Definition 5.2. An antichain over $Q$ is a set $p \subseteq 2^Q$ such that $\forall s, s' \in p$: $s \not\subseteq s'$ and $s' \not\subseteq s$. Intuitively, $p$ is a set of pairwise incomparable subsets of $Q$.

Let $L$ be the set of antichains over $Q$. Based on the subset-inclusion relation, we define the partial orders over $L$ as follows:

Definition 5.3. For two antichains $p, p' \in L$, let $p \sqsubseteq p'$ iff $\forall s' \in p' \cdot \exists s \in p$: $s \subseteq s'$.

Given a set $p \subseteq 2^Q$ (not necessarily an antichain), a set $s \in p$ is maximal in $p$ iff $\forall s' \in p$: $s \not\subseteq s'$. Similarly, $s \in p$ is minimal in $p$ iff $\forall s' \in p$: $s' \not\subseteq s$. Denote $\lceil p \rceil$ (resp. $\lfloor p \rfloor$) for the set of maximal (resp. minimal) elements of $p$. We are now ready to define the operations over a set of antichains.

Definition 5.4. Given two antichains $p, p' \in L$, the $\sqsubseteq$-lub (least upper bound) is the antichain $p \sqcup p' = \lfloor \{s \cup s' | s \in p \land s' \in p' \} \rfloor$ and the $\sqsubseteq$-glb (greatest lower bound) is the antichain $p \sqcap p' = \lfloor \{s | s \in p \lor s \in p' \} \rfloor$.

Examples

Figure 5.2: Determinization for checking universality of finite automaton $A$. 
Examples

\begin{itemize}
    \item Given two antichains $p$ and $p'$, let us describe this antichain method by checking universality for the partial orders over $\mathcal{P}(L)$. Let $\mathcal{P}$ be a set of antichains and $\cup$ and $\cap$ be the union and intersection of antichains.
    \item The operations over a set of antichains.
    \item Let $\mathcal{P}$ be a set of antichains and $\cup$ and $\cap$ be the union and intersection of antichains.
    \item Let $\mathcal{P}$ be a set of antichains and $\cup$ and $\cap$ be the union and intersection of antichains.
\end{itemize}

Remark 5.2. Given a set $\mathcal{P}$ of antichains, let $\mathcal{P}$ be a set of antichains and $\cup$ and $\cap$ be the union and intersection of antichains.

Definition 5.3. Given a set $\mathcal{P}$ of antichains, let $\mathcal{P}$ be a set of antichains and $\cup$ and $\cap$ be the union and intersection of antichains.

Definition 5.2. Given a set $\mathcal{P}$ of antichains, let $\mathcal{P}$ be a set of antichains and $\cup$ and $\cap$ be the union and intersection of antichains.

\begin{itemize}
    \item Let $\mathcal{P}$ be a set of antichains and $\cup$ and $\cap$ be the union and intersection of antichains.
    \item Let $\mathcal{P}$ be a set of antichains and $\cup$ and $\cap$ be the union and intersection of antichains.
\end{itemize}

Examples

\begin{itemize}
    \item Given two antichains $p$ and $p'$, let us describe this antichain method by checking universality for the partial orders over $\mathcal{P}(L)$. Let $\mathcal{P}$ be a set of antichains and $\cup$ and $\cap$ be the union and intersection of antichains.
    \item The operations over a set of antichains.
    \item Let $\mathcal{P}$ be a set of antichains and $\cup$ and $\cap$ be the union and intersection of antichains.
    \item Let $\mathcal{P}$ be a set of antichains and $\cup$ and $\cap$ be the union and intersection of antichains.
\end{itemize}

Remark 5.2. Given a set $\mathcal{P}$ of antichains, let $\mathcal{P}$ be a set of antichains and $\cup$ and $\cap$ be the union and intersection of antichains.

Definition 5.3. Given a set $\mathcal{P}$ of antichains, let $\mathcal{P}$ be a set of antichains and $\cup$ and $\cap$ be the union and intersection of antichains.

Definition 5.2. Given a set $\mathcal{P}$ of antichains, let $\mathcal{P}$ be a set of antichains and $\cup$ and $\cap$ be the union and intersection of antichains.

\begin{itemize}
    \item Let $\mathcal{P}$ be a set of antichains and $\cup$ and $\cap$ be the union and intersection of antichains.
    \item Let $\mathcal{P}$ be a set of antichains and $\cup$ and $\cap$ be the union and intersection of antichains.
\end{itemize}
Two algorithms,
  - Backwards, greatest fixed point,
  - Forwards, least fixed point,
are equivalent.
Experimental Result (CAV’06)

Universality - Experimental results

![Graph showing the relationship between the number of states and execution time for Antichains and dk.bricks.automaton.](image-url)
Experimental Result (CAV’06)

Universality - Experimental results

![Graph showing execution time vs. number of states for different algorithms.](image)

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

- **Antichains**
- **dk.brics.automaton**
Inclusion Problem of two NFAs

- Let $A = (Q_A, \Sigma, \delta_A, q^0_A, F_A)$ and $B = (Q_B, \Sigma, \delta_B, q^0_B, F_B)$
- The language inclusion can be checked using an antichain algorithm based on a richer lattice.
- An antichain $q$ over $Q_A \times 2^{Q_B}$ is a set such that for all $\forall (q_1, s_1), (q_2, s_2) \in p. (q_1 = q_2) \land (s_1 \neq s_2) \rightarrow s_1 \not\subseteq s_2 \land s_2 \not\subseteq s_1$
A Collection of Researches on Antichains

- for finite automata, Wulf et. al. CAV’06
- for finite tree automata, Bouajjani et. al. CIAA’08
- for Büchi automata, Doyen et. al. TACAS’07
- for visibly pushdown automata, Ogawa et. al. ???’09

- for event-clock timed automata
- for event-clock timed Büchi automata
- for event-clock visibly pushdown timed automata (meaningful?)
A Collection of Researches on Antichains

- for finite automata, Wulf et. al. CAV’06
- for finite tree automata, Bouajjani et. al. CIAA’08
- for Büchi automata, Doyen et. al. TACAS’07
- for visibly pushdown automata, Ogawa et. al. ???’09

still left, time issues

- for event-clock timed automata
- for event-clock timed Büchi automata
- for event-clock visibly pushdown timed automata (meaningful?)
Conclusion

- Antichain algorithm does not improve the complexity in theory, it is significant in practice.
  - e.g. there are no implementations of complementing Büchi automata, but now ALASKA
- In a model checking view, antichain algorithm adopts “lazy determination”
  - In lazy model checking, a model is expanded only when needed.
  - In antichain algorithm, an automaton is determinized only when needed.