# Antichain algorithms.

Using Antichains to solve reachabillity problems
on non-deterministic finite automata.

Albert-Ludwigs-Universität Freiburg

Samuel Roth

Proseminar on Automata Theory at the chair of Software Engineering.
Supervised by Alexander Nutz.

# Content

# Content

# Partial orders

- $V$ be a finite set
- $\preceq$ a binary relation $\preceq \subseteq V \times V$
- $\preceq$ reflexive, transitive and anti-symmetric then it is called a partial order
- $(\preceq, V)$ is called a partially ordered set.

# Partial orders

- $V$ be a finite set
- $\preceq$ a binary relation $\preceq \subseteq V \times V$
- $\preceq$ reflexive, transitive and anti-symmetric then it is called a partial order
- $(\preceq, V)$ is called a partially ordered set.

## Example

- $(\leq, \{1, 2, 3, 42\})$, the $\leq$ order of the natural numbers.

# Partial orders

- $V$ be a finite set
- $\preceq$ a binary relation $\preceq \subseteq V \times V$
- $\preceq$ reflexive, transitive and anti-symmetric then it is called a partial order
- $(\preceq, V)$ is called a partially ordered set.

**Example**

- $(\leq, \{1,2,3,42\})$, the $\leq$ order of the natural numbers.
- $(\subseteq, 2^V))$, subset-inclusion in a powerset.

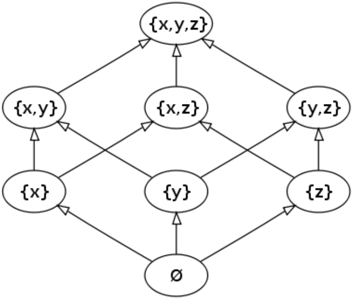
# Content

# Antichain

- subsets of $V$ pairwise *incompatible* with regard to $\preceq$.

# Antichain

- subsets of $V$ pairwise *incompatible* with regard to $\preceq$.

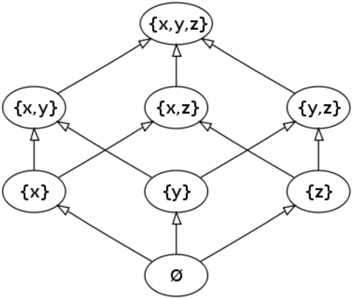



**Example**

- Powerset of $\{x, y, z\}$ with $\subseteq$ as partial order.

# Antichain

- subsets of $V$ pairwise *incompatible* with regard to $\preceq$.

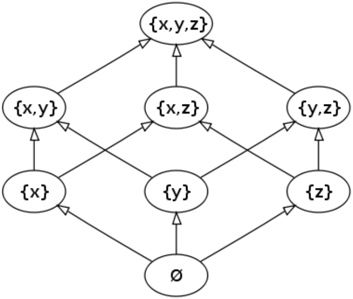



**Example**

- Powerset of $\{x,y,z\}$ with $\subseteq$ as partial order.
- $\{\{x,y\},\{x,z\}\}$ is a antichain.

# Antichain

- subsets of $V$ pairwise *incompatible* with regard to $\preceq$.




**Example**

- Powerset of $\{x,y,z\}$ with $\subseteq$ as partial order.
- $\{\{x,y\},\{x,z\}\}$ is a antichain.
- $\{\{x\},\{x,z\}\}$ is not a antichain.

# Content

# Downward closure, Maximum of $S \subseteq V$

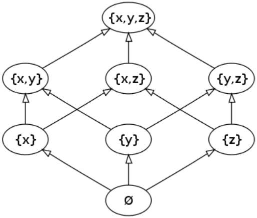
## Downward closure

$Down(\preceq, S) := \{v' \in V \mid \exists v \in S v' \preceq v\}$

# Downward closure, Maximum of $S \subseteq V$

## Downward closure

$Down(\preceq, S) := \{v' \in V \mid \exists v \in S v' \preceq v\}$




## Examples

- $Down(\subseteq, \{x, y\}) = \{\emptyset, \{x\}, \{y\}, \{x, y\}\}$

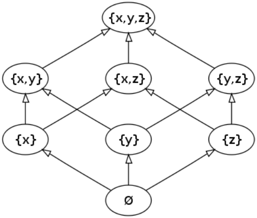
# Downward closure, Maximum of $S \subseteq V$

## Downward closure

$Down(\preceq, S) := \{v' \in V \mid \exists v \in S v' \preceq v\}$

## Maximum

$Max(\preceq, S) := \{v \in S \mid \forall v' \in S : v \preceq v' \Rightarrow v' \preceq v\}$




## Examples

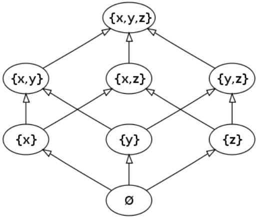
# Downward closure, Maximum of $S \subseteq V$

## Downward closure

$Down(\preceq, S) := \{v' \in V \mid \exists v \in S v' \preceq v\}$

## Maximum

$Max(\preceq, S) := \{v \in S \mid \forall v' \in S : v \preceq v' \Rightarrow v' \preceq v\}$




## Examples

- $Max(\subseteq, \{\{x\}, \{x,y\}, \{x,z\}\}) = \{\{x,y\}, \{x,z\}\}$

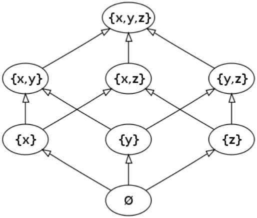
# Downward closure, Maximum of $S \subseteq V$

## Downward closure

$Down(\preceq, S) := \{v' \in V \mid \exists v \in S v' \preceq v\}$

## Maximum

$Max(\preceq, S) := \{v \in S \mid \forall v' \in S : v \preceq v' \Rightarrow v' \preceq v\}$




## Examples

- $Max(\subseteq, \{\{x\}, \{x,y\}, \{x,z\}\}) = \{\{x,y\}, \{x,z\}\}$
- $Max(\subseteq, \{\{x\}, \{x,y\}, \{x,z\}, \{x,y,z\}\}) = \{\{x,y,z\}\}$

# Content

# Antichain as a representation for a downward closed set $S \subseteq V$.

- Use $S' := Max(\preceq, S)$ to represent $S$.

# Antichain as a representation for a downward closed set $S \subseteq V$.

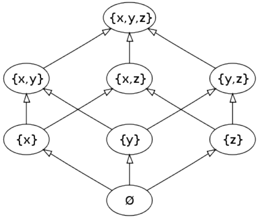
- Use $S' := Max(\preceq, S)$ to represent $S$.
- The question $v \in S$ becomes $\exists v' \in S' : v \preceq v'$

# Antichain as a representation for a downward closed set $S \subseteq V$.

- Use $S' := Max(\preceq, S)$ to represent $S$.
- The question $v \in S$ becomes $\exists v' \in S' : v \preceq v'$

**Example**

$S_1 := \{\emptyset, \{x\}, \{y\}\}$ so $S'_1 = \{\{x\}, \{y\}\}$
$S_2 := \{\emptyset, \{x\}, \{y\}, \{x, y\}\}$ so $S'_2 = \{\{x, y\}\}$



# Content

# Powerset determinization of Non-deterministic finite automata

- Let $A := (Loc, Init, Fin, \delta, \Sigma)$ be a finite automaton.
- $G(A) := (V, E, In, \overline{Fin})$ is the corresponding powerset automaton.

# Powerset determinization of Non-deterministic finite automata

- Let $A := (Loc, Init, Fin, \delta, \Sigma)$ be a finite automaton.
- $G(A) := (V, E, In, \overline{Fin})$ is the corresponding powerset automaton.
- $V := 2^{Loc}$

# Powerset determinization of Non-deterministic finite automata

- Let $A := (Loc, Init, Fin, \delta, \Sigma)$ be a finite automaton.
- $G(A) := (V, E, In, \overline{Fin})$ is the corresponding powerset automaton.
- $V := 2^{Loc}$
- $In := \{v \in 2^{Loc} | Init \in v\}$

# Powerset determinization of Non-deterministic finite automata

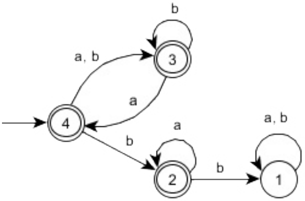
- Let $A := (Loc, Init, Fin, \delta, \Sigma)$ be a finite automaton.
- $G(A) := (V, E, In, \overline{Fin})$ is the corresponding powerset automaton.
- $V := 2^{Loc}$
- $In := \{v \in 2^{Loc} | Init \in v\}$
- $\overline{Fin} := \{v \in 2^{Loc} | v \subseteq Loc \setminus Fin\}$

# Powerset determinization of Non-deterministic finite automata

- Let $A := (Loc, Init, Fin, \delta, \Sigma)$ be a finite automaton.
- $G(A) := (V, E, In, \overline{Fin})$ is the corresponding powerset automaton.
- $V := 2^{Loc}$
- $In := \{v \in 2^{Loc} | Init \in v\}$
- $\overline{Fin} := \{v \in 2^{Loc} | v \subseteq Loc \setminus Fin\}$
- $(v_1, v_2) \in E$ iff there exists a $\sigma \in \Sigma$ such that $\bigcup_{q \in v_1} \delta(q, \sigma) = v_2$.

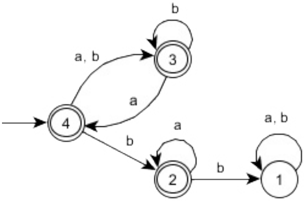
# Example $A := (Loc, Init, Fin, \delta, \Sigma)$ to $G(A) := (V, E, In, \overline{Fin})$

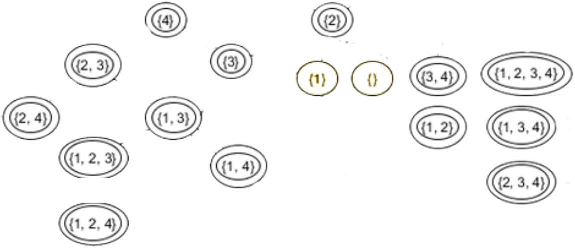
- $V := 2^{Loc}$
- $\overline{Fin} := \{v \in 2^{Loc} | v \subseteq Loc \setminus Fin\}$
- $In := \{v \in 2^{Loc} | Init \in v\}$
- $(v_1, v_2) \in E$ iff there exists a $\sigma \in \Sigma$ such that $\bigcup_{q \in v_1} \delta(q, \sigma) = v_2$.



# Example $A := (Loc, Init, Fin, \delta, \Sigma)$ to $G(A) := (V, E, In, \overline{Fin})$

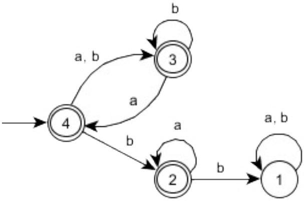
- $V := 2^{Loc}$
- $\overline{Fin} := \{v \in 2^{Loc} | v \subseteq Loc \setminus Fin\}$
- $In := \{v \in 2^{Loc} | Init \in v\}$
- $(v_1, v_2) \in E$ iff there exists a $\sigma \in \Sigma$ such that $\bigcup_{q \in v_1} \delta(q, \sigma) = v_2$.

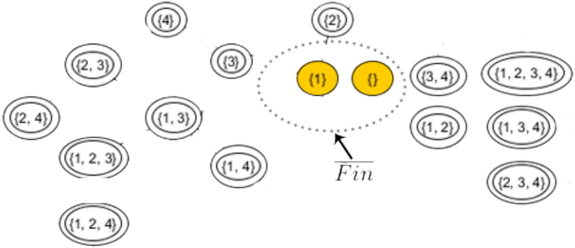


# Example $A := (Loc, Init, Fin, \delta, \Sigma)$ to $G(A) := (V, E, In, \overline{Fin})$

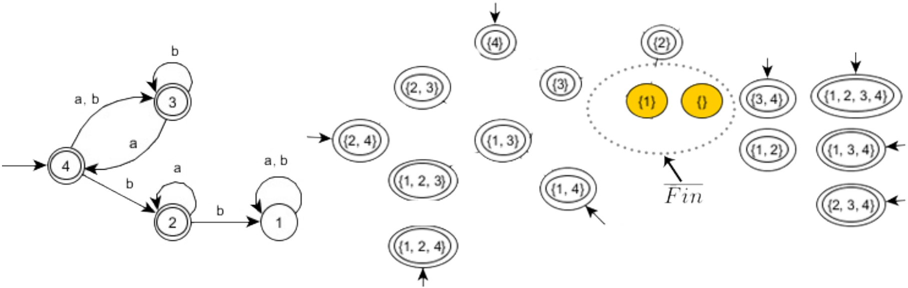
- $V := 2^{Loc}$
- $\overline{Fin} := \{v \in 2^{Loc} | v \subseteq Loc \setminus Fin\}$
- $In := \{v \in 2^{Loc} | Init \in v\}$
- $(v_1, v_2) \in E$ iff there exists a $\sigma \in \Sigma$ such that $\bigcup_{q \in v_1} \delta(q, \sigma) = v_2$.



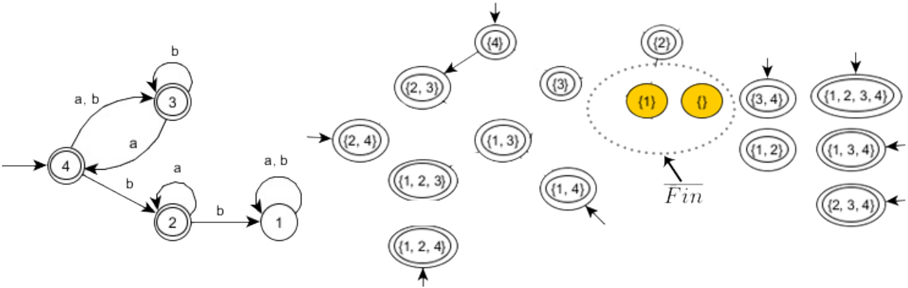
# Example $A := (Loc, Init, Fin, \delta, \Sigma)$ to $G(A) := (V, E, In, \overline{Fin})$

- $V := 2^{Loc}$
- $\overline{Fin} := \{v \in 2^{Loc} | v \subseteq Loc \setminus Fin\}$
- $In := \{v \in 2^{Loc} | Init \in v\}$
- $(v_1, v_2) \in E$ iff there exists a $\sigma \in \Sigma$ such that $\bigcup_{q \in v_1} \delta(q, \sigma) = v_2$.



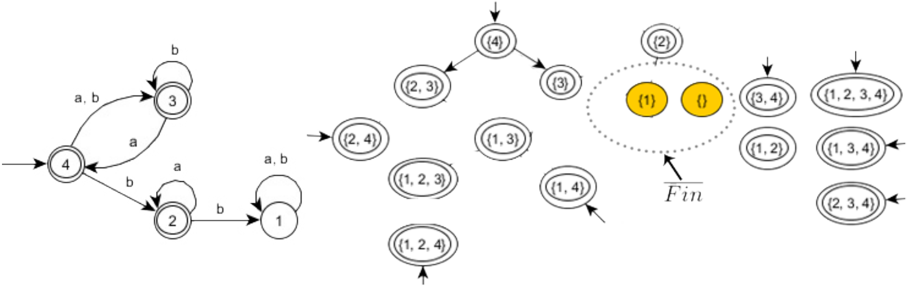
# Example $A := (Loc, Init, Fin, \delta, \Sigma)$ to $G(A) := (V, E, In, \overline{Fin})$

- $V := 2^{Loc}$
- $\overline{Fin} := \{v \in 2^{Loc} | v \subseteq Loc \setminus Fin\}$
- $In := \{v \in 2^{Loc} | Init \in v\}$
- $(v_1, v_2) \in E$ iff there exists a $\sigma \in \Sigma$ such that $\bigcup_{q \in v_1} \delta(q, \sigma) = v_2$.



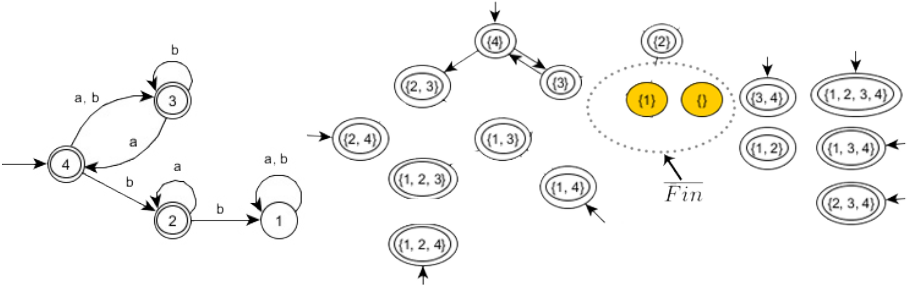
# Example $A := (Loc, Init, Fin, \delta, \Sigma)$ to $G(A) := (V, E, In, \overline{Fin})$

- $V := 2^{Loc}$
- $\overline{Fin} := \{v \in 2^{Loc} | v \subseteq Loc \setminus Fin\}$
- $In := \{v \in 2^{Loc} | Init \in v\}$
- $(v_1, v_2) \in E$ iff there exists a $\sigma \in \Sigma$ such that $\bigcup_{q \in v_1} \delta(q, \sigma) = v_2$.



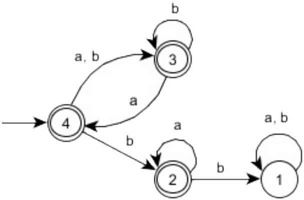
# Example $A := (Loc, Init, Fin, \delta, \Sigma)$ to $G(A) := (V, E, In, \overline{Fin})$

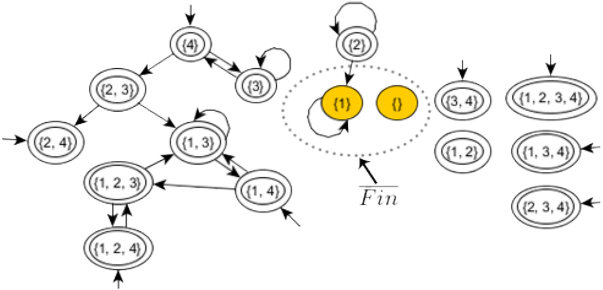
- $V := 2^{Loc}$
- $\overline{Fin} := \{v \in 2^{Loc} | v \subseteq Loc \setminus Fin\}$
- $In := \{v \in 2^{Loc} | Init \in v\}$
- $(v_1, v_2) \in E$ iff there exists a $\sigma \in \Sigma$ such that $\bigcup_{q \in v_1} \delta(q, \sigma) = v_2$.



# Example $A := (Loc, Init, Fin, \delta, \Sigma)$ to $G(A) := (V, E, In, \overline{Fin})$

- $V := 2^{Loc}$
- $\overline{Fin} := \{v \in 2^{Loc} | v \subseteq Loc \setminus Fin\}$
- $In := \{v \in 2^{Loc} | Init \in v\}$
- $(v_1, v_2) \in E$ iff there exists a $\sigma \in \Sigma$ such that $\bigcup_{q \in v_1} \delta(q, \sigma) = v_2$.



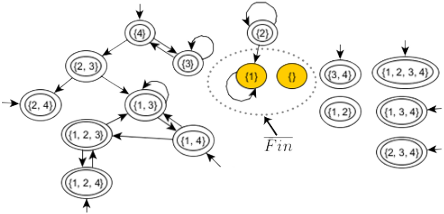
# Content

# Reachability problem in $G(A) = (V, E; In, \overline{Fin})$

- Asks if a subset $S \subseteq V$ is reachable from $In$.
- Where *Reachable* here means there is a *path* from $In$ to $S$. This is $v_1, \cdots v_n$ such that $(v_i, v_{i+1}) \in E$ for all $0 < i < n$ and $v_1 \in In$ and $v_n \in S$.

# Reachability problem in $G(A) = (V, E; In, \overline{Fin})$

- Asks if a subset $S \subseteq V$ is reachable from *In*.
- Where *Reachable* here means there is a *path* from *In* to $S$. This is $v_1, \cdots v_n$ such that $(v_i, v_{i+1}) \in E$ for all $0 < i < n$ and $v_1 \in In$ and $v_n \in S$.




**Example**

$S := \{\{1\}, \{1,2,3\}\}$ is reachable from *In*.

# The predecessors of $S \subseteq V$ in $G(A) = (V, E; In, \overline{Fin})$

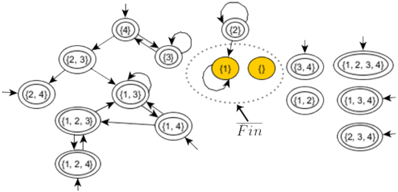
- The predecessors of $S \subseteq V$ are

$$pre(S) := \{v_1 \in V \mid \exists v_2 \in S : (v_1, v_2) \in E\}$$

# The predecessors of $S \subseteq V$ in $G(A) = (V, E; In, \overline{Fin})$

- The predecessors of $S \subseteq V$ are

$$pre(S) := \{v_1 \in V \mid \exists v_2 \in S : (v_1, v_2) \in E\}$$




## Example

$S := \{\{1\}, \{1,2,3\}\}$ then $pre(S) = \{\{1\}, \{2\}, \{1,2,4\}, \{1,4\}\}$.

# Content

# Backward reachability fixpoint algorithm in $G(A) = (V, E; In, \overline{Fin})$

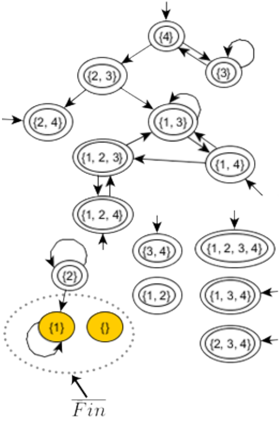
- Solves the reachability problem for $S \subseteq V$ by computing the monotone growing sequence of sets

$$B_0 = S; B_i = B_{i-1} \cup pre(B_{i-1})$$

# Backward reachability fixpoint algorithm in $G(A) = (V, E; In, \overline{Fin})$

## Example starting with $\overline{Fin}$

$B_0 = S; B_i = B_{i-1} \cup pre(B_{i-1})$

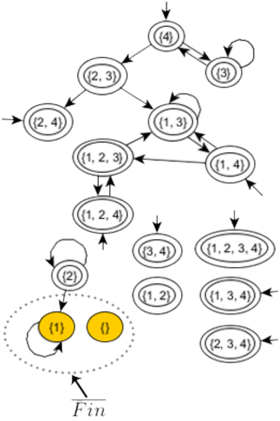


# Backward reachability fixpoint algorithm in $G(A) = (V, E; In, \overline{Fin})$

## Example starting with $\overline{Fin}$

$B_0 = S; B_i = B_{i-1} \cup pre(B_{i-1})$

$B_0 = \{\{\}, \{1\}\}$



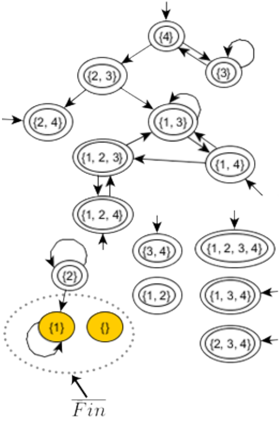
# Backward reachability fixpoint algorithm in $G(A) = (V, E; In, \overline{Fin})$

## Example starting with $\overline{Fin}$

$B_0 = S; B_i = B_{i-1} \cup pre(B_{i-1})$

$B_0 = \{\{\}, \{1\}\}$

$B_1 = B_0 \cup pre(B_0)$

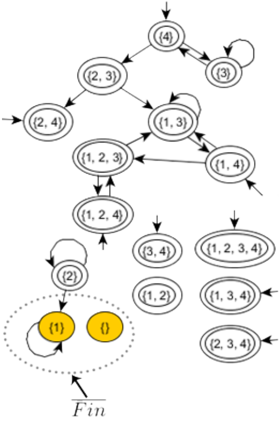


# Backward reachability fixpoint algorithm in $G(A) = (V, E; In, \overline{Fin})$

## Example starting with $\overline{Fin}$

$$B_0 = S; B_i = B_{i-1} \cup pre(B_{i-1})$$

$$B_0 = \{\{\}, \{1\}\}$$
$$B_1 = B_0 \cup pre(B_0)$$
$$= \{\{\}, \{1\}\} \cup pre(\{\{\}, \{1\}\})$$



# Backward reachability fixpoint algorithm in $G(A) = (V, E; In, \overline{Fin})$

## Example starting with $\overline{Fin}$

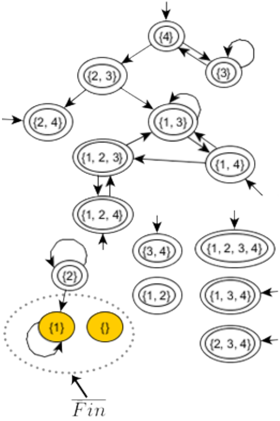
$B_0 = S; B_i = B_{i-1} \cup pre(B_{i-1})$

$B_0 = \{\{\}, \{1\}\}$

$B_1 = B_0 \cup pre(B_0)$

$= \{\{\}, \{1\}\} \cup pre(\{\{\}, \{1\}\})$

$= \{\{\}, \{1\}\} \cup \{\{1\}, \{2\}\} = \{\{\}, \{1\}, \{2\}\}$

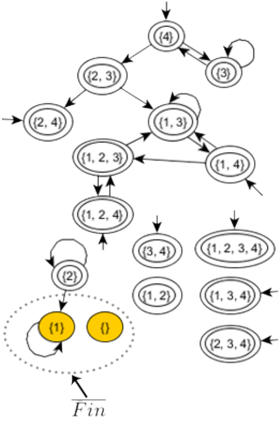


# Backward reachability fixpoint algorithm in $G(A) = (V, E; In, \overline{Fin})$

## Example starting with $\overline{Fin}$

$$B_0 = S; B_i = B_{i-1} \cup pre(B_{i-1})$$

$$B_0 = \{\{\}, \{1\}\}$$
$$B_1 = B_0 \cup pre(B_0)$$
$$= \{\{\}, \{1\}\} \cup pre(\{\{\}, \{1\}\})$$
$$= \{\{\}, \{1\}\} \cup \{\{1\}, \{2\}\} = \{\{\}, \{1\}, \{2\}\}$$
$$B_2 = B_1 \cup pre(B_1)$$

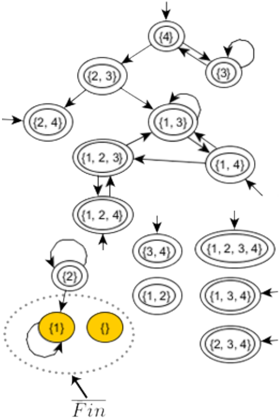


# Backward reachability fixpoint algorithm in $G(A) = (V, E; In, \overline{Fin})$

## Example starting with $\overline{Fin}$

$$B_0 = S; B_i = B_{i-1} \cup pre(B_{i-1})$$

$$B_0 = \{\{\}, \{1\}\}$$
$$B_1 = B_0 \cup pre(B_0)$$
$$= \{\{\}, \{1\}\} \cup pre(\{\{\}, \{1\}\})$$
$$= \{\{\}, \{1\}\} \cup \{\{1\}, \{2\}\} = \{\{\}, \{1\}, \{2\}\}$$
$$B_2 = B_1 \cup pre(B_1)$$
$$= \{\{\}, \{1\}, \{2\}\} \cup pre(\{\{\}, \{1\}, \{2\}\})$$

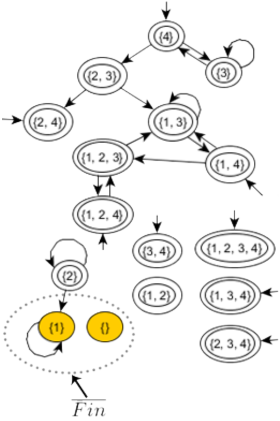


# Backward reachability fixpoint algorithm in $G(A) = (V, E; In, \overline{Fin})$

## Example starting with $\overline{Fin}$

$$B_0 = S; B_i = B_{i-1} \cup pre(B_{i-1})$$

$$
\begin{aligned}
B_0 &= \{\{\}, \{1\}\} \\
B_1 &= B_0 \cup pre(B_0) \\
&= \{\{\}, \{1\}\} \cup pre(\{\{\}, \{1\}\}) \\
&= \{\{\}, \{1\}\} \cup \{\{1\}, \{2\}\} = \{\{\}, \{1\}, \{2\}\} \\
B_2 &= B_1 \cup pre(B_1) \\
&= \{\{\}, \{1\}, \{2\}\} \cup pre(\{\{\}, \{1\}, \{2\}\}) \\
&= \{\{\}, \{1\}, \{2\}\} \cup \{\{1\}, \{2\}\}) = \{\{\}, \{1\}, \{2\}\}
\end{aligned}
$$



# Content

# Antichain Backward reachability algorithm

- Antichains can be used as representations for closed sets.
- Where can we introduce antichains in our algorithm?

# Antichain Backward reachability algorithm

- Antichains can be used as representations for closed sets.
- Where can we introduce antichains in our algorithm?

**Lemma 1**

Given $G = (V, E; In, \overline{Fin})$ then $pre(S)$ is downward closed for all downward closed sets $S \subseteq V$

# Antichain Backward reachability algorithm

- Antichains can be used as representations for closed sets.
- Where can we introduce antichains in our algorithm?

**Lemma 1**

Given $G = (V, E; In, \overline{Fin})$ then $pre(S)$ is downward closed for all downward closed sets $S \subseteq V$

**Lemma 2**

$\overline{Fin}$ is downward closed.

# Antichain Backward reachability algorithm
# Proof for Lemma 1

**Proof "*pre*($S$) is downward closed for all downward closed $S$."**

# Antichain Backward reachability algorithm
# Proof for Lemma 1

**Proof "$pre(S)$ is downward closed for all downward closed $S$."**

Let $S \subseteq V$ be downward closed. We need to show that

$$v_1 \in pre(S) \; and \; v_2 \subseteq v_1 \Rightarrow v_2 \in pre(S)$$

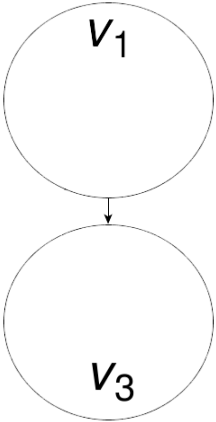
# Antichain Backward reachability algorithm
# Proof for Lemma 1

**Proof "$pre(S)$ is downward closed for all downward closed $S$."**

Let $S \subseteq V$ be downward closed. We need to show that

$$v_1 \in pre(S) \text{ and } v_2 \subseteq v_1 \Rightarrow v_2 \in pre(S)$$

Let $v_1 \in pre(S)$ this means there exists $v_3 \in S$ where $(v_1, v_3) \in E$ and a $\sigma \in \Sigma$ s.t. $\bigcup_{q \in v_1} \delta(q, \sigma) = v_3$



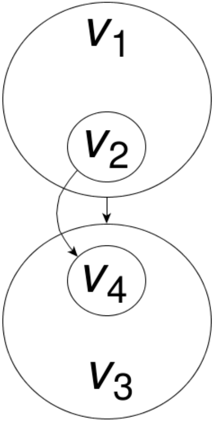
# Antichain Backward reachability algorithm
# Proof for Lemma 1

**Proof "*pre*($S$) is downward closed for all downward closed $S$."**

Let $S \subseteq V$ be downward closed. We need to show that

$$v_1 \in pre(S) \text{ and } v_2 \subseteq v_1 \Rightarrow v_2 \in pre(S)$$

Let $v_1 \in pre(S)$ this means there exists $v_3 \in S$ where $(v_1, v_3) \in E$ and a $\sigma \in \Sigma$ s.t. $\bigcup_{q \in v_1} \delta(q, \sigma) = v_3$
Looking at $v_2 \subseteq v_1$ we get $\bigcup_{q \in v_2} \delta(q, \sigma) = v_4 \subseteq v_3$



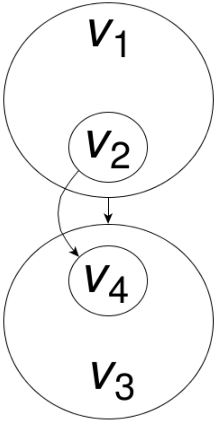
# Antichain Backward reachability algorithm
# Proof for Lemma 1

**Proof "*pre*($S$) is downward closed for all downward closed $S$."**

Let $S \subseteq V$ be downward closed. We need to show that

$$v_1 \in pre(S) \text{ and } v_2 \subseteq v_1 \Rightarrow v_2 \in pre(S)$$

Let $v_1 \in pre(S)$ this means there exists $v_3 \in S$ where $(v_1, v_3) \in E$ and a $\sigma \in \Sigma$ s.t. $\bigcup_{q \in v_1} \delta(q, \sigma) = v_3$
Looking at $v_2 \subseteq v_1$ we get $\bigcup_{q \in v_2} \delta(q, \sigma) = v_4 \subseteq v_3$
Hence $v_2 \in pre(S)$ □



# Antichain Backward reachability algorithm
# Proof for Lemma 2

**Recap: Definition of $\overline{Fin}$**

$\overline{Fin} := \{v \in 2^{Loc} | v \subseteq Loc \setminus Fin\}$

**Proof "$\overline{Fin}$ is downward closed."**

If $v_1 \in \overline{Fin}$ and $v_2 \subseteq v_1$ then $v_2 \subseteq v_1 \subseteq Loc \setminus Fin$ hence $v_2 \in \overline{Fin}$ $\square$

# Antichain Backward reachability algorithm

- We extend the fixpoint algorithm

$$B_0 = S; B_i = B_{i-1} \cup pre(B_{i-1})$$

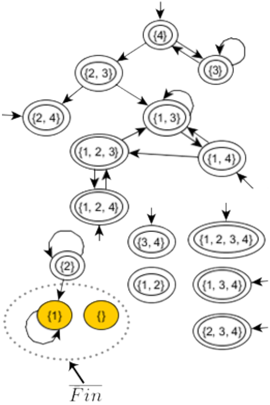
to the antichain fixpoint algorithm

$$\widetilde{B}_0 = Max(\subseteq, S); \widetilde{B}_i = Max(\subseteq, \widetilde{B}_{i-1} \cup pre(Down(\subseteq, \widetilde{B}_{i-1})))$$

# Antichain Backward reachability algorithm. Starting with $S = \overline{Fin} = \{\{\},\{1\}\}$;

## Example starting with $\overline{Fin}$

$\widetilde{B}_0 = Max(\subseteq, S)$;
$\widetilde{B}_i = Max(\subseteq, \widetilde{B}_{i-1} \cup pre(Down(\subseteq, \widetilde{B}_{i-1})))$

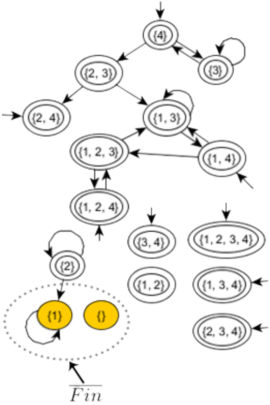


## Antichain Backward reachability algorithm. Starting with $S = \overline{Fin} = \{\{\},\{1\}\}$;

### Example starting with $\overline{Fin}$

$\widetilde{B}_0 = Max(\subseteq, S)$;
$\widetilde{B}_i = Max(\subseteq, \widetilde{B}_{i-1} \cup pre(Down(\subseteq, \widetilde{B}_{i-1})))$

$\widetilde{B}_0 = Max(\subseteq, \{\{\},\{1\}\}) = \{\{1\}\}$



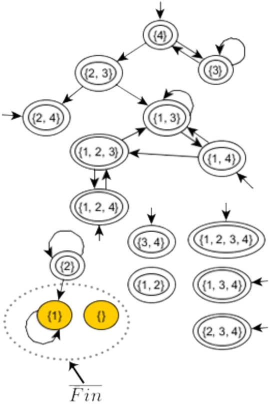
# Antichain Backward reachability algorithm. Starting with $S = \overline{Fin} = \{\{\},\{1\}\}$;

## Example starting with $\overline{Fin}$

$$\widetilde{B}_0 = Max(\subseteq, S);$$
$$\widetilde{B}_i = Max(\subseteq, \widetilde{B}_{i-1} \cup pre(Down(\subseteq, \widetilde{B}_{i-1})))$$

$$\widetilde{B}_0 = Max(\subseteq, \{\{\},\{1\}\}) = \{\{1\}\}$$

$$\widetilde{B}_1 = Max(\subseteq, \{1\} \cup pre(Down(\subseteq, \{1\})))$$



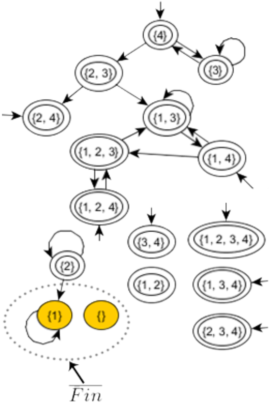
# Antichain Backward reachability algorithm. Starting with $S = \overline{Fin} = \{\{\},\{1\}\}$;

## Example starting with $\overline{Fin}$

$$\widetilde{B}_0 = Max(\subseteq, S);$$
$$\widetilde{B}_i = Max(\subseteq, \widetilde{B}_{i-1} \cup pre(Down(\subseteq, \widetilde{B}_{i-1})))$$

$$\widetilde{B}_0 = Max(\subseteq, \{\{\},\{1\}\}) = \{\{1\}\}$$

$$\widetilde{B}_1 = Max(\subseteq, \{1\} \cup pre(Down(\subseteq, \{1\})))$$
$$= Max(\subseteq, \{1\} \cup pre(\{\{\},\{1\}\}))$$



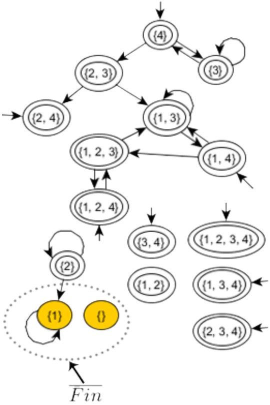
# Antichain Backward reachability algorithm. Starting with $S = \overline{Fin} = \{\{\}, \{1\}\}$;

## Example starting with $\overline{Fin}$

$\widetilde{B}_0 = Max(\subseteq, S)$;
$\widetilde{B}_i = Max(\subseteq, \widetilde{B}_{i-1} \cup pre(Down(\subseteq, \widetilde{B}_{i-1})))$

$$\widetilde{B}_0 = Max(\subseteq, \{\{\}, \{1\}\}) = \{\{1\}\}$$

$$\begin{aligned}\widetilde{B}_1 &= Max(\subseteq, \{1\} \cup pre(Down(\subseteq, \{1\}))) \\ &= Max(\subseteq, \{1\} \cup pre(\{\{\}, \{1\}\})) \\ &= Max(\subseteq, \{1\} \cup \{\{1\}, \{2\}\}) = \{\{1\}, \{2\}\}\end{aligned}$$



# Antichain Backward reachability algorithm. Starting with $S = \overline{Fin} = \{\{\},\{1\}\}$;

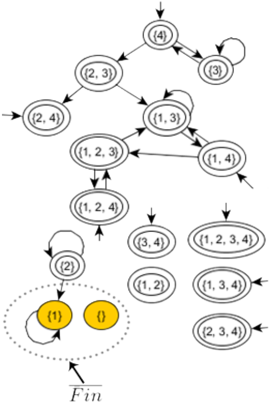
## Example starting with $\overline{Fin}$

$\widetilde{B}_0 = Max(\subseteq, S)$;
$\widetilde{B}_i = Max(\subseteq, \widetilde{B}_{i-1} \cup pre(Down(\subseteq, \widetilde{B}_{i-1})))$

$\widetilde{B}_0 = Max(\subseteq, \{\{\},\{1\}\}) = \{\{1\}\}$

$$\begin{aligned}\widetilde{B}_1 &= Max(\subseteq, \{1\} \cup pre(Down(\subseteq, \{1\})))\\ &= Max(\subseteq, \{1\} \cup pre(\{\{\},\{1\}\}))\\ &= Max(\subseteq, \{1\} \cup \{\{1\},\{2\}\}) = \{\{1\},\{2\}\}\end{aligned}$$

$\widetilde{B}_2 = Max(\subseteq, \{\{1\},\{2\}\} \cup pre(Down(\subseteq, \{\{1\},\{2\}\})))$



# Antichain Backward reachability algorithm. Starting with $S = \overline{Fin} = \{\{\},\{1\}\}$;

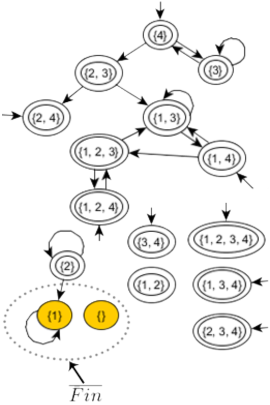
## Example starting with $\overline{Fin}$

$$\widetilde{B}_0 = Max(\subseteq, S);$$
$$\widetilde{B}_i = Max(\subseteq, \widetilde{B}_{i-1} \cup pre(Down(\subseteq, \widetilde{B}_{i-1})))$$

$$\widetilde{B}_0 = Max(\subseteq, \{\{\},\{1\}\}) = \{\{1\}\}$$

$$\widetilde{B}_1 = Max(\subseteq, \{1\} \cup pre(Down(\subseteq, \{1\})))$$
$$= Max(\subseteq, \{1\} \cup pre(\{\{\},\{1\}\}))$$
$$= Max(\subseteq, \{1\} \cup \{\{1\},\{2\}\}) = \{\{1\},\{2\}\}$$

$$\widetilde{B}_2 = Max(\subseteq, \{\{1\},\{2\}\} \cup pre(Down(\subseteq, \{\{1\},\{2\}\})))$$
$$= Max(\subseteq, \{\{1\},\{2\}\} \cup pre(\{\{\},\{1\},\{2\}\}))$$



# Antichain Backward reachability algorithm. Starting with $S = \overline{Fin} = \{\{\},\{1\}\}$;

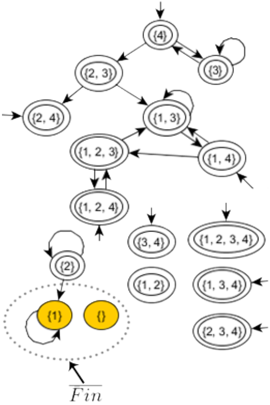
## Example starting with $\overline{Fin}$

$\widetilde{B}_0 = Max(\subseteq, S)$;
$\widetilde{B}_i = Max(\subseteq, \widetilde{B}_{i-1} \cup pre(Down(\subseteq, \widetilde{B}_{i-1})))$

$$\widetilde{B}_0 = Max(\subseteq, \{\{\},\{1\}\}) = \{\{1\}\}$$

$$\begin{aligned}\widetilde{B}_1 &= Max(\subseteq, \{1\} \cup pre(Down(\subseteq, \{1\}))) \\ &= Max(\subseteq, \{1\} \cup pre(\{\{\},\{1\}\})) \\ &= Max(\subseteq, \{1\} \cup \{\{1\},\{2\}\}) = \{\{1\},\{2\}\}\end{aligned}$$

$$\begin{aligned}\widetilde{B}_2 &= Max(\subseteq, \{\{1\},\{2\}\} \cup pre(Down(\subseteq, \{\{1\},\{2\}\}))) \\ &= Max(\subseteq, \{\{1\},\{2\}\} \cup pre(\{\{\},\{1\},\{2\}\})) \\ &= Max(\subseteq, \{\{1\},\{2\}\} \cup \{\{1\},\{2\}\}) = \{\{1\},\{2\}\}\end{aligned}$$



# Content

# Conclusion

- Antichains can be used as representations of closed sets.

# Conclusion

- Antichains can be used as representations of closed sets.
- In the powerset construction $\overline{Fin}$ is a downward closed set.

# Conclusion

- Antichains can be used as representations of closed sets.
- In the powerset construction $\overline{Fin}$ is a downward closed set.
- On the powerset automaton $pre(S)$ for closed $S$ Is downward closed. Thus we can use antichains in the classic backward reachability algorithm.

# Conclusion

- Antichains can be used as representations of closed sets.
- In the powerset construction $\overline{Fin}$ is a downward closed set.
- On the powerset automaton $pre(S)$ for closed $S$ Is downward closed. Thus we can use antichains in the classic backward reachability algorithm.
- To be efficient, further improvements are possible and will be shown in antichain talk II.

# References I

Doyen, Laurent and Raskin, Jean-François
Antichain algorithms for finite automata
**International Conference on Tools and Algorithms for the Construction and Analysis of Systems**, 2–22, 2010.

De Wulf, Martin and Doyen, Laurent and Henzinger, Thomas A and Raskin, J-F.
Antichains: A new algorithm for checking universality of finite automata.
**International Conference on Computer Aided Verification**, 17–30, 2006.