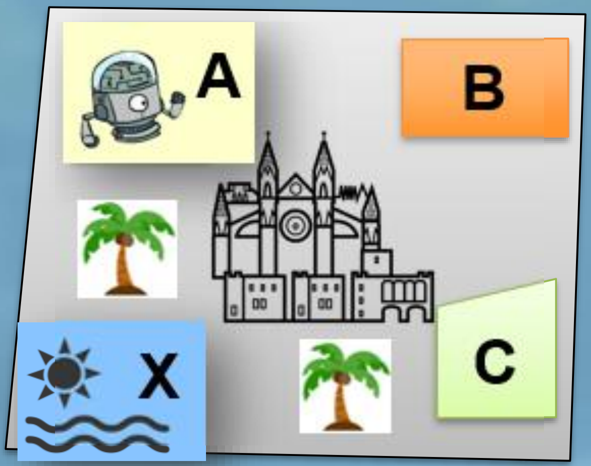
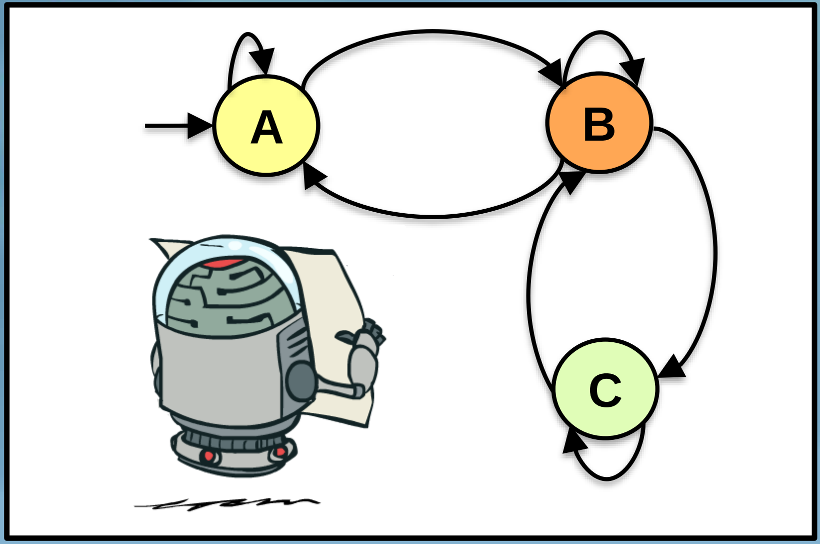
Graz University of Technology
Institute for Applied Information Processing and Communications

Automata and LTL Model Checking

Bettina Könighofer





Model Checking SS23

May 25th 2023

Model Checking of LTL

given an LTL property φ and a Kripke structure M
check whether $M \models \varphi$

1. Construct $\neg\varphi$
2. Construct a **Büchi** automaton $\mathcal{S}_{\neg\varphi}$
3. **Translate M to an automaton $\mathcal{A}$.**
4. Construct the automaton $\mathcal{B}$ with $\mathcal{L}(\mathcal{B}) = \mathcal{L}(\mathcal{A}) \cap \mathcal{L}(\mathcal{S}_{\neg\varphi})$
5. If $\mathcal{L}(\mathcal{B}) = \emptyset \Rightarrow \mathcal{A}$ satisfies φ
6. Otherwise, a word $v \cdot w^{\omega} \in \mathcal{L}(\mathcal{B})$ is a counterexample
   - a computation in M that does not satisfy φ

Outline

- Finite automata on finite words
- Automata on infinite words (Büchi automata)
- Deterministic vs non-deterministic Büchi automata
- Intersection of Büchi automata
- Checking emptiness of Büchi automata
- Generalized Büchi automata
- **Model checking using automata**
- Translation of LTL to Büchi automata

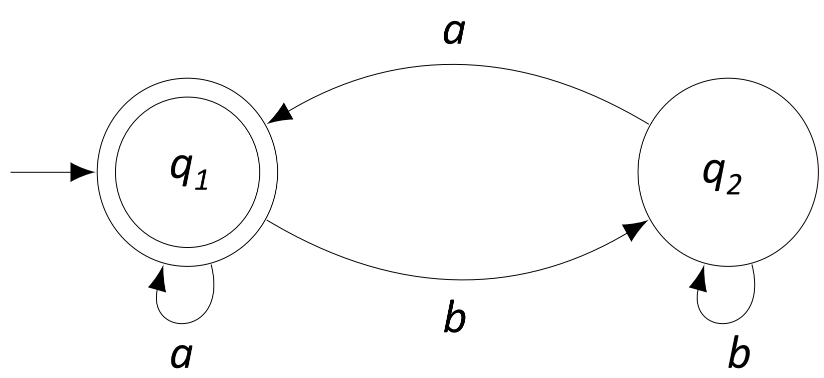
Outline

- Finite automata on finite words
- Automata on infinite words (Büchi automata)
- Deterministic vs non-deterministic Büchi automata
- Intersection of Büchi automata
- Checking emptiness of Büchi automata
- Generalized Büchi automata
- Model checking using automata

Automata on **Infinite** Words (Büchi)

$$\mathcal{B} = (\Sigma, Q, \Delta, Q^0, F)$$

- An infinite run ρ is accepting ⇔ it visits an accepting state an infinite number of times.



Outline

- Finite automata on finite words
- Automata on infinite words (Büchi automata)
- Deterministic vs non-deterministic Büchi automata
- Intersection of Büchi automata
- Checking emptiness of Büchi automata
- Generalized Büchi automata
- Model checking using automata

Det. and Non-det. Büchi Automata

- Deterministic Büchi automata are **strictly less expressive** than nondeterministic ones.
  - Not every nondeterministic Büchi automaton has an equivalent deterministic Büchi one.
- Deterministic Büchi automata are **not** closed under complementation.
- Nondeterministic Büchi automata are closed under complementation.
  - The construction is very complicated. (Safra Construction)

Outline

- Finite automata on finite words
- Automata on infinite words (Büchi automata)
- Deterministic vs non-deterministic Büchi automata
- **Intersection of Büchi automata**
- Checking emptiness of Büchi automata
- Generalized Büchi automata
- Model checking using automata

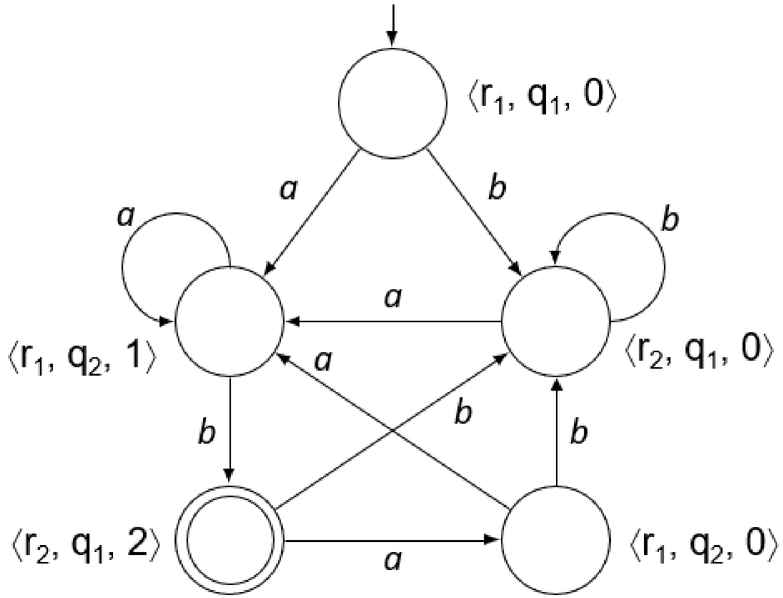
Intersection of Büchi Automata

- Given $\mathcal{B}_1 = (\Sigma, Q_1, \Delta_1, Q_1^0, F_1)$ and $\mathcal{B}_2 = (\Sigma, Q_2, \Delta_2, Q_2^0, F_2)$
- $\mathcal{B} = (\Sigma, Q, \Delta, Q^0, F)$ s.t. $\mathcal{L}(\mathcal{B}) = \mathcal{L}(\mathcal{B}_1) \cap \mathcal{L}(\mathcal{B}_2)$ is defined as follows:
  - $Q = Q_1 \times Q_2 \times \{0, 1, 2\}$
  - $Q^0 = Q_1^0 \times Q_2^0 \times \{0\}$
  - $F = Q_1 \times Q_2 \times \{2\}$

$((q_1,q_2,x), a, (q'_1,q'_2,x')) \in \Delta \Leftrightarrow$

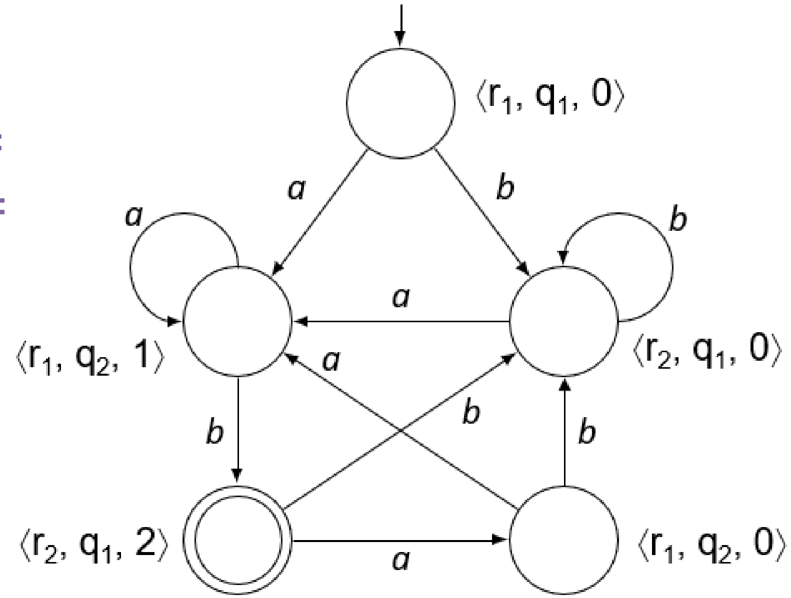
(1) $(q_1,a,q'_1) \in \Delta_1$ and $(q_2,a,q'_2) \in \Delta_2$

(2) If $x=0$ and $q'_1 \in F_1$ then $x'=1$
If $x=1$ and $q'_2 \in F_2$ then $x'=2$
If $x=2$ then $x'=0$
Else, $x'=x$



Intersection of Büchi Automata

- Given $\mathcal{B}_1 = (\Sigma, Q_1, \Delta_1, Q_1^0, F_1)$ and $\mathcal{B}_2 = (\Sigma, Q_2, \Delta_2, Q_2^0, F_2)$
- $\mathcal{B} = (\Sigma, Q, \Delta, Q^0, F)$ s.t. $\mathcal{L}(\mathcal{B}) = \mathcal{L}(\mathcal{B}_1) \cap \mathcal{L}(\mathcal{B}_2)$ is defined as follows:
  - $Q = Q_1 \times Q_2 \times \{0, 1\}$
  - $Q^0 = Q_1^0 \times Q_2^0 \times \{0\}$
  - For Δ
    - (2) If $x=0$ and $q_1 \in F_1$ then $x'=$
      If $x=1$ and $q_2 \in F_2$ then $x'=$
      Else, $x'=x$
  - For F
    - $F = F_1 \times Q_2 \times \{0\}$



SCOS Secure & Correct Systems

Intersection of Büchi Automata

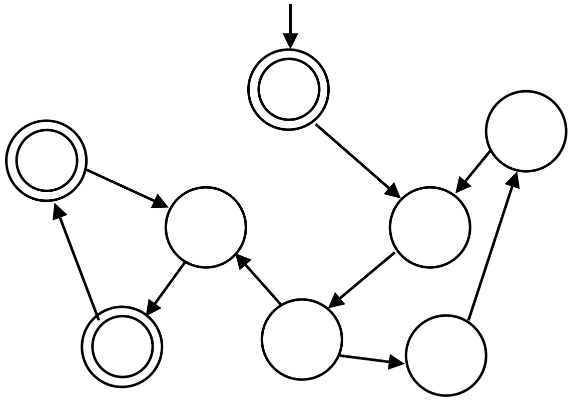
- Question
  - How do we define the transition relation for $\mathcal{B}$, if x is over {0,1} only?
- Answer
  - For $\mathbf{\Delta}$
    - (2) If x=0 and $q_1 \in \mathbf{F}_1$ then x'=1
      If x=1 and $q_2 \in \mathbf{F}_2$ then x'=0
      Else, x'=x
  - For $\mathbf{F}$
    - $\mathbf{F} = \mathbf{F}_1 \times \mathbf{Q}_2 \times \{0\}$

Outline

- Finite automata on finite words
- Automata on infinite words (Büchi automata)
- Deterministic vs non-deterministic Büchi automata
- Intersection of Büchi automata
- Checking emptiness of Büchi automata
- Generalized Büchi automata
- Automata and Kripke Structures
- Model checking using automata
- Translation of LTL to Büchi automata

Checking for emptiness of $\mathcal{L}(\mathcal{B})$

- How to check for **L(A) = ∅?**
- Empty if there is no reachable accepting state on a cycle.



Outline

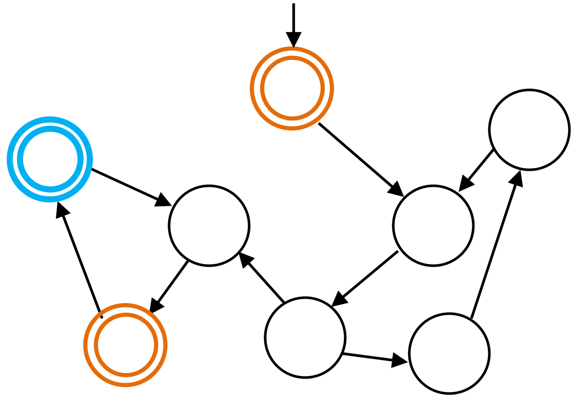
- Finite automata on finite words
- Automata on infinite words (Büchi automata)
- Deterministic vs non-deterministic Büchi automata
- Intersection of Büchi automata
- Checking emptiness of Büchi automata
- Generalized Büchi automata
- Automata and Kripke Structures
- Model checking using automata
- Translation of LTL to Büchi automata

Generalized Büchi automata

- Have several sets of accepting states

- $\mathcal{B} = (\Sigma, \mathbf{Q}, \Delta, \mathbf{Q}^0, \mathbf{F})$ is a generalized Büchi automaton:
  - $\mathbf{F} = \{P_1, \ldots, P_k\}$, where for every $1 \leq i \leq k$, $P_i \subseteq Q$

- A run ρ of $\mathcal{B}$ is accepting if for each $P_i \in \mathbf{F}$, $inf(\rho) \cap P_i \neq \emptyset$



Translation from Generalized Büchi to Büchi

- Given $\mathcal{B} = (\Sigma, Q_1, \Delta_1, Q_1^0, F_1)$ with $F = \{P_1, \ldots, P_k\}$

- How can we construct a Büchi automaton $\mathcal{B}'$ that accepts the same language?

Translation from Generalized Büchi to Büchi

- $\mathcal{B} = (\Sigma, Q_1, \Delta_1, Q_1^0, F_1)$ with $F = \{P_1, \ldots, P_k\}$
- $\mathcal{B}' = (\Sigma, Q \times \{0,1,\ldots,k\}, \Delta', Q^0 x0, Q \times k)$ with:
- The transition relation Δ':
  $((q,x),a,(q',y)) \in \Delta'$ when $(q,a,q') \in \Delta$ and x and y are as follows:
  - If $q' \in P_i$ and $x=i$, then $y=i+1$ for $i<k$
  - If $x=k$, then $y=0$.
  - Otherwise, $x = y$.

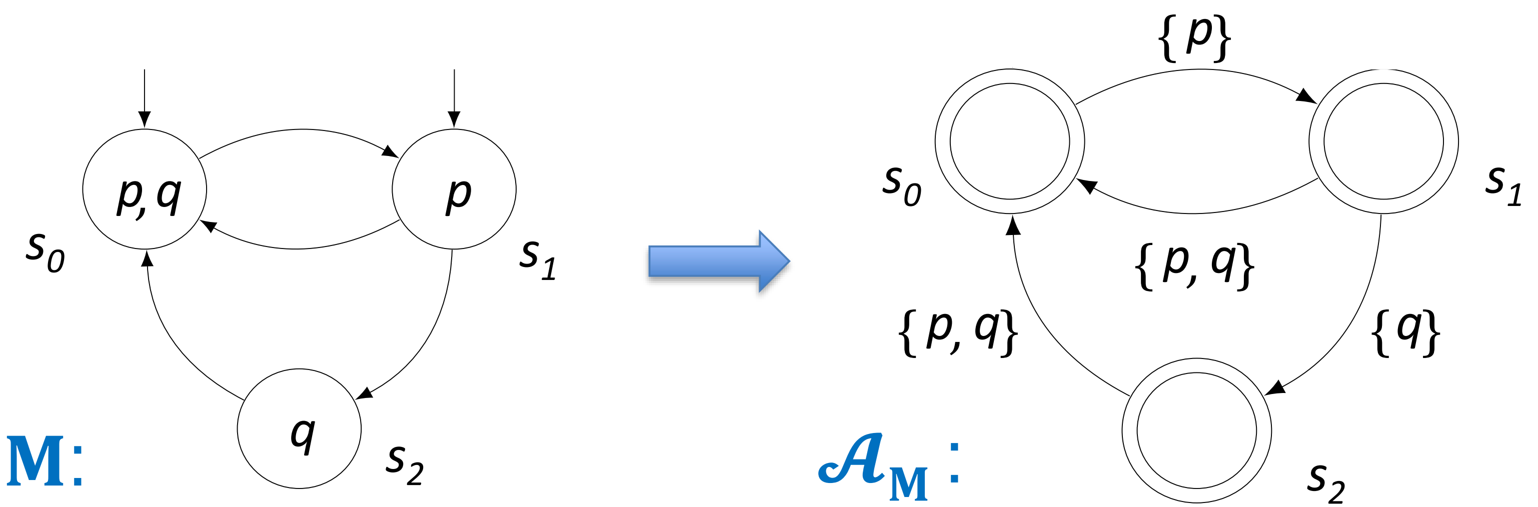
Size of $\mathcal{B}'$ = (size of $\mathcal{B}$) $\times$ (k+1)

Outline

- Finite automata on finite words
- Automata on infinite words (Büchi automata)
- Deterministic vs non-deterministic Büchi automata
- Intersection of Büchi automata
- Checking emptiness of Büchi automata
- Generalized Büchi automata
- Automata and Kripke Structures
- Model checking using automata
- Translation of LTL to Büchi automata

Kripke Structure $\mathbf{M}$ to Büchi Automaton $\mathcal{A}_{\mathbf{M}}$

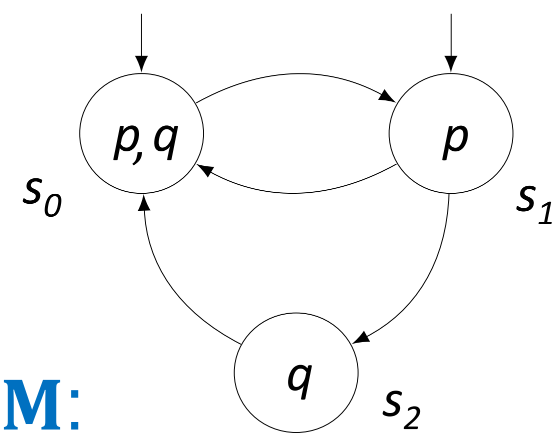
- Move labels to incoming transitions
  - Push labels backwards
- All states are accepting
- ToDo: What about initial states?



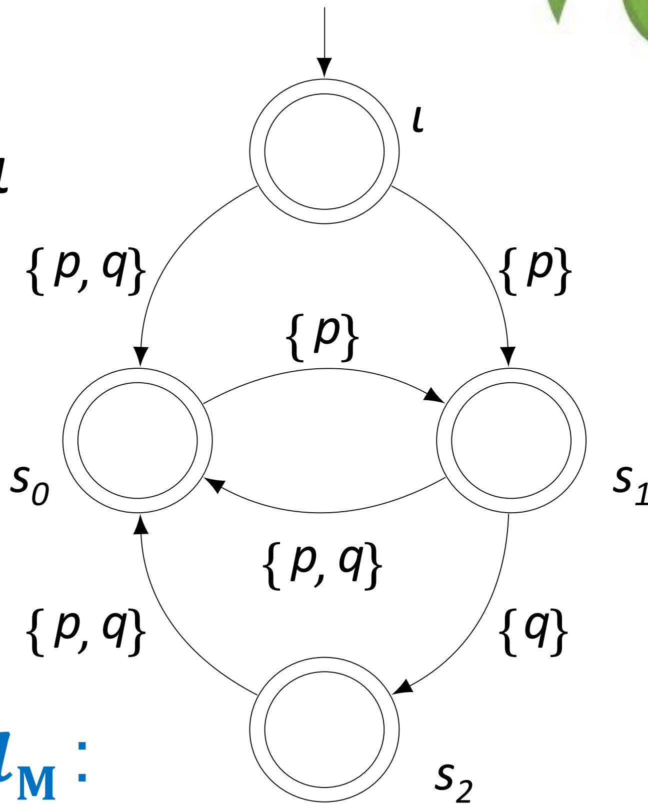
Kripke Structure **M** to Büchi Automaton $\mathcal{A}_{\mathbf{M}}$

- Move labels to incoming transitions
- All states are accepting
- Introduce new initial state ι

M:



$\mathcal{A}_{\mathbf{M}}$:



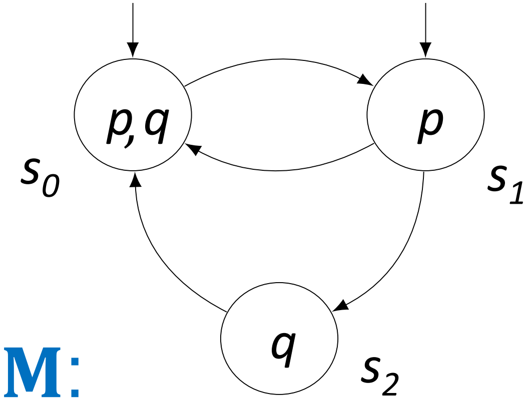
Automata and Kripke Structures

$\mathbf{M} = (\mathbf{S}, \mathbf{S}_0, \mathbf{R}, AP, \mathbf{L}) \Rightarrow \mathcal{A}_{\mathbf{M}} = (\Sigma, \mathbf{S} \cup \{\iota\}, \Delta, \{\iota\}, \mathbf{S} \cup \{\iota\})$,

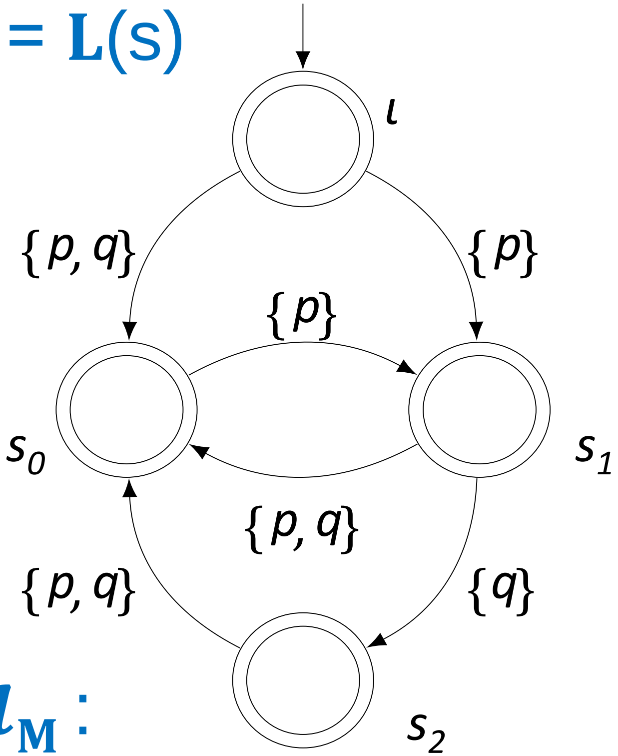
where $\Sigma = P(AP)$.

- $(s,\alpha,s') \in \Delta$ for $s,s' \in \mathbf{S} \Leftrightarrow (s,s') \in R$ and $\alpha = \mathbf{L}(s')$
- $(\iota,\alpha,s) \in \Delta \Leftrightarrow s \in \mathbf{S}_0$ and $\alpha = \mathbf{L}(s)$

M:



$\mathcal{A}_{\mathbf{M}}$:

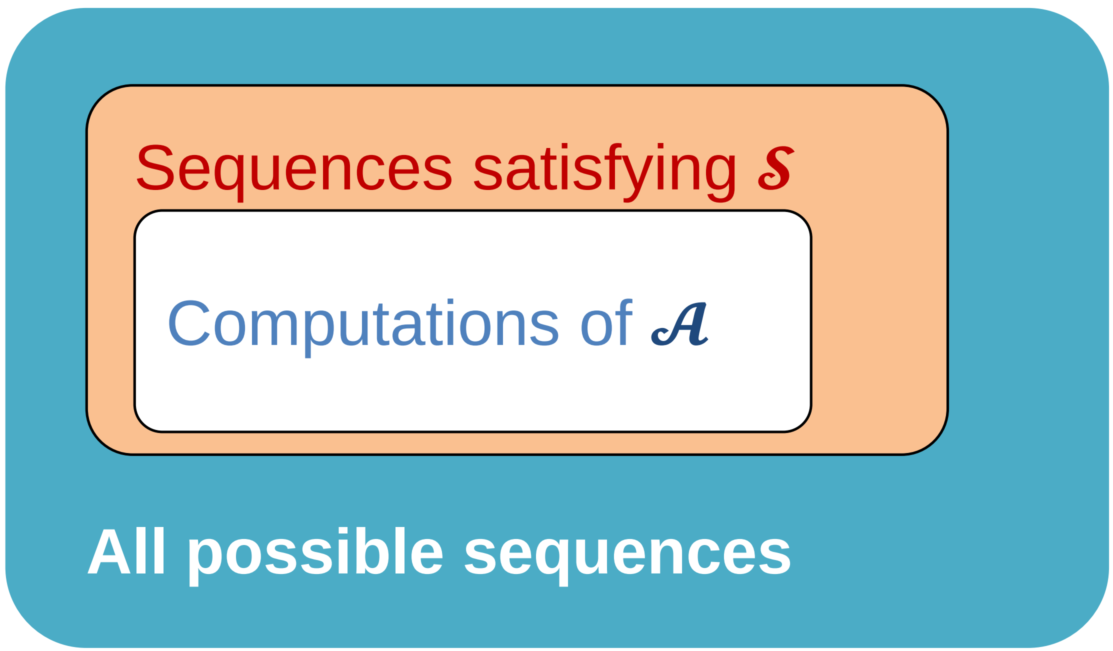


Outline

- Finite automata on finite words
- Automata on infinite words (Büchi automata)
- Deterministic vs non-deterministic Büchi automata
- Intersection of Büchi automata
- Checking emptiness of Büchi automata
- Generalized Büchi automata
- Automata and Kripke Structures
- **Model checking using automata**
- Translation of LTL to Büchi automata

Model Checking when System $\mathcal{A}$ and Spec $\mathcal{S}$ are given as Büchi automata

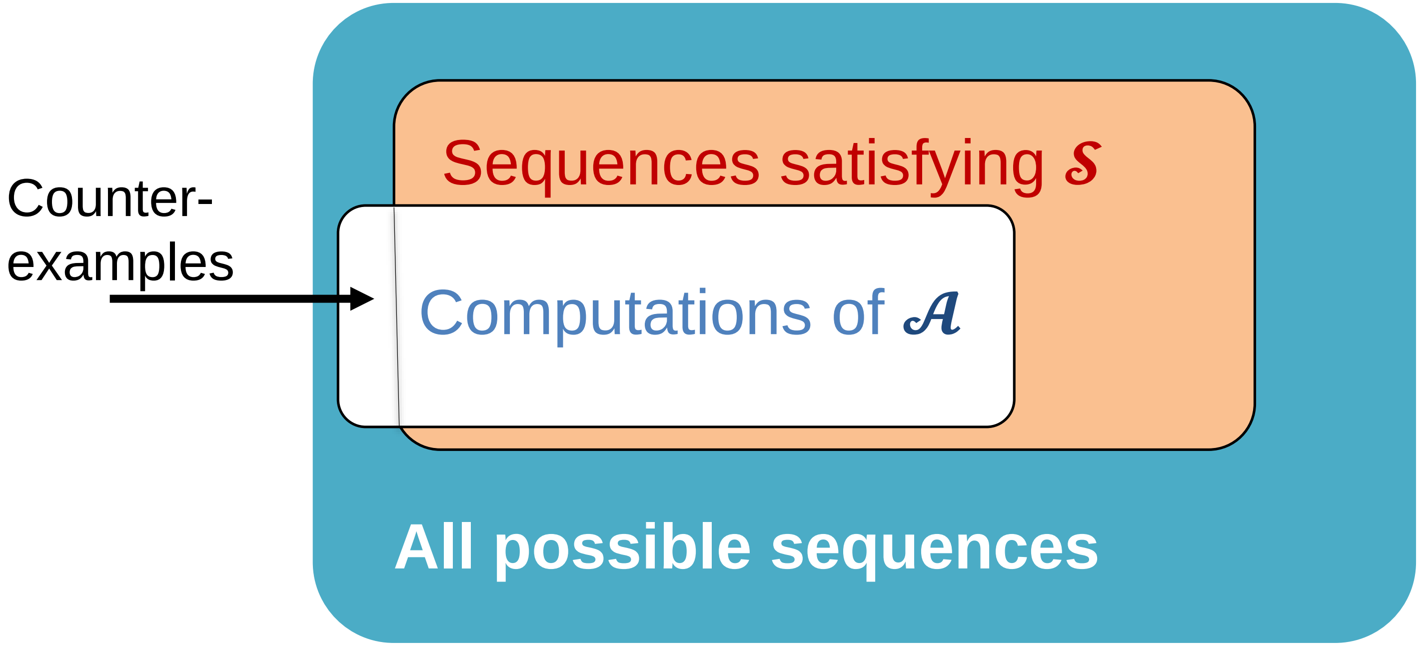
- $\mathcal{A}$ **satisfies** $\mathcal{S}$ if $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{S})$
  - Is any behavior of $\mathcal{A}$ allowed by $\mathcal{S}$?



SCOS
Secure & Correct Systems

Model Checking when System $\mathcal{A}$ and Spec $\mathcal{S}$ are given as Büchi automata

- **$\mathcal{A}$ does not satisfy $\mathcal{S}$** if $\mathcal{L}(\mathcal{A}) \not\subseteq \mathcal{L}(\mathcal{S})$



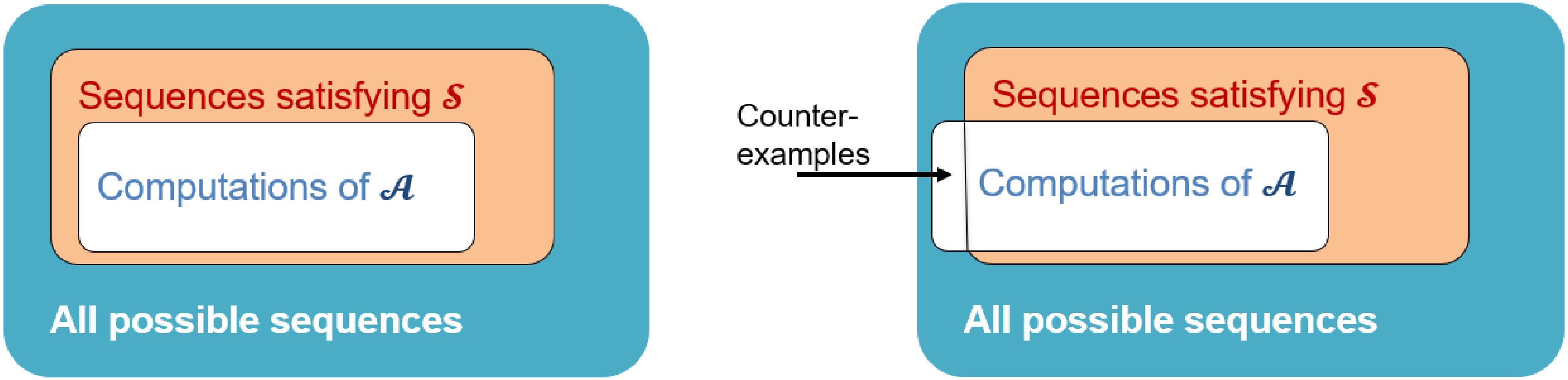
Model Checking when System $\mathcal{A}$ and Spec $\mathcal{S}$ are given as Büchi automata

- Check whether $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{S})$
- Equivalent:

$$\mathcal{L}(\mathcal{A}) \nsubseteq \mathcal{L}(\mathcal{S}) \equiv \boxed{\mathcal{L}(\mathcal{A}) \cap \mathcal{L}(\overline{\mathcal{S}}) \neq \emptyset}$$

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

$\mathcal{L}(\mathcal{A}) \nsubseteq \mathcal{L}(\mathcal{S}) \equiv \mathcal{L}(\mathcal{A}) \cap \mathcal{L}(\overline{\mathcal{S}}) \neq \emptyset$

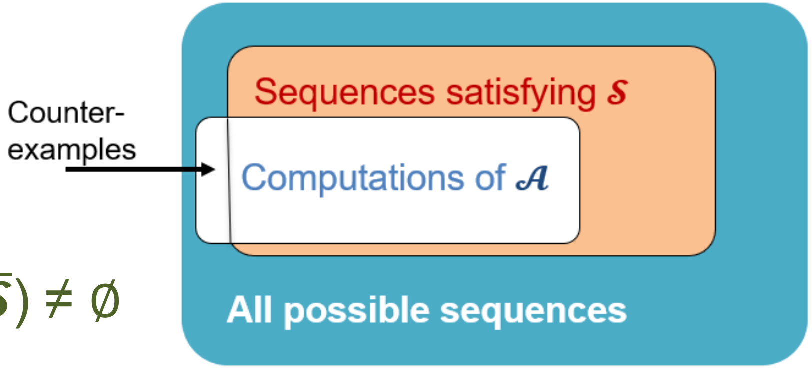


SCOS Secure & Correct Systems

Model Checking – suggested algorithm

when system $\mathcal{A}$ and spec $\mathcal{S}$ are given as Büchi automata

1. Complement $\mathcal{S}$. The resulting Büchi automaton is $\overline{\mathcal{S}}$
2. Construct the automaton $\mathcal{B}$ with $\mathcal{L}(\mathcal{B}) = \mathcal{L}(\mathcal{A}) \cap \mathcal{L}(\overline{\mathcal{S}})$
3. If $\mathcal{L}(\mathcal{B}) = \emptyset \Rightarrow \mathcal{A}$ satisfies $\mathcal{S}$
4. Otherwise, a word $v \cdot w^{\omega} \in \mathcal{L}(\mathcal{B})$ is a counterexample
   - a computation in $\mathcal{A}$ that does not satisfy $\mathcal{S}$



$$\mathcal{L}(\mathcal{A}) \not\subseteq \mathcal{L}(\mathcal{S}) \equiv \mathcal{L}(\mathcal{A}) \cap \mathcal{L}(\overline{\mathcal{S}}) \neq \emptyset$$

Model Checking – suggested algorithm

when system $\mathcal{A}$ and spec $\mathcal{S}$ are given as Büchi automata

very hard!

1. Complement $\mathcal{S}$. The resulting Büchi automaton is $\overline{\mathcal{S}}$
2. ✓ Construct the automaton $\mathcal{B}$ with $\mathcal{L}(\mathcal{B}) = \mathcal{L}(\mathcal{A}) \cap \mathcal{L}(\overline{\mathcal{S}})$
3. ✓ If $\mathcal{L}(\mathcal{B}) = \emptyset \Rightarrow \mathcal{A}$ satisfies $\mathcal{S}$
4. ✓ Otherwise, a word $v \cdot w^{\omega} \in \mathcal{L}(\mathcal{B})$ is a counterexample
   - a computation in $\mathcal{A}$ that does not satisfy $\mathcal{S}$

Model Checking – suggested algorithm

when system $\mathcal{A}$ and spec $\mathcal{S}$ are given as Büchi automata

very hard!

1. Complement $\mathcal{S}$. The resulting Büchi automaton is $\overline{\mathcal{S}}$
2. Construct the automaton $\mathcal{B}$ with $\mathcal{L}(\mathcal{B}) = \mathcal{L}(\mathcal{A}) \cap \mathcal{L}(\overline{\mathcal{S}})$
3. If $\mathcal{L}(\mathcal{B}) = \emptyset \Rightarrow \mathcal{A}$ satisfies $\mathcal{S}$
4. Otherwise, a word $v \cdot w^{\omega} \in \mathcal{L}(\mathcal{B})$ is a counterexample
   - a computation in $\mathcal{A}$ that does not satisfy $\mathcal{S}$

How can we avoid building the **complement** of $\mathcal{S}$?

Model Checking of LTL

given **an LTL property φ** and a Kripke structure M
check whether $M \models \varphi$

Model Checking of LTL

given **an LTL property** φ and a Kripke structure M
check whether M $\models \varphi$

1. Construct $\neg\varphi$
2. Construct a Büchi automaton $\mathcal{S}_{\neg\varphi}$
3. Translate M to an automaton $\mathcal{A}$.
4. Construct the automaton $\mathcal{B}$ with $\mathcal{L}(\mathcal{B}) = \mathcal{L}(\mathcal{A}) \cap \mathcal{L}(\mathcal{S}_{\neg\varphi})$
5. If $\mathcal{L}(\mathcal{B}) = \emptyset \Rightarrow \mathcal{A}$ satisfies φ
6. Otherwise, a word $v \cdot w^{\omega} \in \mathcal{L}(\mathcal{B})$ is a counterexample
   - a computation in M that does not satisfy φ

Model Checking of LTL

given **an LTL property φ** and a Kripke structure M
check whether $M \models \varphi$

next topic

1. Construct $\neg\varphi$
2. Construct a Büchi automaton $\mathcal{S}_{\neg\varphi}$
3. Translate M to an automaton $\mathcal{A}$.
4. Construct the automaton $\mathcal{B}$ with $\mathcal{L}(\mathcal{B}) = \mathcal{L}(\mathcal{A}) \cap \mathcal{L}(\mathcal{S}_{\neg\varphi})$
5. If $\mathcal{L}(\mathcal{B}) = \emptyset \Rightarrow \mathcal{A}$ satisfies φ
6. Otherwise, a word $v \cdot w^{\omega} \in \mathcal{L}(\mathcal{B})$ is a counterexample
   - a computation in M that does not satisfy φ

Outline

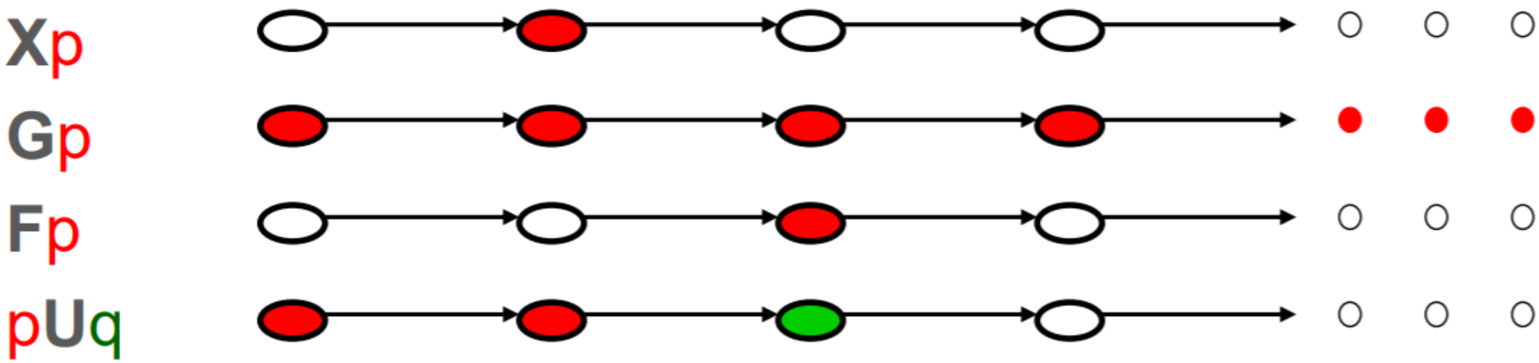
- Finite automata on finite words
- Automata on infinite words (Büchi automata)
- Deterministic vs non-deterministic Büchi automata
- Intersection of Büchi automata
- Checking emptiness of Büchi automata
- Generalized Büchi automata
- Automata and Kripke Structures
- Model checking using automata
- **Translation of LTL to Büchi automata**

Translation of LTL to Büchi automata

1. Translate φ into **generalized Büchi Automaton**
   - $\mathcal{A}_\varphi$ accepts exactly all the traces that satisfy φ

2. Translate **generalized Büchi** to **Büchi** automaton ✓

Recall LTL Semantics



Rewriting

- Algorithm only handles
  - $\neg, \wedge, \vee, \boldsymbol{X}, \boldsymbol{U}$
  - Use rewriting Rules
    - $\neg G\varphi = F\neg\varphi$
    - $F\varphi = true\ U\varphi$
    - $G\varphi = \neg F\neg\varphi$

From LTL formula φ to GBA $\mathcal{A}_\varphi$

- Step 1: Define the state space $\mathcal{A}_\varphi$ based on φ

The set of all **good sets** of **cl(φ)** defines the state space of $\mathcal{A}_\varphi$

- cl(φ) are subformulas of φ and their negation
- Formally:
  - $\varphi \in cl(\varphi)$.
  - **If** $\boldsymbol{\varphi_1 \in cl(\varphi), then \neg\varphi_1 \in cl(\varphi)}$.
  - **If** $\boldsymbol{\neg\varphi_1 \in cl(\varphi), then\ \varphi_1 \in cl(\varphi)}$.
  - If $\varphi_1 \vee \varphi_2 \in cl(\varphi), then\ \varphi_1 \in cl(\varphi)\ and\ \varphi_2 \in cl(\varphi)$.
  - If $X\ \varphi_1 \in cl(\varphi), then\ \varphi_1 \in cl(\varphi)$.
  - If $\varphi_1\ U\ \varphi_2 \in cl(\varphi), then\ \varphi_1 \in cl(\varphi)\ and\ \varphi_2 \in cl(\varphi)$.

From LTL formula φ to GBA $\mathcal{A}_\varphi$

- Step 1: Define the state space $\mathcal{A}_\varphi$ based on φ

The set of all **good sets** of **cl(φ)** defines the state space of $\mathcal{A}_\varphi$

- cl(φ) are subformulas of φ and their negation
- Example cl(φ) for $\varphi \coloneqq (\neg p \; U \; ((Xq) \vee r))$

$$\mathrm{cl}(\varphi) = \{\varphi, \neg\varphi, \neg p, p, q, \neg q, r, \neg r\ , (Xq), \neg(Xq), ((Xq) \vee r), \neg((Xq) \vee r), \}$$

From LTL formula φ to GBA $\mathcal{A}_\varphi$

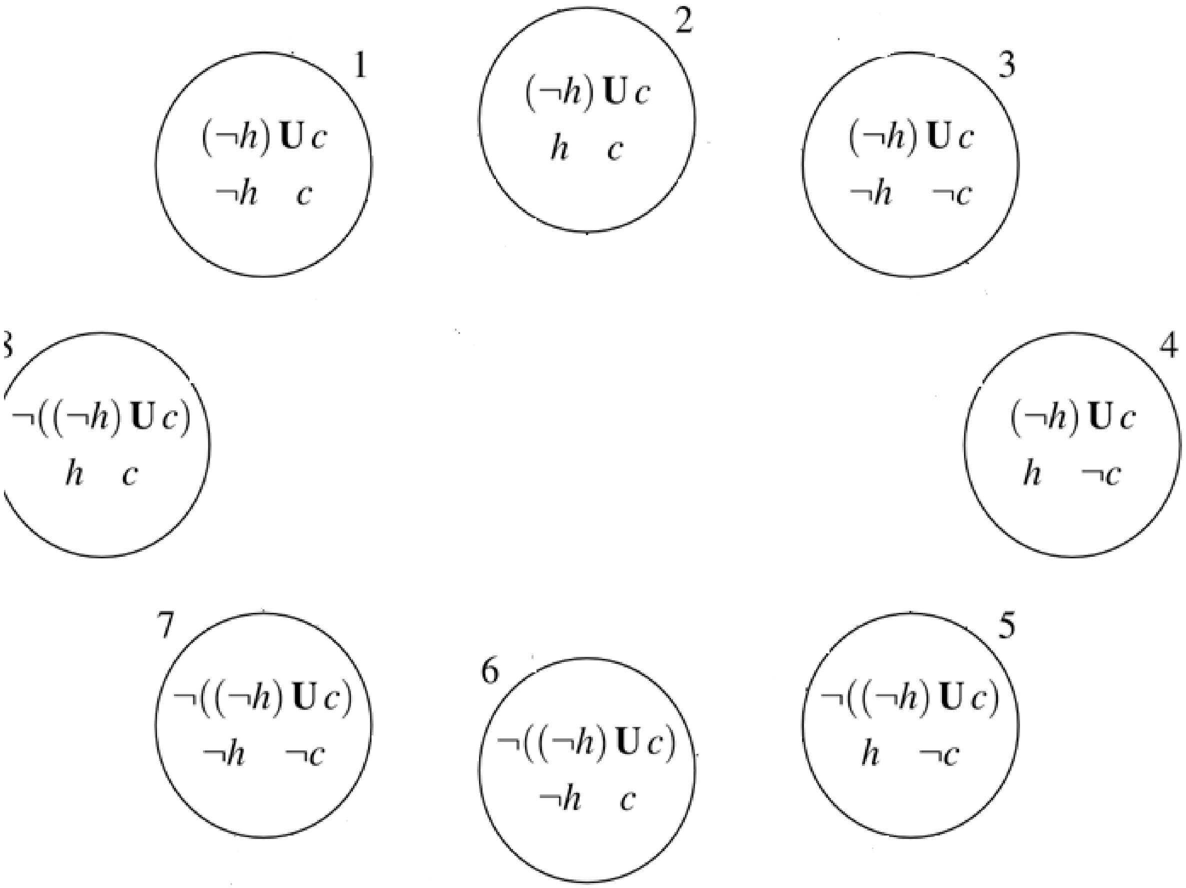
- Step 1: Define the state space $\mathcal{A}_\varphi$ based on φ

The set of all **good sets** of **cl(φ)** defines the state space of $\mathcal{A}_\varphi$

- cl(φ) are subformulas of φ and their negation
- ***Good sets:***
  S $\subseteq$ *cl*(φ) is ***good*** *in cl(φ)* if *S* is a maximal set of formulas in *cl(φ)* that is **consistent**
  - For all $\varphi_1 \in cl(\varphi)$: $\varphi_1 \in S \Leftrightarrow \neg \varphi_1 \notin S$

Give the state space of $\mathcal{A}_\varphi$ for the formula $\varphi = (\neg h \ \mathbf{U}\ c)$

$cl(\varphi) = \{\neg h, h, \neg c, c, (\neg h \ \mathbf{U}\ c), \neg(\neg h \ \mathbf{U}\ c)\}$



From LTL formula φ to GBA $\mathcal{A}_\varphi$

$\mathcal{A}_\varphi = (\mathcal{P}(AP), \mathbf{Q}, \mathbf{\Delta}, \mathbf{Q}^0, \mathbf{F})$

- $\mathbf{Q} \subseteq \mathcal{P}(cl(\varphi))$ is the set of all the good sets in $cl(\varphi)$.
- Next: $\mathbf{\Delta}$

Each state of $\mathcal{A}_\varphi$ is **labelled** with **a set of properties** that should be satisfied **on all paths starting at that state**

From LTL formula φ to GBA $\mathcal{A}_\varphi$

$\mathcal{A}_\varphi = (\mathcal{P}(AP), \mathbf{Q}, \mathbf{\Delta}, \mathbf{Q}^0, \mathbf{F})$

- For q, q' ∈Q and $\sigma \subseteq AP$, $(q,\sigma,q') \in \mathbf{\Delta}$ if:

1. $\sigma = q' \cap AP$ (push labels backwards)
2. $\boldsymbol{X}\varphi_1 \in q \Leftrightarrow \varphi_1 \in q'$
3. $\neg(\boldsymbol{X}\varphi_1) \in q \Leftrightarrow \neg\varphi_1 \in q'$

4. φ_1 U $\varphi_2 \in$ q ⇔ either $\varphi_2 \in$ q **or both** $\varphi_1 \in$ q **and** φ_1 U $\varphi_2 \in$ q'

5. $\neg(\varphi_1$ U $\varphi_2) \in$ q ⇔ either $\neg\varphi_2 \in$ q and **either** $\neg\varphi_1 \in$ q **or** $\neg(\varphi_1$ U $\varphi_2) \in$ q'

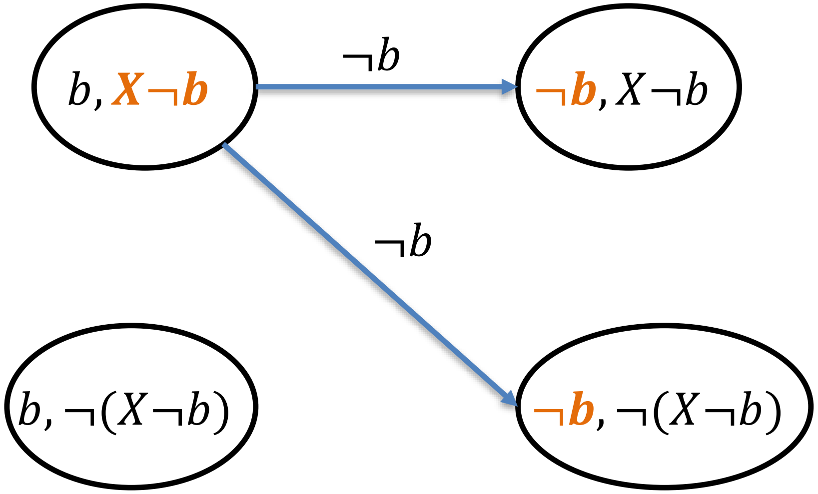
$\varphi = X\neg b$

ToDo

1. $\sigma = q' \cap AP$ (push labels backwards)
2. $\boldsymbol{X}\varphi_1 \in q \Leftrightarrow \varphi_1 \in q'$
3. $\neg(\boldsymbol{X}\varphi_1) \in q \Leftrightarrow \neg\varphi_1 \in q'$

Draw the state space and the transitions.

$cl(\varphi) = \{b, \neg b, X\neg b, \neg(X\neg b)\}$



SCOS Secure & Correct Systems

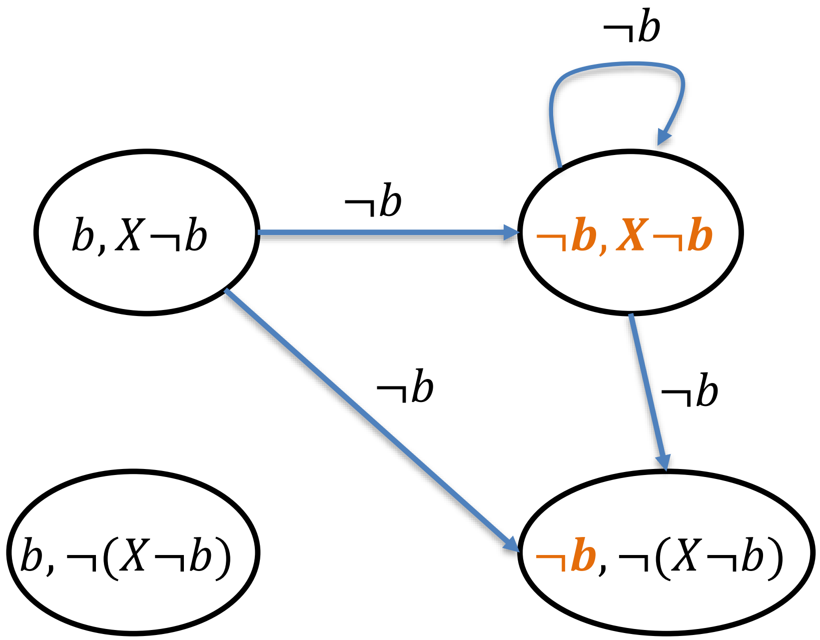
$\varphi = X\neg b$

ToDo

1. $\sigma = q' \cap AP$ (push labels backwards)
2. $\boldsymbol{X}\varphi_1 \in q \Leftrightarrow \varphi_1 \in q'$
3. $\neg(\boldsymbol{X}\varphi_1) \in q \Leftrightarrow \neg\varphi_1 \in q'$

Draw the state space and the transitions.

$cl(\varphi) = \{b, \neg b, X\neg b, \neg(X\neg b)\}$



SCOS Secure & Correct Systems

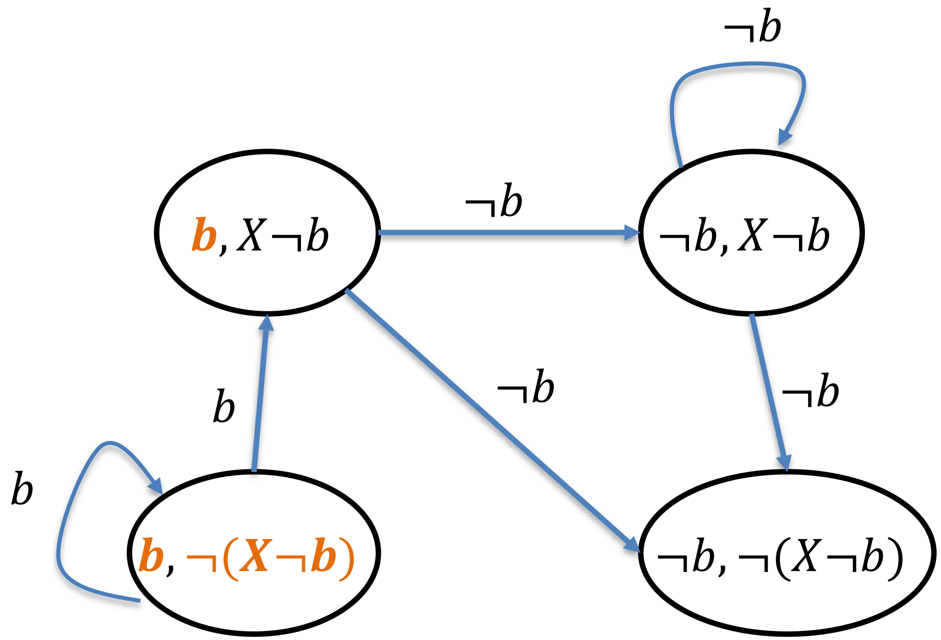
$\varphi = X\neg b$

ToDo

1. $\sigma = q' \cap AP$ (push labels backwards)
2. $\boldsymbol{X}\varphi_1 \in q \Leftrightarrow \varphi_1 \in q'$
3. $\neg(\boldsymbol{X}\varphi_1) \in q \Leftrightarrow \neg\varphi_1 \in q'$

Draw the state space and the transitions.

$cl(\varphi) = \{b, \neg b, X\neg b, \neg(X\neg b)\}$



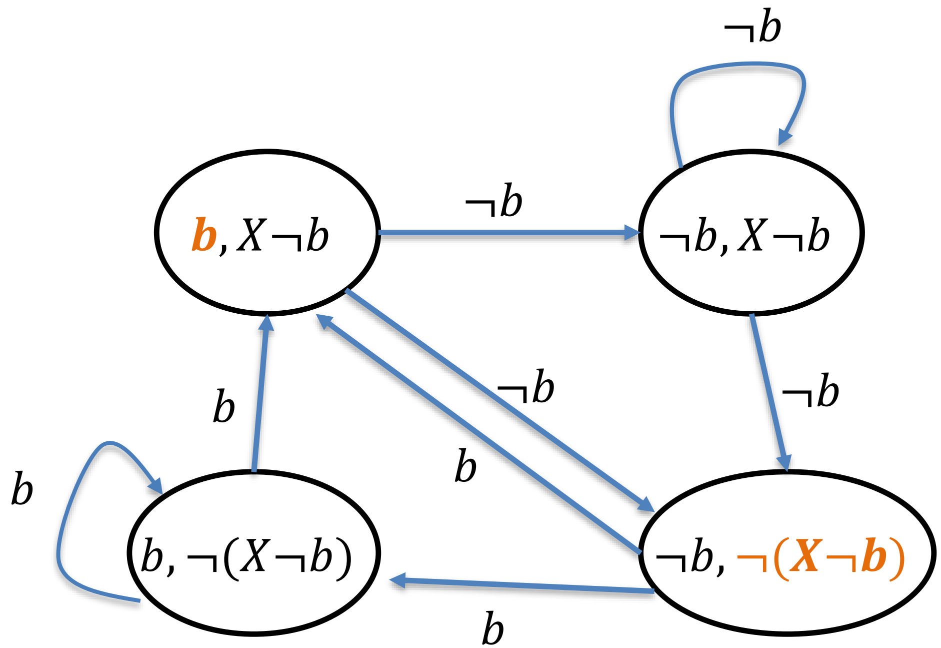
$\varphi = X\neg b$

1. $\sigma = q' \cap AP$ (push labels backwards)
2. $\boldsymbol{X}\varphi_1 \in q \Leftrightarrow \varphi_1 \in q'$
3. $\neg(\boldsymbol{X}\varphi_1) \in q \Leftrightarrow \neg\varphi_1 \in q'$

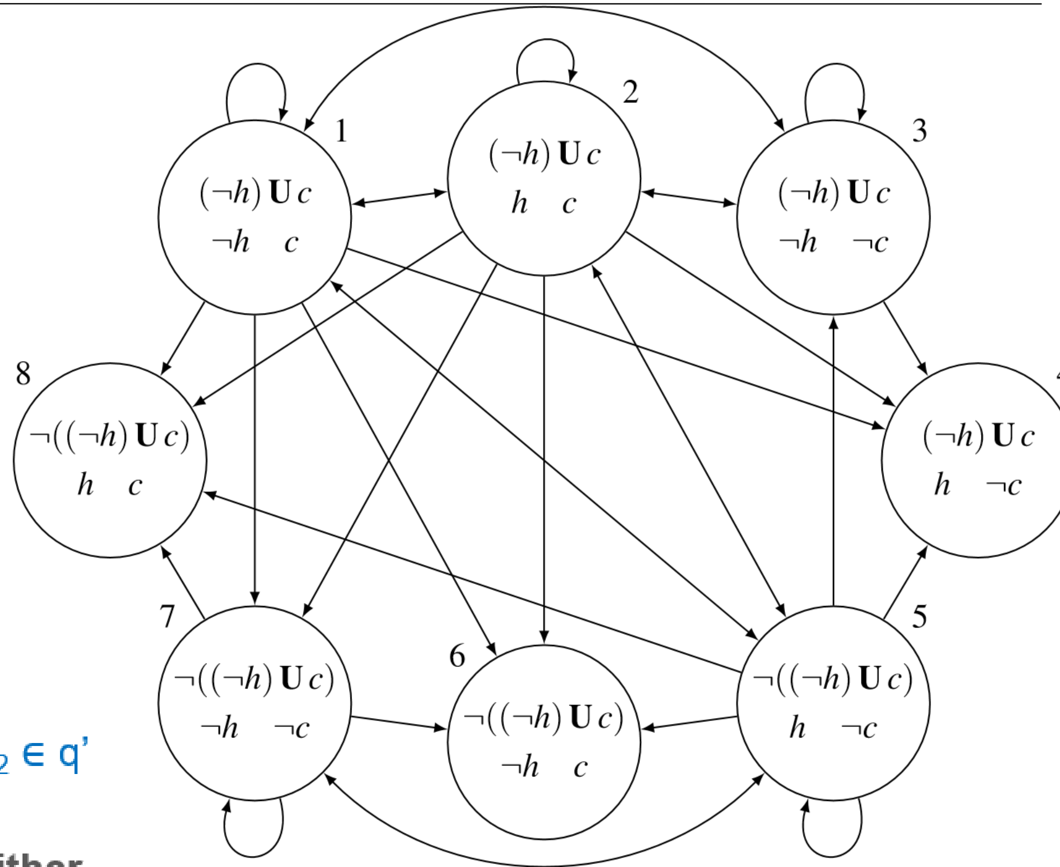
ToDo

Draw the state space and the transitions.

$cl(\varphi) = \{b, \neg b, X\neg b, \neg(X\neg b)\}$



φ= (¬*h* U *c*)



1. $\sigma = q' \cap AP$

3. φ_1 U $\varphi_2 \in$ q ⇔ either $\varphi_2 \in$ q **or both** $\varphi_1 \in$ q **and** φ_1 U $\varphi_2 \in$ q'

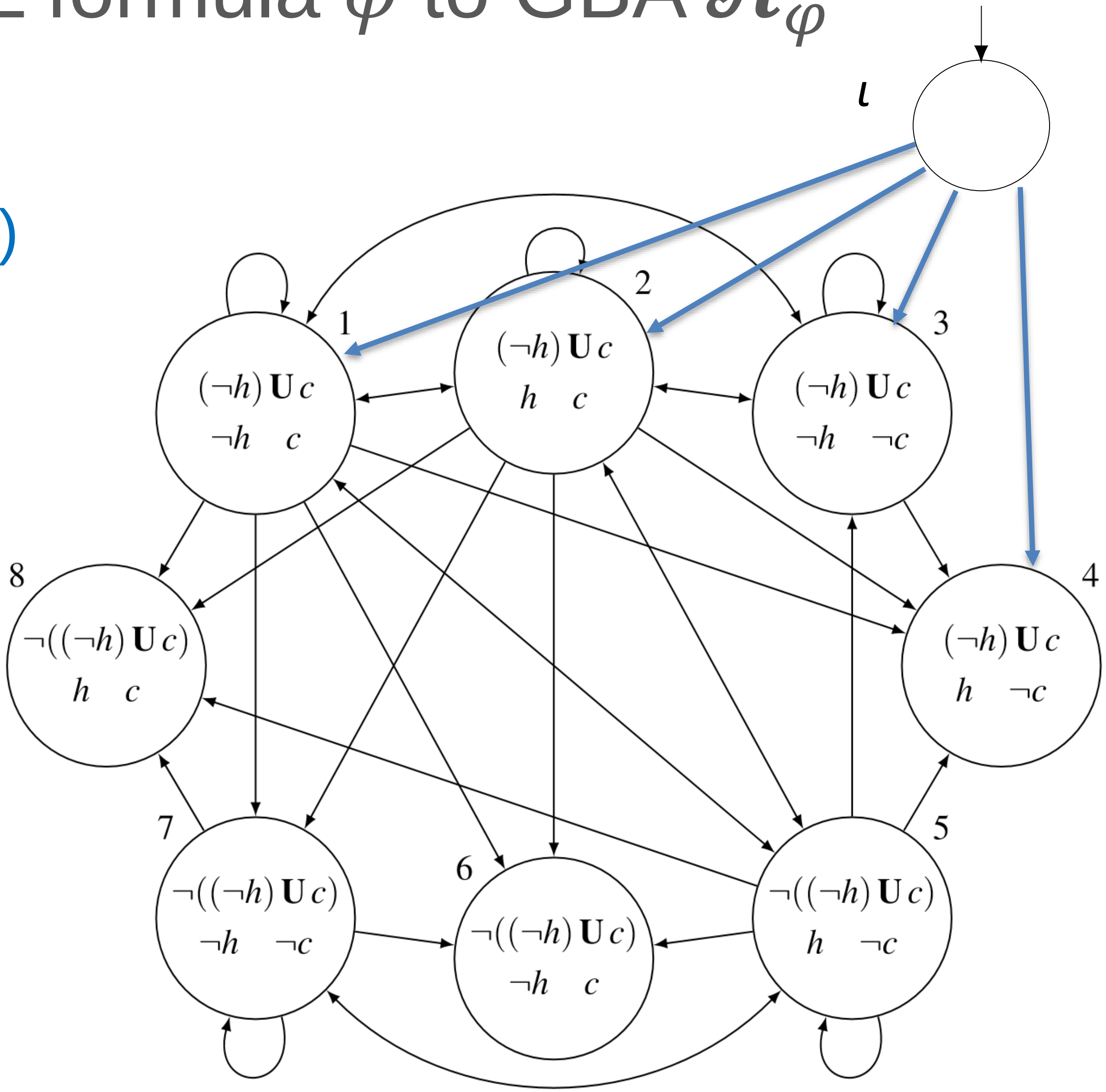
4. ¬(φ_1 U φ_2) ∈ q ⇔ either ¬$\varphi_2 \in$ q and **either** ¬ $\varphi_1 \in$ q **or** ¬(φ_1 U φ_2) ∈ q'

From LTL formula φ to GBA $\mathcal{A}_\varphi$

$\mathcal{A}_\varphi = (\mathcal{P}(AP), \mathbf{Q}, \mathbf{\Delta}, \{\iota\}, \mathbf{F})$

- $\mathbf{Q} \subseteq \mathcal{P}(cl(\varphi)) \cup \{\iota\}$ is the set of all the good sets in $cl(\varphi) \cup \{\iota\}$.

- $(\iota, \alpha, q) \in \mathbf{\Delta} \Leftrightarrow$ $\varphi \in q$ and $\sigma = q \cap AP$

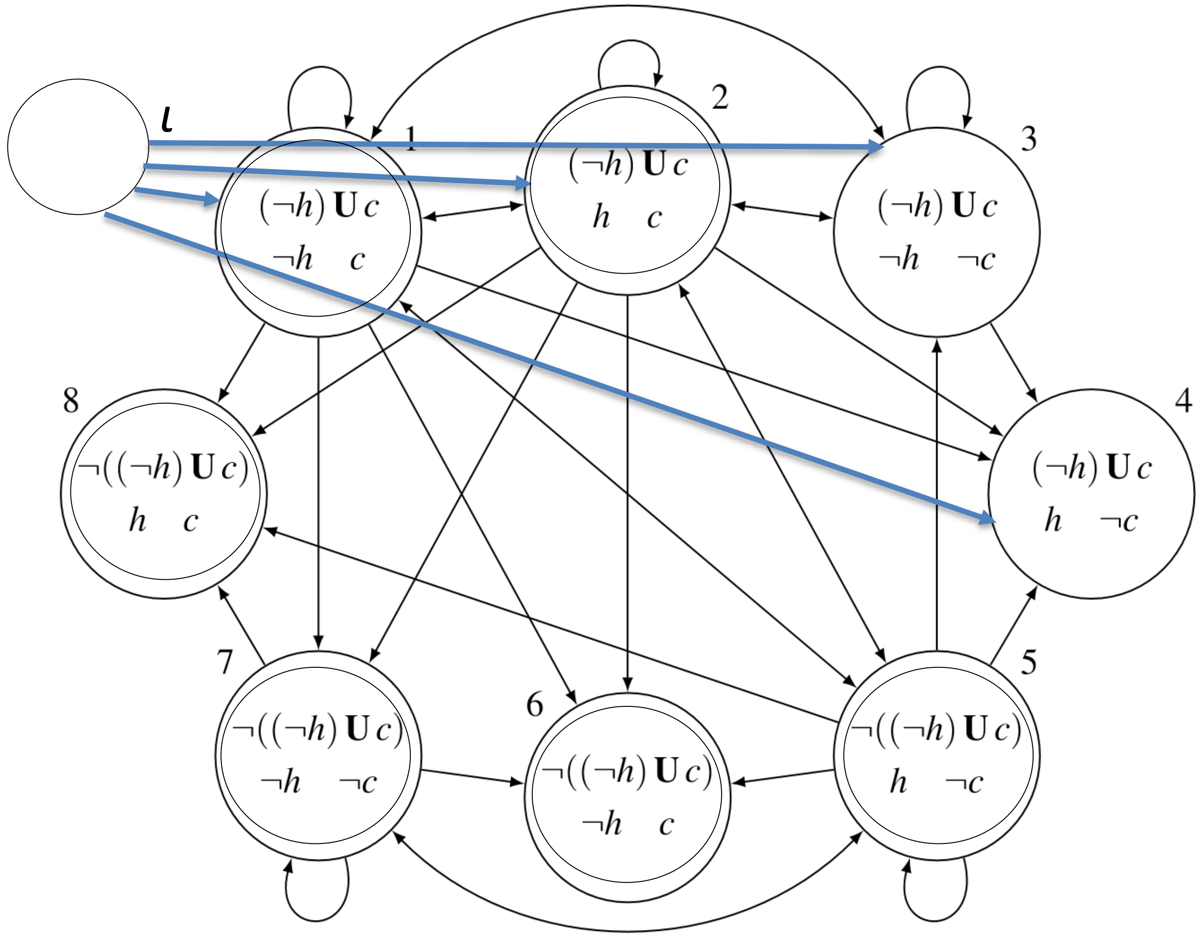


From LTL formula φ to GBA $\mathcal{A}_\varphi$

$\mathcal{A}_\varphi = (\mathcal{P}(AP), \mathbf{Q}, \mathbf{\Delta}, \{\iota\}, \mathbf{F})$

- $\mathbf{Q} \subseteq \mathcal{P}(cl(\varphi)) \cup \{\iota\}$ is the set of all the good sets in $cl(\varphi) \cup \{\iota\}$.
  - $(\iota, \alpha, q) \in \mathbf{\Delta} \Leftrightarrow \varphi \in q$ and $\sigma = q \cap AP$
- For every $\varphi_1 \mathbin{U} \varphi_2 \in cl(\varphi)$, $\mathbf{F}$ includes the set
  - $F_{\varphi_1 \mathbin{U} \varphi_2} = \{q \in \mathbf{Q} \mid \varphi_2 \in q \text{ or } \neg(\varphi_1 \mathbin{U} \varphi_2) \in q\}$.

$\varphi = (\neg h \text{ U } c)$



- F = {{1, 2, 5, 6, 7, 8}}

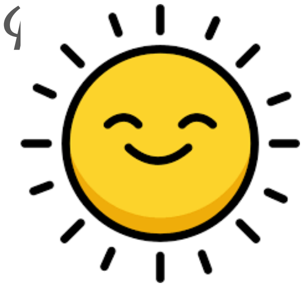
From LTL formula φ to GBA $\mathcal{A}_\varphi$

$\mathcal{A}_\varphi = (\mathcal{P}(AP), \mathbf{Q}, \mathbf{\Delta}, \{\iota\}, \mathbf{F})$

- $\mathbf{Q} \subseteq \mathcal{P}(cl(\varphi)) \cup \{\iota\}$ is the set of all the good sets in $cl(\varphi) \cup \{\iota\}$.
  - $(\iota, \alpha, q) \in \mathbf{\Delta} \Leftrightarrow \varphi \in q$ and $\sigma = q \cap AP$
- For every $\varphi_1 \,\mathsf{U}\, \varphi_2 \in cl(\varphi)$, $\mathbf{F}$ includes the set
  - $F_{\varphi_1 \mathsf{U} \varphi_2} = \{q \in \mathbf{Q} \mid \varphi_2 \in q \text{ or } \neg(\varphi_1 \,\mathsf{U}\, \varphi_2) \in q\}$.

- What is the complexity?

From LTL formula φ to GBA $\mathcal{A}_\varphi$



$\mathcal{A}_\varphi = (\mathcal{P}(AP), \mathbf{Q}, \mathbf{\Delta}, \{\iota\}, \mathbf{F})$

- $\mathbf{Q} \subseteq \mathcal{P}(cl(\varphi)) \cup \{\iota\}$ is the set of all the good sets in $cl(\varphi) \cup \{\iota\}$.
  - $(\iota, \alpha, q) \in \mathbf{\Delta} \Leftrightarrow \varphi \in q$ and $\sigma = q \cap AP$
- For every $\varphi_1 \, U \, \varphi_2 \in cl(\varphi)$, $\mathbf{F}$ includes the set
  - $F_{\varphi_1 U \varphi_2} = \{q \in \mathbf{Q} \mid \varphi_2 \in q \text{ or } \neg(\varphi_1 \, U \, \varphi_2) \in q\}$.

- What is the complexity?
  - $\mathcal{A}_\varphi$ is **always exponential** in the size of φ.


THANK
YOU!