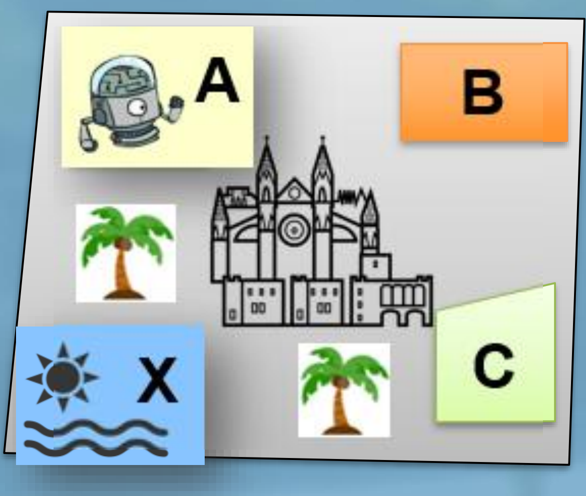
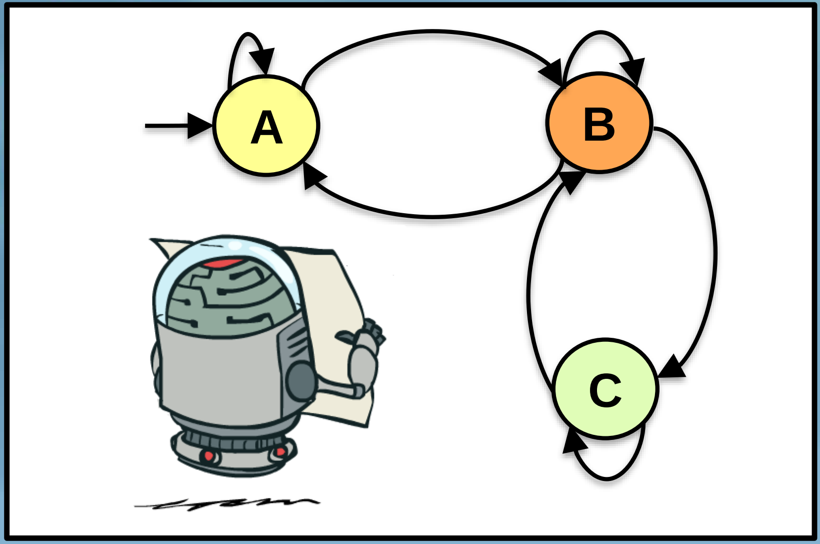
Graz University of Technology
Institute for Applied Information
Processing and Communications

Automata and LTL Model Checking
Part-2

Bettina Könighofer





Model Checking SS22

May 12 2022

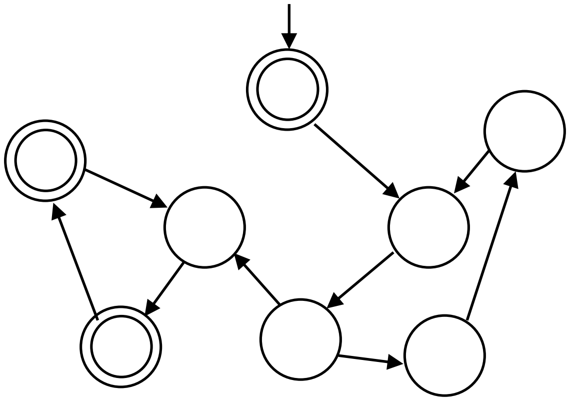
Outline

- 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 $\mathcal{L}(\mathcal{B})$

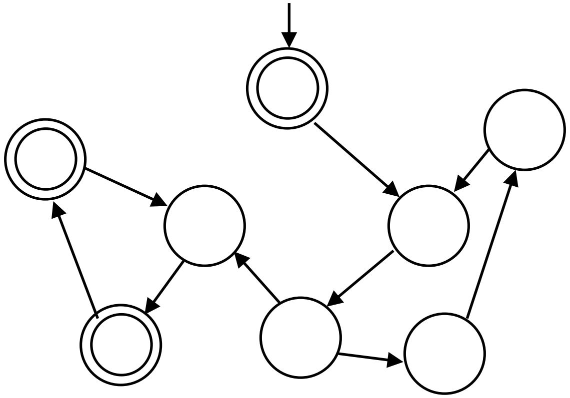
- An infinite run $\boldsymbol{\rho}$ is accepting $\Leftrightarrow$ it visits an accepting state an infinite number of times.
  - **$\mathrm{inf}(\boldsymbol{\rho}) \cap \mathbf{F} \neq \emptyset$**

ToDo How to check for **L(A) = Ø?**



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

- An infinite run $\boldsymbol{\rho}$ is accepting $\Leftrightarrow$ it visits an accepting state an infinite number of times.
  - **$\inf(\rho) \cap \mathbf{F} \neq \emptyset$**
- How to check for **L(A) = ∅?**
- Empty if there is no reachable accepting state on a cycle.



Non-emptiness ⇔ Existence of reachable accepting cycles

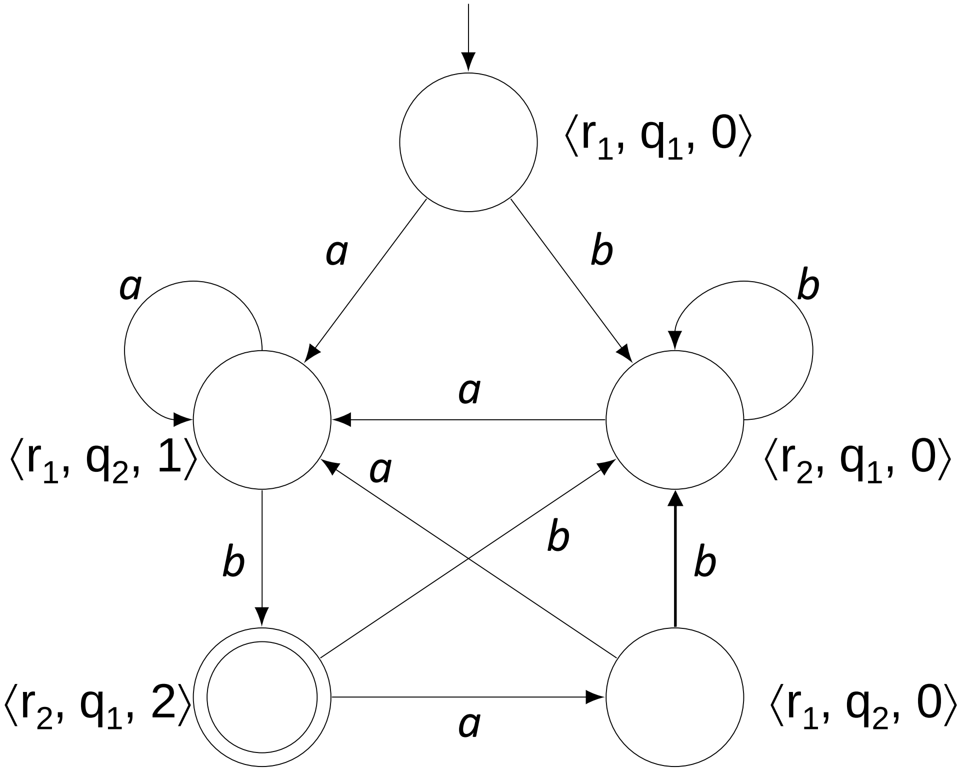
Lemma: ***Let $\mathcal{B} = (\Sigma, Q, \Delta, Q^0, F)$ be a Büchi automaton. The following conditions are equivalent:***

- *$\mathcal{L}(\mathcal{B})$ is nonempty.*
- *$\mathcal{B}$ contains a strongly connected component **C**, which includes an accepting state. Moreover, **C** is reachable from an initial state of $\mathcal{B}$.*
- *The graph induced by $\mathcal{B}$ contains a path from an initial state of $\mathcal{B}$ to a state $t \in F$ and a path from t back to itself.*

Example

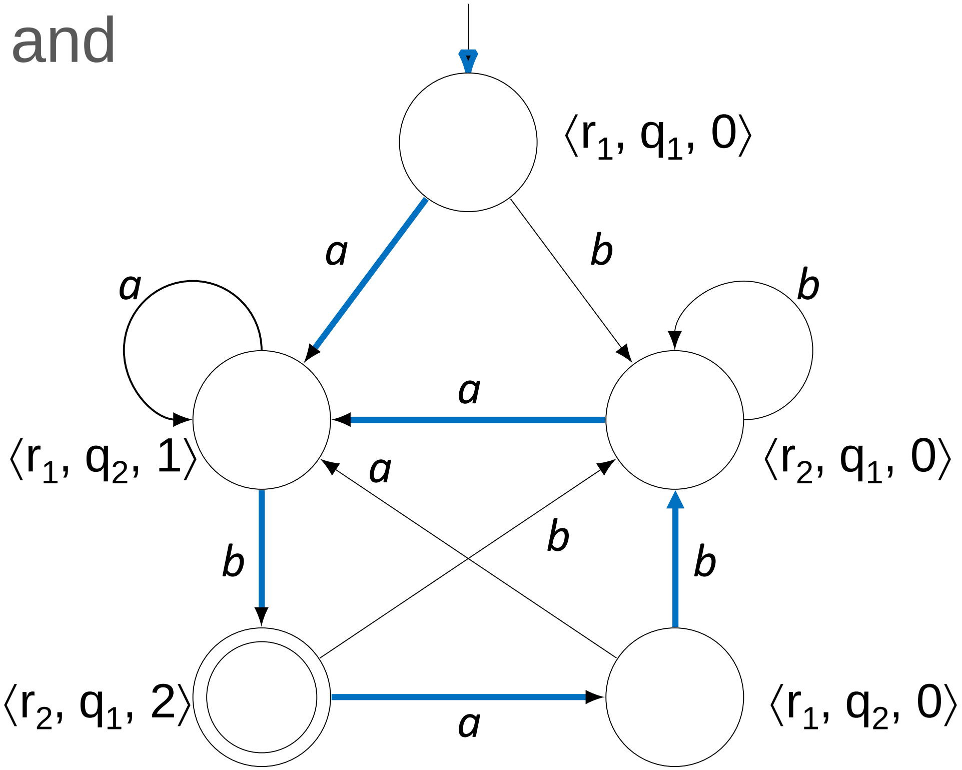
ToDo

- Is the language $\mathcal{L}(\mathcal{B})$ empty?
- If not, what is the $\mathcal{L}(\mathcal{B})$



Example

- The language $\mathcal{L}(\mathcal{B})$ is nonempty.
  - $\mathcal{L}(\mathcal{B})$ = {inf number of a's and inf number of b's}
- <r$_2$,q$_1$,2> is accepting and reachable from <r$_1$,q$_1$,0> and reachable from itself



Outline

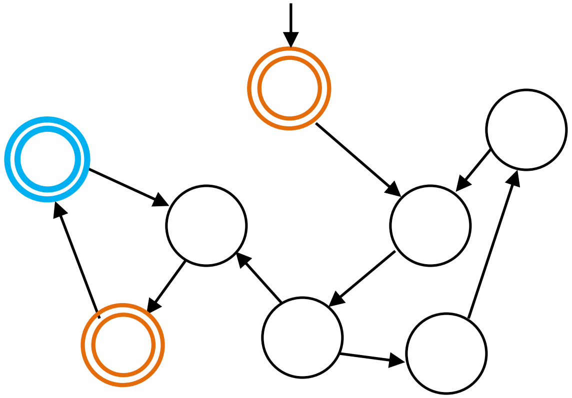
- Finite automata on finite words
- Automata on infinite words (Büchi automata)
- Deterministic vs non-deterministic Büchi automata
- Intersection of Büchi automata
- Checking emptiness of Büchi automata
- Generalized Büchi automata
- Automata and Kripke Structures
- Model checking using automata
- Translation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

Generalized Büchi automata

- Have several sets of accepting states

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

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



Translation from Generalized Büchi to Büchi

- Given $\boldsymbol{\mathcal{B}} = (\mathbf{\Sigma},\mathbf{Q}_1,\mathbf{\Delta}_1,\mathbf{Q}_1^0,\mathbf{F}_1)$ with $\mathbf{F} = \{P_1, \ldots, P_k\}$

- How does it work to construct a Büchi automaton $\boldsymbol{\mathcal{B}}'$ that accepts the same language?

Translation from Generalized Büchi to Büchi

- $\mathcal{B} = (\mathbf{\Sigma},\mathbf{Q}_1,\mathbf{\Delta}_1,\mathbf{Q}_1{}^0,\mathbf{F}_1)$ with $\mathbf{F} = \{P_1, \ldots, P_k\}$
- $\mathcal{B}' = (\mathbf{\Sigma}, \mathbf{Q} \times \{0,1,\ldots,k\}, \mathbf{\Delta}', \mathbf{Q}^0 x0, \mathbf{Q} \times k)$ with:

Translation from Generalized Büchi to Büchi

- $\mathcal{B} = (\boldsymbol{\Sigma}, \mathbf{Q}_1, \boldsymbol{\Delta}_1, \mathbf{Q}_1^0, \mathbf{F}_1)$ with $\mathbf{F} = \{P_1, \ldots, P_k\}$
- $\mathcal{B}' = (\boldsymbol{\Sigma}, \mathbf{Q} \times \{0,1,\ldots,k\}, \boldsymbol{\Delta}', \mathbf{Q}^0 x0, \ \mathbf{Q} \times k)$ with:
- The transition relation $\boldsymbol{\Delta}'$:
  $((q,x),a,(q',y)) \in \boldsymbol{\Delta}'$ when $(q,a,q') \in \boldsymbol{\Delta}$ 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)

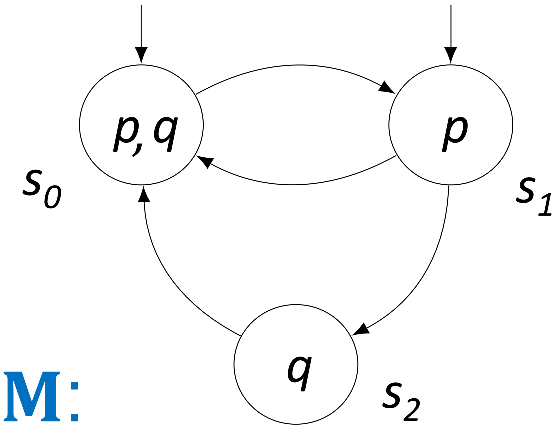
Outline

- 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

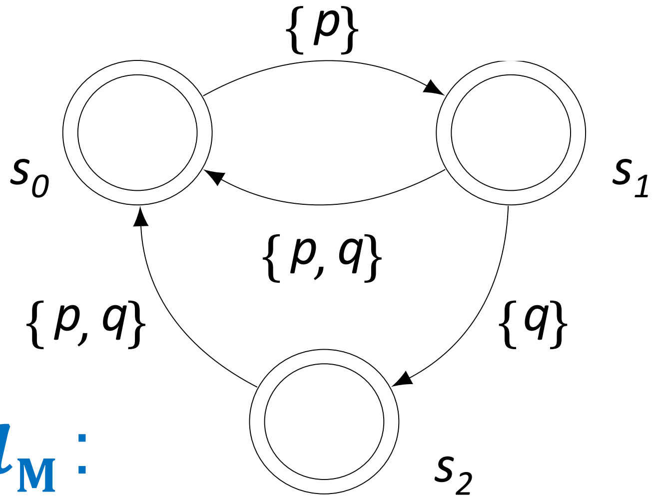
Kripke Structure $\mathbf{M}$ to Büchi Automaton $\mathcal{A}_{\mathbf{M}}$

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

$\mathbf{M}$:



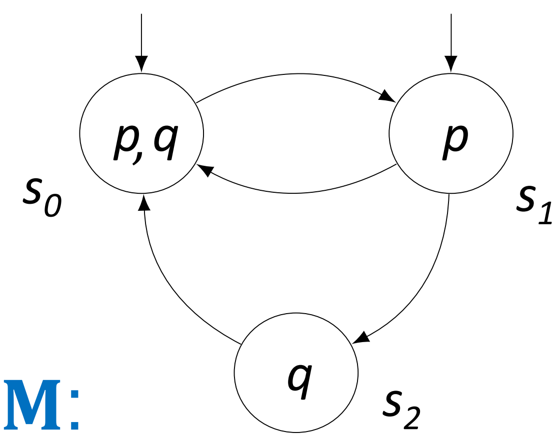
$\mathcal{A}_{\mathbf{M}}$:



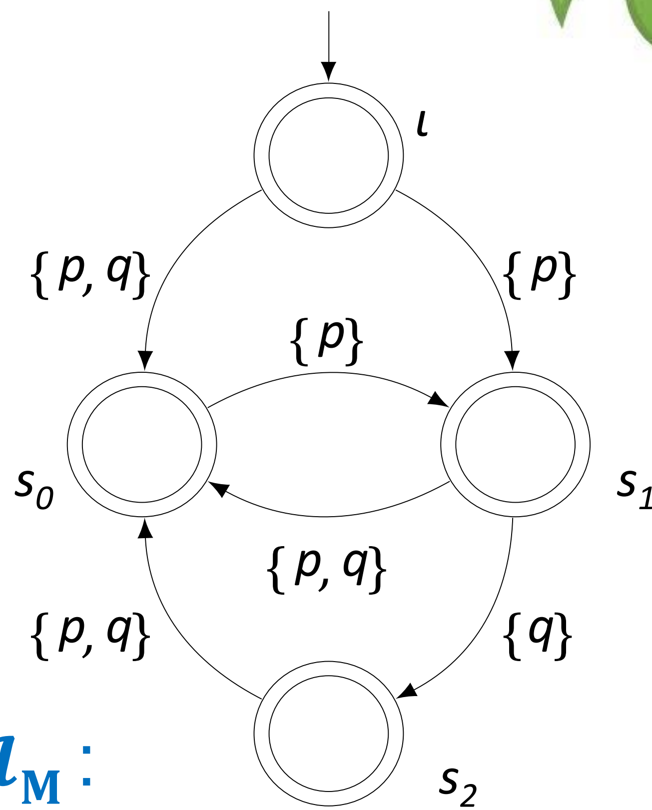
Kripke Structure **M** to Büchi Automaton $\mathcal{A}_{\mathbf{M}}$

- Move labels to incoming transitions
- All states are accepting

M:



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

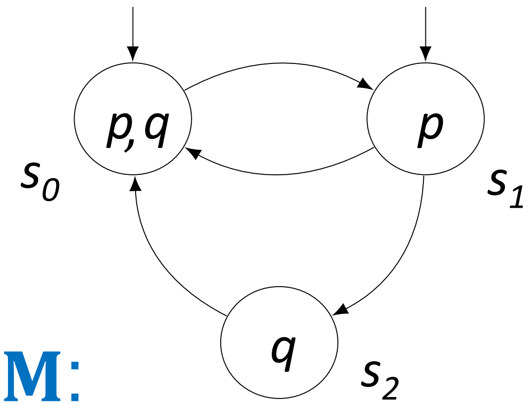


Automata and Kripke Structures

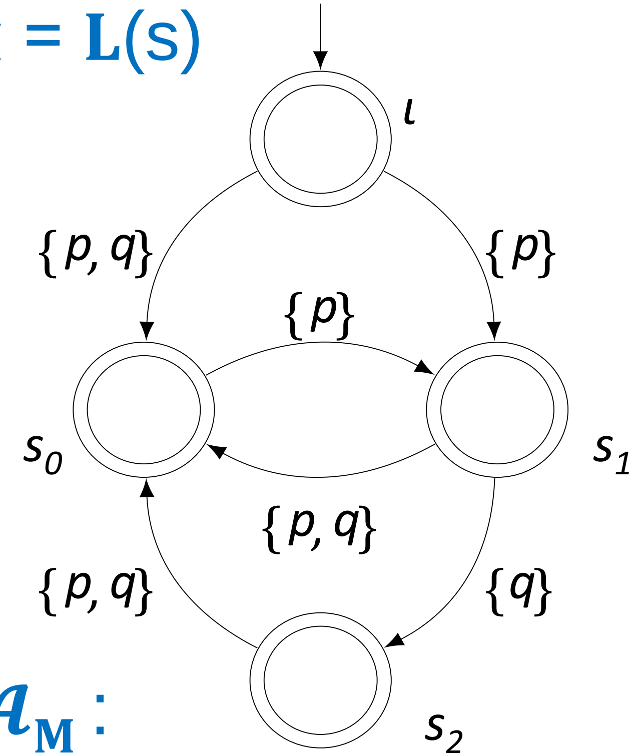
$\mathbf{M} = (\mathbf{S}, \mathbf{S}_0, \mathbf{R}, AP, \mathbf{L}) \Rightarrow \mathcal{A}_\mathbf{M} = (\Sigma, \mathbf{S}\cup\{\iota\}, \Delta, \{\iota\}, \mathbf{S}\cup\{\iota\})$,
where $\Sigma = P(AP)$.

- $(s,\alpha,s') \in \Delta$ 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:



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

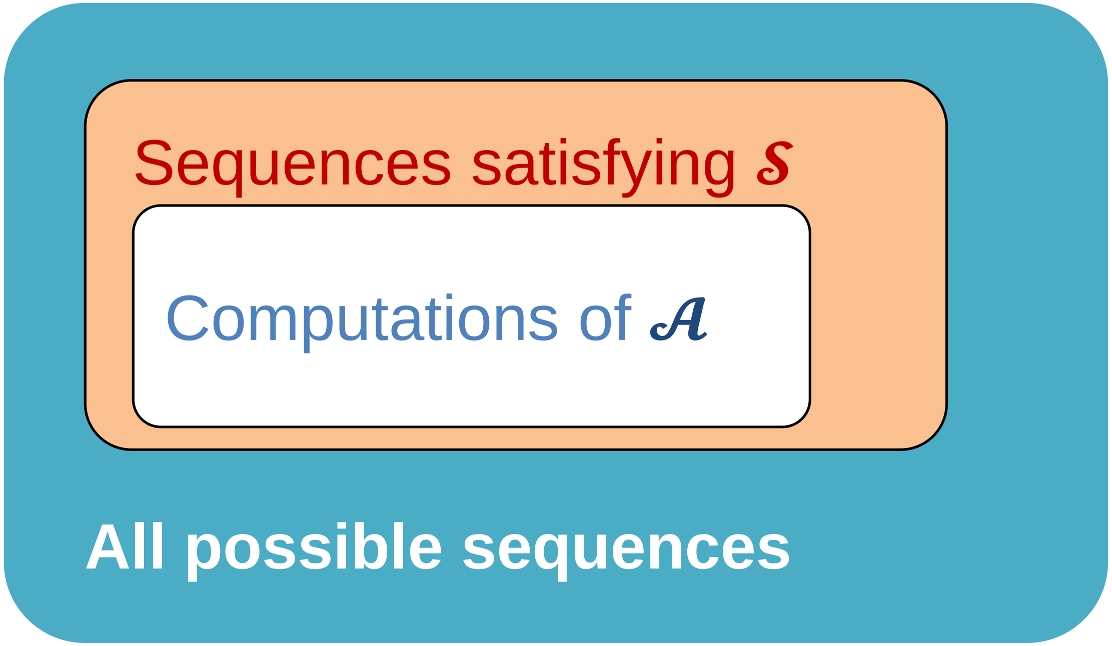


Outline

- 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

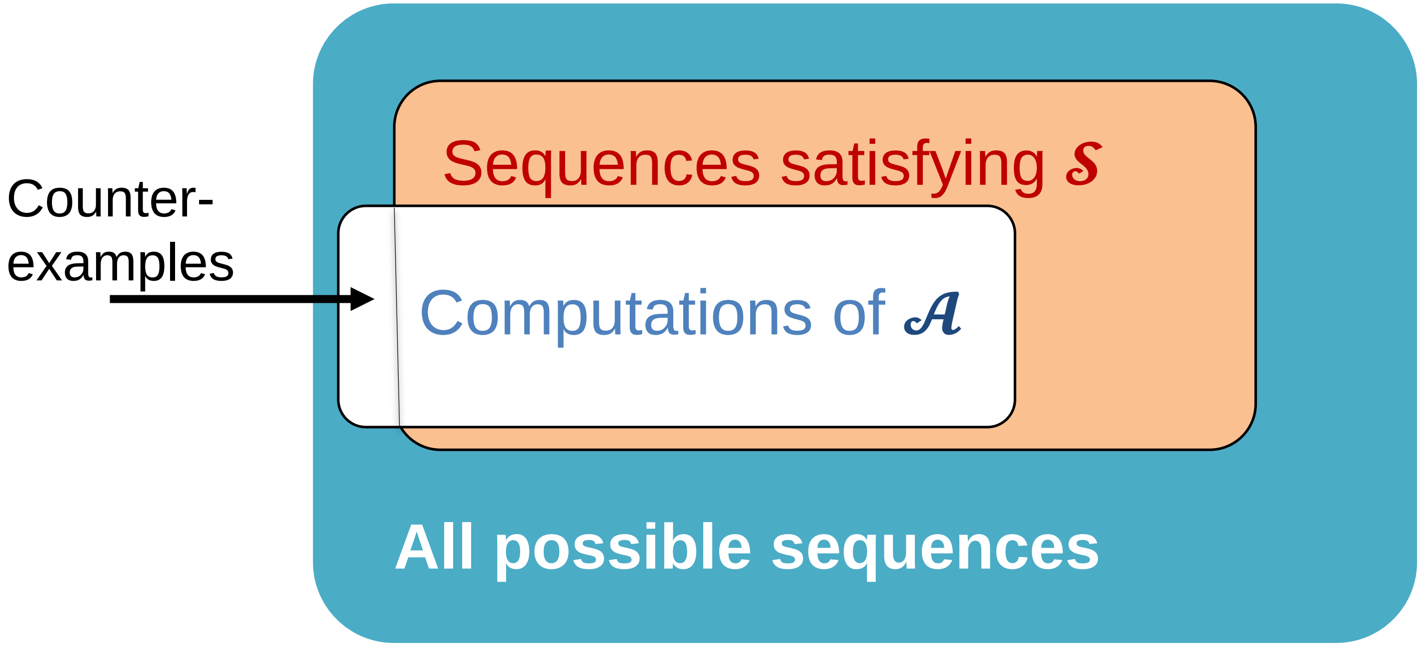
Model Checking when system $\mathcal{A}$ and spec $\mathcal{S}$ are given as Büchi automata

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



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

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



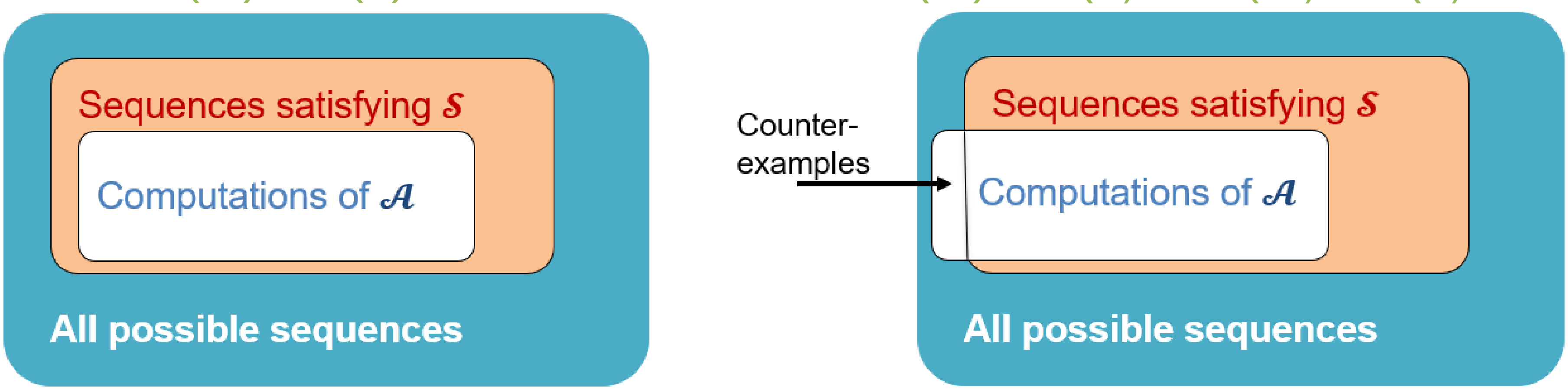
Model Checking when system $\mathcal{A}$ and spec $\mathcal{S}$ are given as Büchi automata

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

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

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

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



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 $\mathcal{S}$

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

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 $\models \varphi$

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

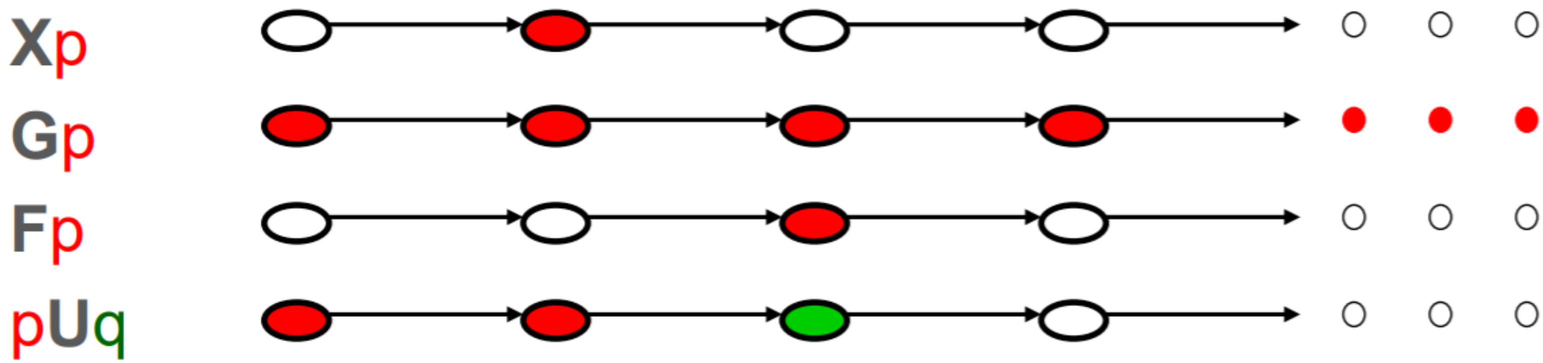
- Finite automata on finite words
- Automata on infinite words (Büchi automata)
- Deterministic vs non-deterministic Büchi automata
- Intersection of Büchi automata
- Checking emptiness of Büchi automata
- Generalized Büchi automata
- Automata and Kripke Structures
- Model checking using automata
- **Translation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

Translation of LTL to Büchi automata

Given an LTL formula φ, construct a generalized Büchi automaton $\mathcal{A}_\varphi$

- $\mathcal{A}_\varphi$ accepts exactly all the traces that satisfy φ

Recall LTL Semantics



SCOS
Secure & Correct Systems

Translation of LTL to Büchi automata

Given an LTL formula φ, construct a generalized Büchi automaton $\mathcal{A}_\varphi$

1. Translate φ into generalized Büchi Automaton
2. Translate generalized Büchi to Büchi automaton

Rewriting

- Algorithm only handles
  - $\neg, \wedge, \vee, X, U$
  - 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: Based on φ, we define the state space of $\mathcal{A}_\varphi$
- Each state of the automata is **labelled** with **a set of properties/sub-formulas** that should be satisfied **on paths starting at that state**

Closure of an LTL formula φ – cl(φ)

- cl(φ)
  - … subformulas of φ and their negation
  - … subsets of cl(φ) define state space of $\mathcal{A}_\varphi$

Closure of an LTL formula φ – cl(φ)

- cl(φ)
  - … 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 $\varphi_1\ U\ \varphi_2 \in cl(\varphi), then\ \varphi_1 \in cl(\varphi)\ and\ \varphi_2 \in cl(\varphi)$.

Closure of an LTL formula φ – cl(φ)

- cl(φ)
  - … subformulas of φ and their negation

- $\varphi \coloneqq (\neg p \; U \; ((Xq) \vee r))$
- Compute cl(φ)

Closure of an LTL formula φ

- cl(φ)
  - … subformulas of φ and their negation

- $\varphi \coloneqq (\neg \mathrm{p}\ \mathrm{U}\ ((\mathrm{Xq}) \vee \mathrm{r}))$
- $cl((\neg pU((Xq) \vee r))) =$

$$\left\{ \left(\neg pU\left((Xq) \vee r\right)\right), \neg(\neg pU((Xq) \vee r)), \right.$$
$$\neg p, p,$$
$$((Xq) \vee r), \neg((Xq) \vee r),$$
$$(Xq), \neg(Xq),$$
$$\left. q, \neg q, r, \neg r \ \right\}$$

SCOS
Secure & Correct Systems

Good sets in cl(φ)

- $S \subseteq cl(\varphi)$ is ***good*** *in* $cl(\varphi)$ if S is a maximal set of formulas in $cl(\varphi)$ that is **consistent**:

1. For all $\varphi_1 \in cl(\varphi)$: $\varphi_1 \in S \Leftrightarrow \neg \varphi_1 \notin S$,
2. For all $\varphi_1 \vee \varphi_2 \in cl(\varphi)$: $\varphi_1 \vee \varphi_2 \in S \Leftrightarrow$ at least one of φ_1, φ_2 is in S.

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

Give the state space Q of $\mathcal{A}_\varphi$ representing

$$\varphi = (a \vee X\neg b)$$

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. $X\varphi_1 \in q \Leftrightarrow \varphi_1 \in q'$
  3. $\varphi_1 \, U \, \varphi_2 \in q \Leftrightarrow$ either $\varphi_2 \in q$ or both $\varphi_1 \in q$ **and** $\varphi_1 \, U \, \varphi_2 \in q'$

$$\boxed{\varphi_1 U \varphi_2 \equiv \varphi_2 \vee (\varphi_1 \wedge X(\varphi_1 U \varphi_2))}$$

Note that the last condition also means that for all $\neg(\varphi_1 U \varphi_2) \in cl(\varphi)$, we have that $\neg(\varphi_1 \mathbf{U} \varphi_2) \in q$ iff $\neg\varphi_2 \in q$ and either $\neg\varphi_1 \in q$ or $\neg(\varphi_1 \mathbf{U} \varphi_2) \in q'$.

$\varphi = (a \vee X\neg b)$

$$\boxed{\boldsymbol{X}\varphi_1 \in q \Leftrightarrow \varphi_1 \in q'}$$

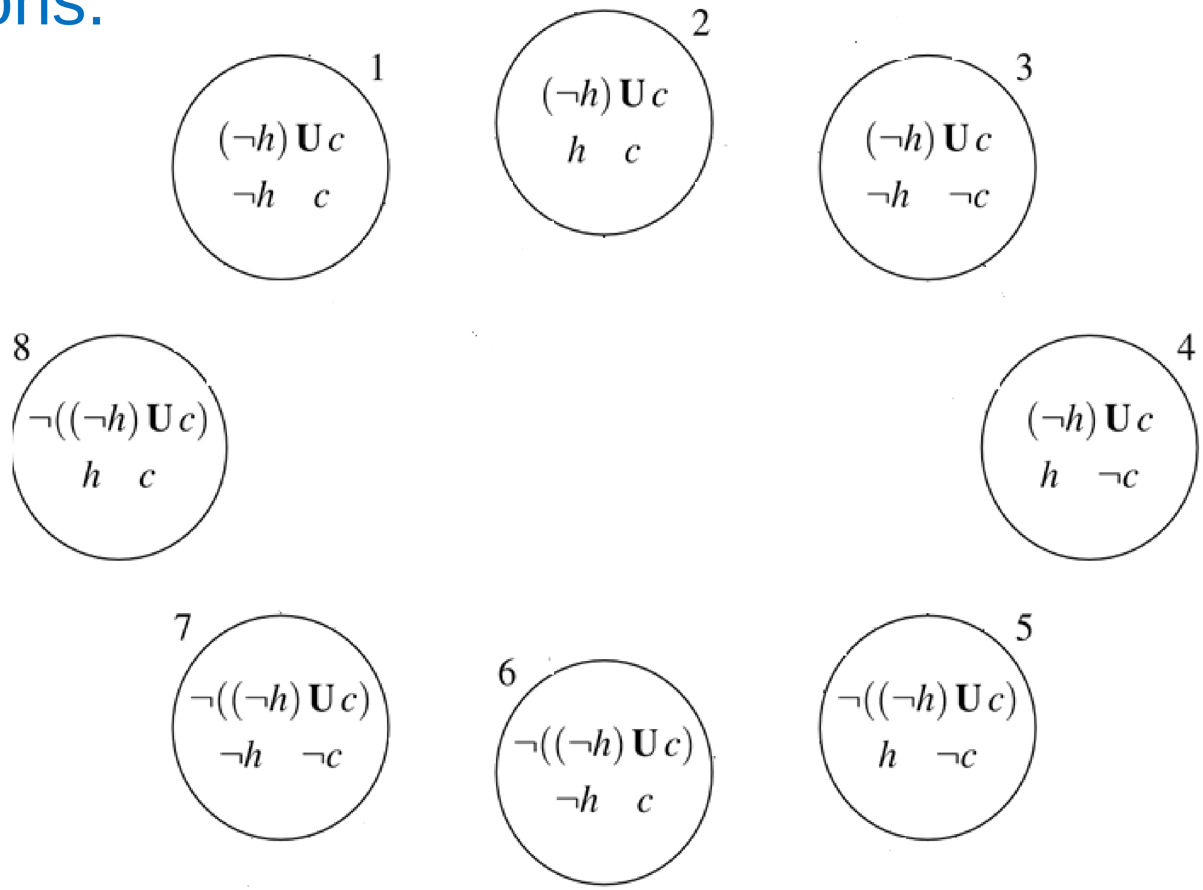
Draw the state space and the transitions.

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

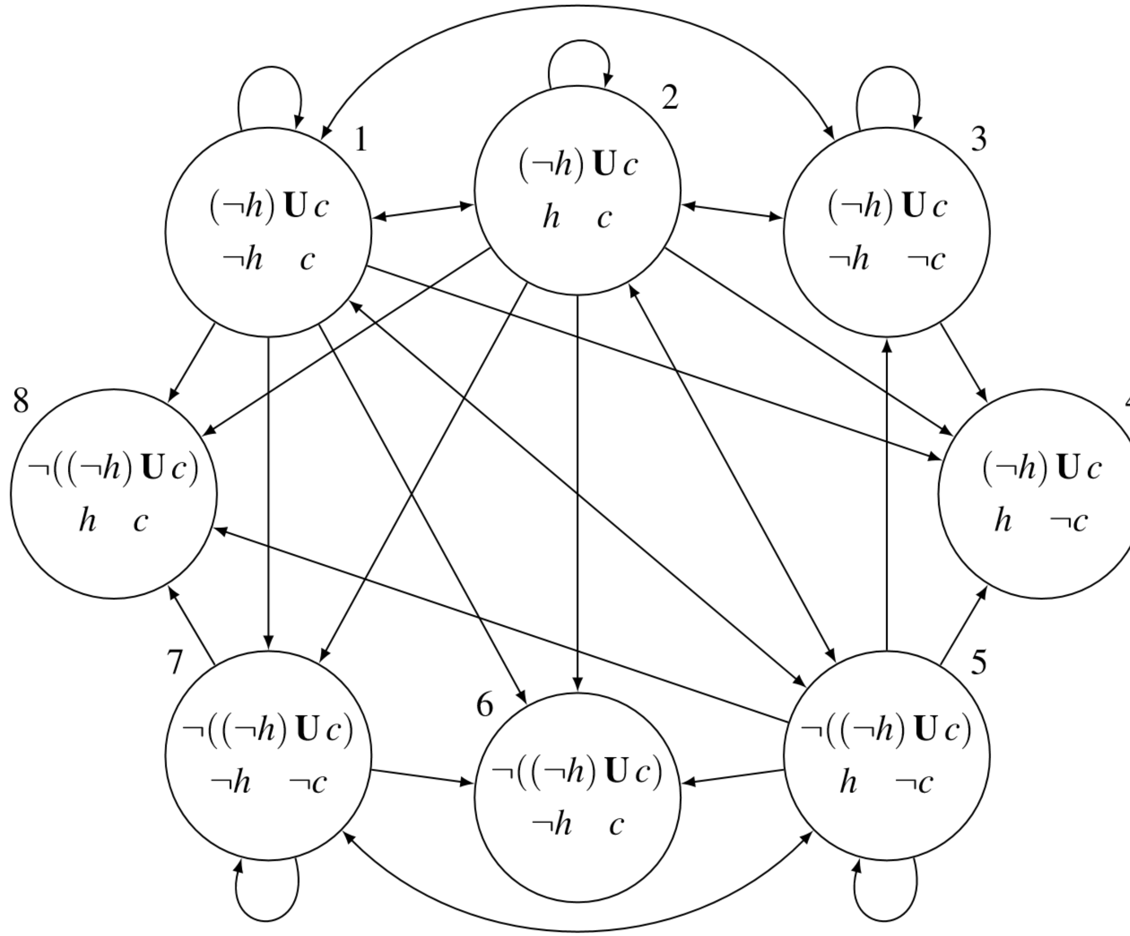
$\varphi_1 \; \mathrm{U} \; \varphi_2 \in q \Leftrightarrow$ either $\varphi_2 \in q$ or both $\varphi_1 \in q$ **and** $\varphi_1 \; \mathrm{U} \; \varphi_2 \in q'$

ToDo

Draw the transitions.



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



States 4, 6, 8 have no outgoing edges

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

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

- What are the initial states?

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 $\mathcal{A}_\varphi$

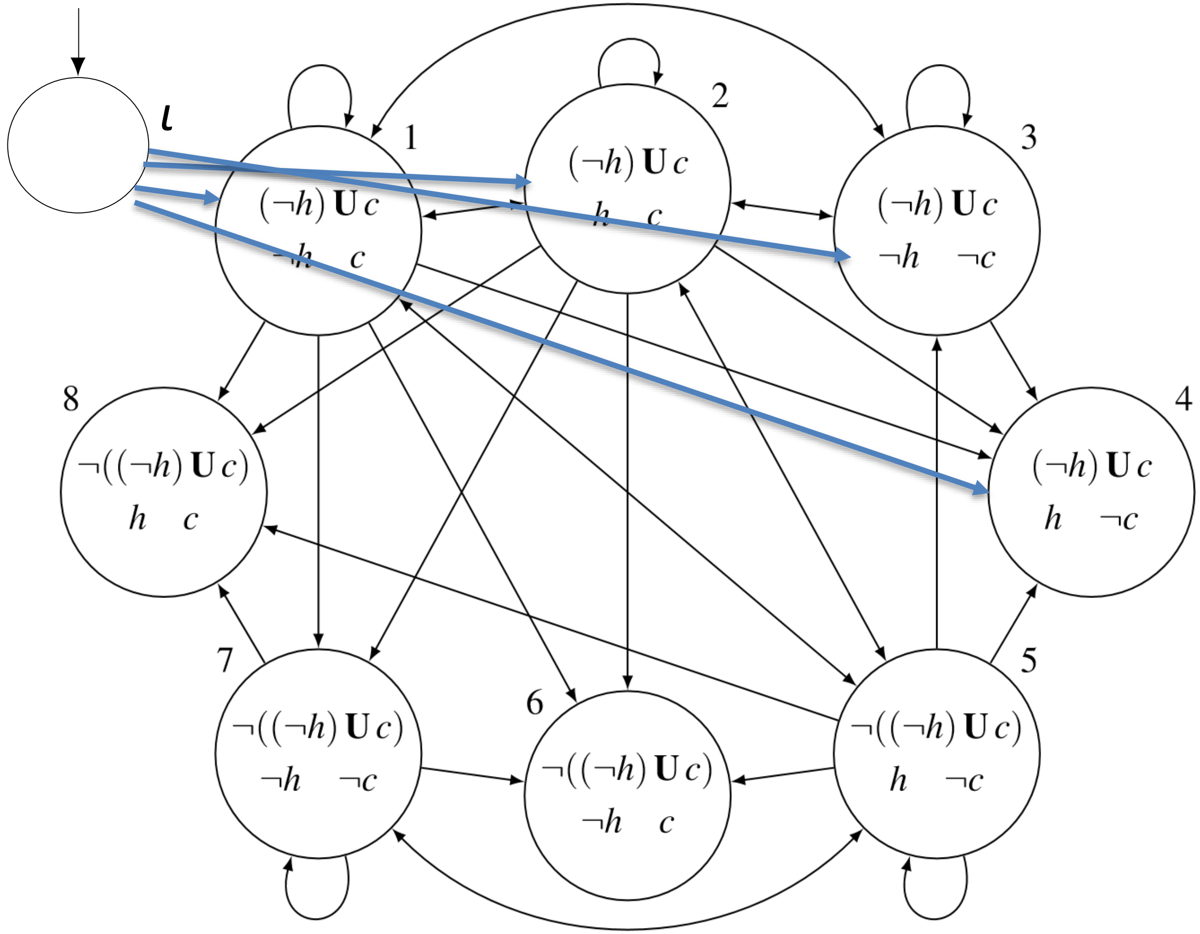
$\mathcal{A}_\varphi = (\mathcal{P}(AP), \mathbf{Q}, \mathbf{\Delta}, \{\iota\}, \mathbf{F})$

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

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

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

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

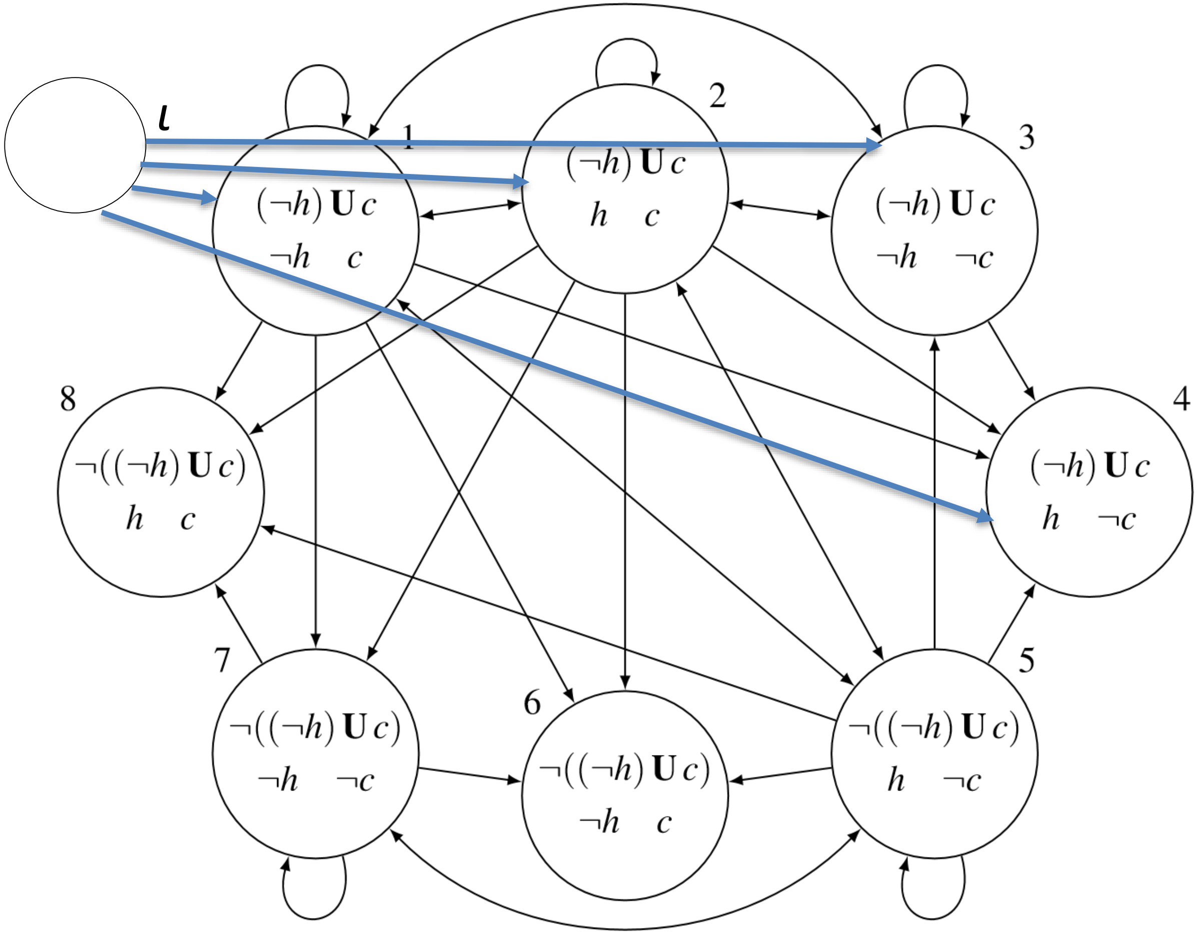


From LTL formula φ to GBA $\mathcal{A}_\varphi$

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

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

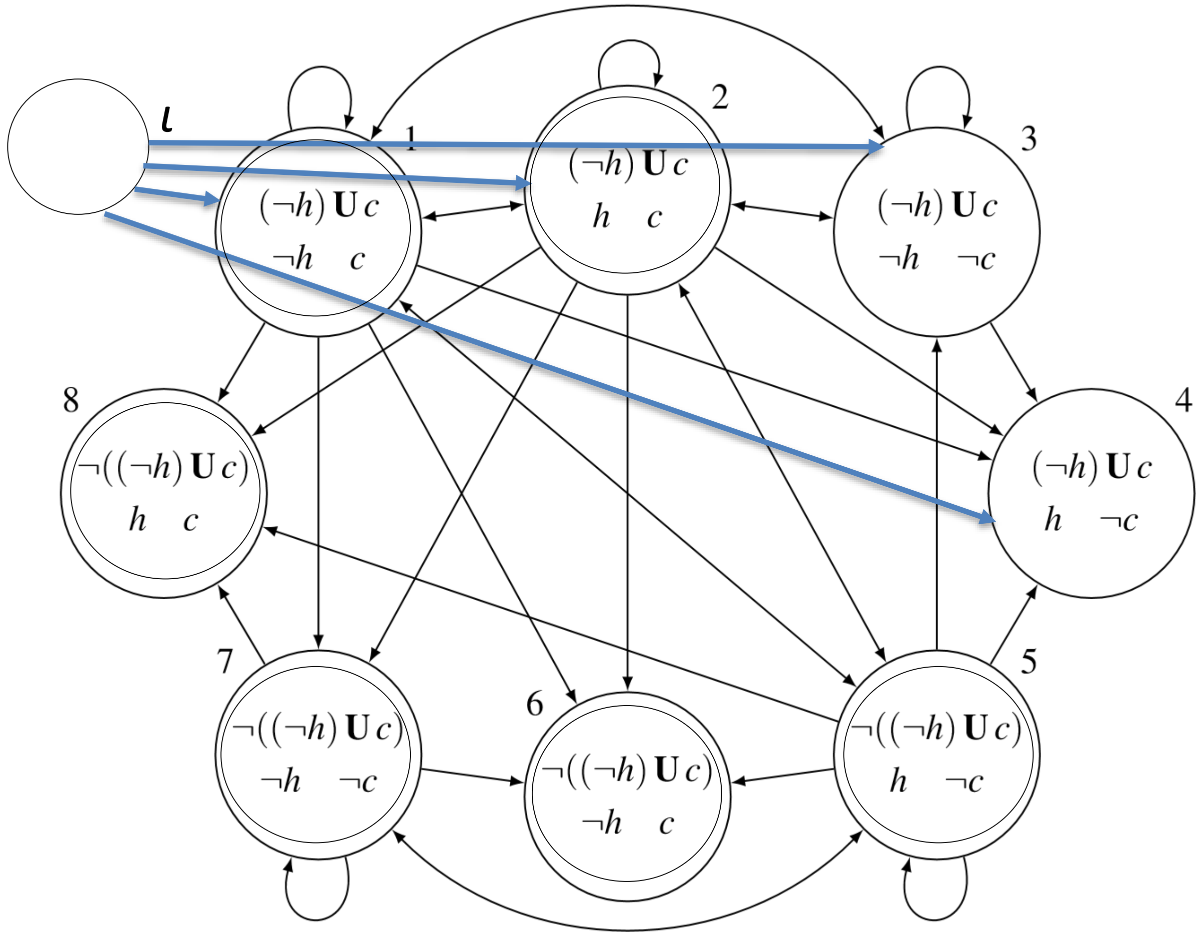
φ= (¬h U c)



ToDo

- What is F?

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



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

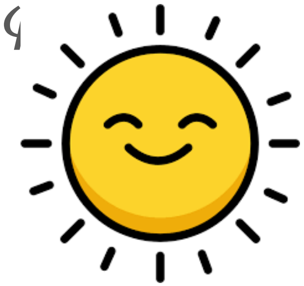
From LTL formula φ to GBA $\mathcal{A}_\varphi$

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

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

- What is the complexity?

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



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

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

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


THANK
YOU!