# Formal techniques for software and hardware verification

Lecturers:

Vladimir Zakharov

Vladislav Podymov

e-mail:

**valdus@yandex.ru**

2020, fall semester

# Lecture 7

Model checking problem for LTL

Model checking algorithms for LTL:
tableau-based,
automata-based

Büchi automata:
basic properties and generalizations

# LTL-formulae (reminder and new definitions)

$AP$ is a set of *atomic propositions* assumed to be given in other definitions

In this lecture the following syntax of *ltl-formulae* is used:

$$\varphi \quad ::= \quad a \mid (\varphi \,\&\, \varphi) \mid (\neg\varphi) \mid (\mathbf{X}\varphi) \mid (\varphi \mathbf{U} \varphi),$$

where $\varphi$ is an ltl-forula, and $a \in AP$

In examples (but not in proofs)
other known connectives are used as well: $\vee$, $\rightarrow$, **F**, **G**, **R**

$$\varphi \vee \psi \equiv \neg(\neg\varphi \,\&\, \neg\psi) \qquad \varphi \rightarrow \psi \equiv \neg\varphi \vee \psi$$

$$\mathbf{F}\varphi \equiv \mathbf{true}\mathbf{U}\varphi \qquad \mathbf{G}\varphi \equiv \neg\mathbf{F}\neg\varphi \qquad \varphi\mathbf{R}\psi \equiv \neg(\neg\varphi\mathbf{U}\neg\psi)$$

In *Lecture 4* all ltl-formulae started with the quantifier **A**

In this lecture the quantifier **A** is omitted due to its redundancy

# LTL-formulae (reminder and new definitions)

An *event* is a subset of $AP$

A *trace* $\tau$ is an infinite sequence of events

$\tau[i]$ is the $i$-th event of a trace $\tau$ (event numbers start with 0)

$\tau^i$ is the suffix of a trace $\tau$ starting with the $i$-th event

A trace $\tau$ satisfies a formula $\varphi$ ($\tau \models \varphi$) in the following cases:

- $\tau \models a$ for $a \in AP \quad \Leftrightarrow \quad a \in tr[0]$
- $\tau \models \psi_1 \,\&\, \psi_2 \quad \Leftrightarrow \quad \tau \models \psi_1$ и $\tau \models \psi_2$
- $\tau \models \mathbf{X}\psi \quad \Leftrightarrow \quad \tau^1 \models \chi$
- $\tau \models \psi_1 \mathbf{U} \psi_2 \quad \Leftrightarrow$ there exists $k$, $k \geq 0$, such that:
  - $\tau^k \models \psi_2$
  - $\tau^m \models \psi_1$ for each $m$ such that $0 \leq m < k$

$Tr(\varphi)$ is the set of all traces $\tau$ such that $\tau \models \varphi$

(a *trace property* defined by a formula $\varphi$)

# LTL-formulae (reminder and new definitions)

**Examples** of ltl-formulae and (informally) corresonding properties:

1. The goal will be achieved eventually
$$\mathbf{F}\mathit{goal}$$
2. Autumn will eventually come, and it will be warm up until that time
$$\mathit{warm}\mathbf{U}\mathit{autumn}$$
3. Two bad days in a row are impossible
$$\mathbf{G}(\mathit{bad\_day} \rightarrow \mathbf{X}\neg\mathit{bad\_day})$$
4. If a request is being sent infinitely often,
then a response occurs at least once
$$\mathbf{GF}\mathit{request} \rightarrow \mathbf{F}\mathit{response}$$
5. An eternity in either Heaven or Hell awaits for me
$$\mathbf{F}(\mathbf{G}\mathit{heaven} \vee \mathbf{G}\mathit{hell})$$

# Kripke structures (reminder)

A *Kripke structure* is a tuple $M = (S, S_0, \rightarrow, L)$, where:

- $S$ is a finite set of states
- $S_0 \subseteq S$ is a set of initial states
- $\rightarrow \subseteq S \times S$ is a **total** transition relation
- $L : S \rightarrow 2^{AP}$ is a labeling function

Totalty of $\rightarrow$ means that for every state $s \in S$ there exists a state $s' \in S$ such that $R(s, s')$

A *path* $\pi$ in a model $M$ from a state $s$ is an infinite state sequence of the form

$$s \rightarrow s_1 \rightarrow s_2 \rightarrow \dots$$

A trace $\alpha(\pi)$ of the path $\pi$ is the sequence

$$L(s), L(s_1), L(s_2), \dots$$

$\Pi(M)$ is the set of all paths in $M$ from the initial states

$Tr(M) = \{\alpha(\pi) \mid \pi \in \Pi(M)\}$

# Model-checking problem for LTL (reminder)

Consider an ltl-formula $\varphi$ and a Kripke structure $M$

A path $\pi$ in $M$ *satisfies* $\varphi$ ($M, \pi \models \varphi$) iff

$$\alpha(\pi) \models \varphi$$

The model $M$ *satisfies* $\varphi$ ($M \models \varphi$) iff every path $\pi$ from $\Pi(M)$ satisfies $\varphi$, i.e.

$$Tr(M) \subseteq Tr(\varphi)$$

**Model checking problem for LTL**:

**given a Kripke structure $M$ and an ltl-formula $\varphi$**

**check whether the relation $M \models \varphi$ holds**

# Tableau-based model checking algorithm for LTL

Those who attended the bachelor course "Mathematical logic and logic programming" *should* already know this algorithm

**But...** it was long ago, and maybe you did not attend the course, or forgot it completely, or just skipped the hard lectures

In any case, some discussion on this topic is needed
(but **no proofs!** — to keep it short and simple)

## Starting point

*Without loss of generality* $\varphi$ is considered to contain **no double negations**: no subformulae of the form $\neg\neg\psi$ $\qquad (\neg\neg\psi \equiv \psi)$

Note that $M \not\models \varphi \Leftrightarrow$
there exists a path $\pi$ in $M$ from an initial state such that $M, \pi \not\models \varphi$

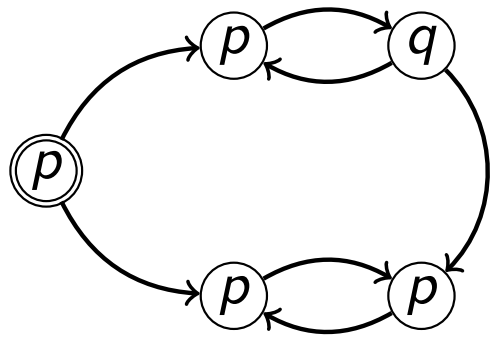
The algorithm ($\mathfrak{A}$) either finds such path $\pi$ (and $M \not\models \varphi$)
or states that there are no such paths (and $M \models \varphi$)

# Tableau-based model checking algorithm for LTL

## Brief algorithm description

## Illustration

$\varphi$: $p\mathbf{U}q$ $M$:



# Tableau-based model checking algorithm for LTL

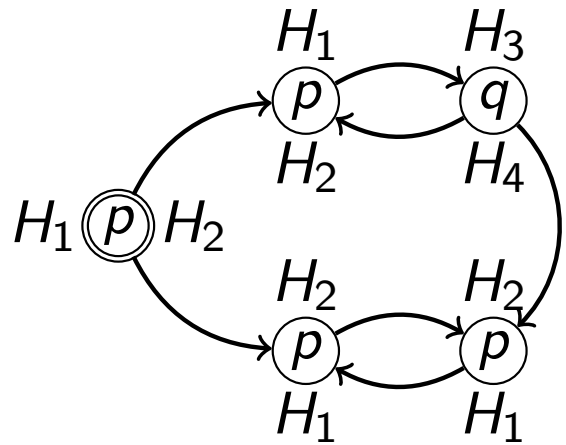
## Brief algorithm description

$\mathfrak{A}$ assigns to each state of $M$ a number of **conjectures** —
special finite sets of formulae, each of which contains on of the formulae

$$\varphi, \qquad \neg\varphi, \qquad \psi \text{ (where } \varphi = \neg\psi)$$

## Illustration

$\varphi$: $p\mathbf{U}q$




$$H_1 = \{p, \neg q, \quad \mathbf{X}(p\mathbf{U}q), \quad p\mathbf{U}q\ \} \qquad H_3 = \{\neg p, q, \quad \mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$$
$$H_2 = \{p, \neg q, \neg\mathbf{X}(p\mathbf{U}q), \neg(p\mathbf{U}q)\} \qquad H_4 = \{\neg p, q, \neg\mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$$

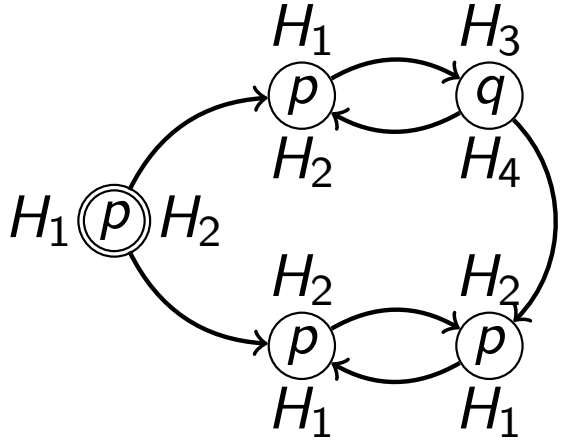
# Tableau-based model checking algorithm for LTL

## Brief algorithm description

Informally, the meaning of a conjecture $H$ assigned to $s$ is:
*it is possible that* there exists a path $\pi$ in $M$ from $s$
which satisfies every formula of $H$ $(\alpha(\pi) \models H)$

## Illustration

$\varphi$: $p\mathbf{U}q$




$$H_1 = \{p, \neg q, \quad \mathbf{X}(p\mathbf{U}q), \quad p\mathbf{U}q \ \} \qquad H_3 = \{\neg p, q, \quad \mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$$
$$H_2 = \{p, \neg q, \neg\mathbf{X}(p\mathbf{U}q), \neg(p\mathbf{U}q)\} \qquad H_4 = \{\neg p, q, \neg\mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$$

# Tableau-based model checking algorithm for LTL

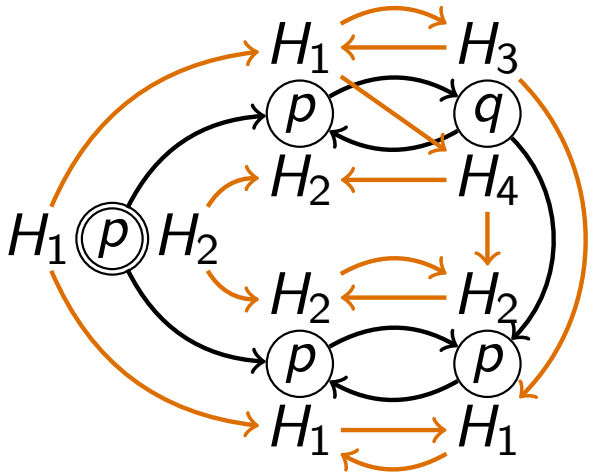
## Brief algorithm description

$\mathfrak{A}$ connects some pairs (state, conjecture) with arks

Informally, the meaning of an ark $(s_1, H_1) \rightarrow (s_2, H_2)$ is:
*it is possible that* there exists a path $\pi = (s_1 \rightarrow s_2 \rightarrow \dots)$ in $M$ such that $\alpha(\pi) \models H_1$ and $\alpha(\pi^1) \models H_2$

## Illustration

$\varphi$: $p\mathbf{U}q$




$$H_1 = \{p, \neg q, \quad \mathbf{X}(p\mathbf{U}q), \quad p\mathbf{U}q \ \} \qquad H_3 = \{\neg p, q, \quad \mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$$
$$H_2 = \{p, \neg q, \neg\mathbf{X}(p\mathbf{U}q), \neg(p\mathbf{U}q)\} \qquad H_4 = \{\neg p, q, \neg\mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$$

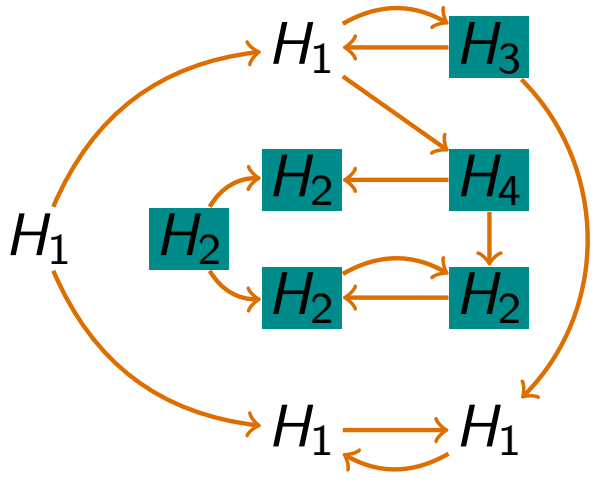
# Tableau-based model checking algorithm for LTL

## Brief algorithm description

The resulting graph — a **Hintikka structure** — is marked in a special way which allows to check for any state $s$ whether there exists a path $\widetilde{\pi}$ in $M$ from $s$ such that the mentioned satisfiability relations are not only "possible", but actually hold

## Illustration

$\varphi$: $p\mathbf{U}q$




$$H_1 = \{p, \neg q, \quad \mathbf{X}(p\mathbf{U}q), \quad p\mathbf{U}q \ \} \qquad H_3 = \{\neg p, q, \quad \mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$$
$$H_2 = \{p, \neg q, \neg\mathbf{X}(p\mathbf{U}q), \neg(p\mathbf{U}q)\} \qquad H_4 = \{\neg p, q, \neg\mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$$

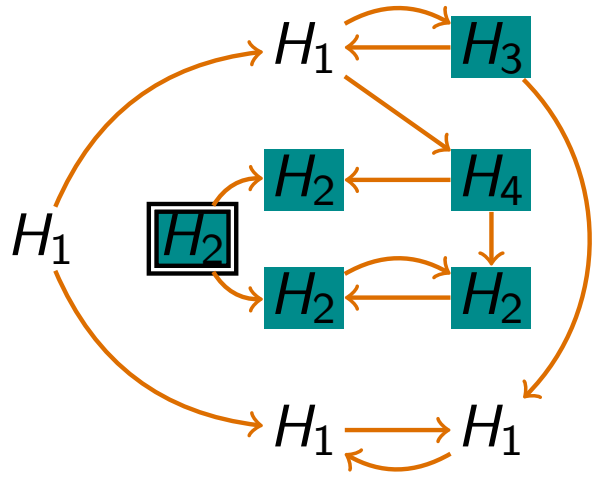
# Tableau-based model checking algorithm for LTL

## Brief algorithm description

$M \not\models \varphi \Leftrightarrow$ there exists at least one mentioned path $\widetilde{\pi}$ from a vertex $(s, H)$ where $s$ is an initial state and $\varphi \notin H$

## Illustration

$\varphi$: $p\mathbf{U}q$




$$H_1 = \{p, \neg q, \quad \mathbf{X}(p\mathbf{U}q), \quad p\mathbf{U}q \ \} \qquad H_3 = \{\neg p, q, \quad \mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$$
$$H_2 = \{p, \neg q, \neg\mathbf{X}(p\mathbf{U}q), \neg(p\mathbf{U}q)\} \qquad H_4 = \{\neg p, q, \neg\mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$$

# Tableau-based model checking algorithm for LTL

A formula $\varphi$ is called negative, if $\varphi = \neg\psi$, and positive otherwise

A Fischer-Ladner closure $[\varphi]_{fl}$ of a formula $\varphi$ is the set of formulae consisting of:

1. all positive subformulae of $\varphi$
2. $\mathbf{X}(\psi\mathbf{U}\chi)$ for each subformula $\psi\mathbf{U}\chi$ of $\varphi$

**Example:** $[\neg(p\mathbf{U}\neg q)]_{fl} = \{p, q, p\mathbf{U}\neg q, \mathbf{X}(p\mathbf{U}\neg q)\}$

A conjecture (for $\varphi$) is a set of formulae of the following form:

$$F \cup \{\neg\psi \mid \chi \in [\varphi]_{fl} \setminus F\},$$

where $F \subseteq [\varphi]_{fl}$

**Example:** $\{\neg p, \neg q, p\mathbf{U}\neg q, \neg\mathbf{X}(p\mathbf{U}\neg q)\}$ is a conjecture for $\neg(p\mathbf{U}\neg q)$

# Tableau-based model checking algorithm for LTL

A conjecture $H$ is internally consistent iff
for all formulae $\psi_1 \,\&\, \psi_2$ and $\chi_1 \mathbf{U} \chi_2$ from $[\varphi]_{fl}$ the following holds:

1. $\psi_1 \,\&\, \psi_2 \in H \quad \Leftrightarrow \quad \{\psi_1, \psi_2\} \subseteq H$
2. $\chi_1 \mathbf{U} \chi_2 \in H \quad \Leftrightarrow \quad \chi_2 \in H$ or $\{\chi_1, \mathbf{X}(\chi_1 \mathbf{U} \chi_2)\} \subseteq H$

*Explanation:* (2) represents the following "fixed-point" equivalence:

$$\chi_1 \mathbf{U} \chi_2 \equiv \chi_2 \vee \chi_1 \,\&\, \mathbf{X}(\chi_1 \mathbf{U} \chi_2)$$

**Examples:**

- $\{\neg p, q, \neg \mathbf{X}(p \mathbf{U} q), \quad p \mathbf{U} q\ \}$ is internally consistent
- $\{\neg p, q, \quad \mathbf{X}(p \mathbf{U} q), \neg(p \mathbf{U} q)\}$ is not internally consistent

$[\varphi]^{-}_{fl} = \{\neg\psi \mid \psi \in [\varphi]_{fl}\}$

$H^i_\tau$ is the set of all formulae $\psi$ from $[\varphi]_{fl} \cup [\varphi]^{-}_{fl}$ such that $\tau^i \models \psi$
($\tau$ *is a trace,* $i \geq 0$)

**Proposition.** **For any trace $\tau$ and any index $i$, $i \geq 0$, the set $H^i_\tau$ is an internally consistent conjecture**

# Tableau-based model checking algorithm for LTL

Conjectures $H_1$, $H_2$ are locally consistent iff for any formula $\mathbf{X}\chi$ from $[\varphi]_{fl}$ the following holds:

$$\mathbf{X}\chi \in H_1 \Leftrightarrow \chi \in H_2$$

**Proposition.** **For any trace $\tau$ and any index $i$, $i \geq 0$, the conjectures $H_\tau^i$ and $H_\tau^{i+1}$ are locally consistent**

A conjecture $H$ concludes a formula $\psi\mathbf{U}\chi$ from $[\varphi]_{fl}$ iff at least one of the following conditions hold:

1. $\chi \in H$
2. $\mathbf{X}(\psi\mathbf{U}\chi) \notin H$

An infinite sequence $\mathfrak{H} = (H_0, H_1, \dots)$ of conjectures is globally consistent iff for any formula $\psi\mathbf{U}\chi$ from $[\varphi]_{fl}$ infinitely many conjectures from $\mathfrak{H}$ conclude the formula $\psi\mathbf{U}\chi$

**Proposition.** **For any trace $\tau$ the sequence $H_\tau^0, H_\tau^1, H_\tau^2, \dots$ is globally consistent**

# Tableau-based model checking algorithm for LTL

For an infinite sequence $\mathfrak{H} = H_0, H_1, H_2, \ldots$ of conjectures, $\tau_{\mathfrak{H}}$ is the trace $H_0 \cap AP, H_1 \cap AP, H_2 \cap AP, \ldots$

**Proposition.** **For any infinite sequence $\mathfrak{H} = (H_0, H_1, H_2, \ldots)$ of conjectures the following holds: if**

- **each conjecture $H_j$ is internally consistent,**
- **each pair $(H_j, H_{j+1})$ is locally consistent, and**
- **$\mathfrak{H}$ is globally consistent,**

**then for each index $i$, $i \geq 0$, the equality $H_i = H^i_{\tau_{\mathfrak{H}}}$ holds**

**Note on globale consistency:**

- $H = \{p, \neg q, \mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$ is an internally consistent conjecture
- $H$, $H$ are locally consistent conjectures
- $H, H, H, \ldots$ is **not** a globally consistent sequence of conjectures
- $(\{p\}, \{p\}, \{p\}, \ldots) \not\models p\mathbf{U}q$

# Tableau-based model checking algorithm for LTL

A conjecture $H$ is an $s$-conjecture for a state $s$ of a Kripke structure $M = (S, S_0, \rightarrow, L)$ iff $L(s) = H \cap AP$

A Hintikka structure $\mathcal{HS}(M, \varphi)$ for $M$ and a formula $\varphi$ is the following labeled directed graph:

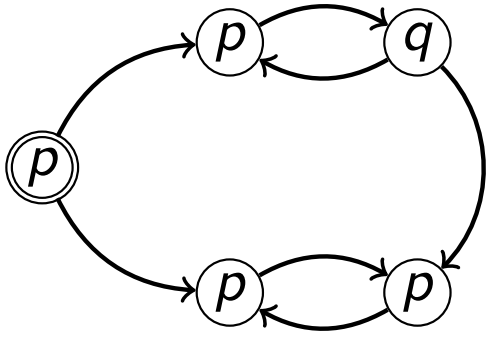
- vertices are all pairs $(s, H)$ such that
  $s \in S$, $H$ is an internally consistent $s$-conjecture
- an arc $(s_1, H_1) \rightarrow (s_2, H_2)$ belongs to the structure $\Leftrightarrow$
  $s_1 \rightarrow s_2$, and $H_1$, $H_2$ are locally consistent conjectures
- a vertex $(s, H)$ is initial $\Leftrightarrow$ $s \in S_0$, and $\varphi \notin H$
- each formula $\psi \mathbf{U} \chi$ from $[\varphi]_{fl}$ corresponds to a unique color $c_{\psi \mathbf{U} \chi}$
- a vertex $(s, H)$ is marked with $c_\Phi \Leftrightarrow H$ concludes $\Phi$

# Tableau-based model checking algorithm for LTL

## Example

$\varphi$: $p\mathbf{U}q$

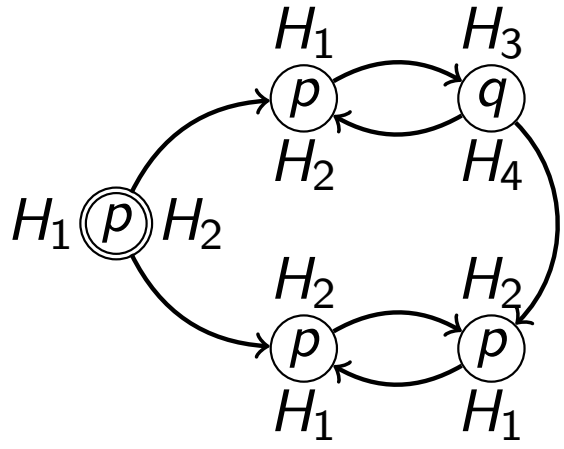
$M$:



# Tableau-based model checking algorithm for LTL

## Example

$\varphi: p\mathbf{U}q$




$H_1 = \{p, \neg q, \quad \mathbf{X}(p\mathbf{U}q), \quad p\mathbf{U}q \ \}$ $\quad H_3 = \{\neg p, q, \quad \mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$

$H_2 = \{p, \neg q, \neg\mathbf{X}(p\mathbf{U}q), \neg(p\mathbf{U}q)\}$ $\quad H_4 = \{\neg p, q, \neg\mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$

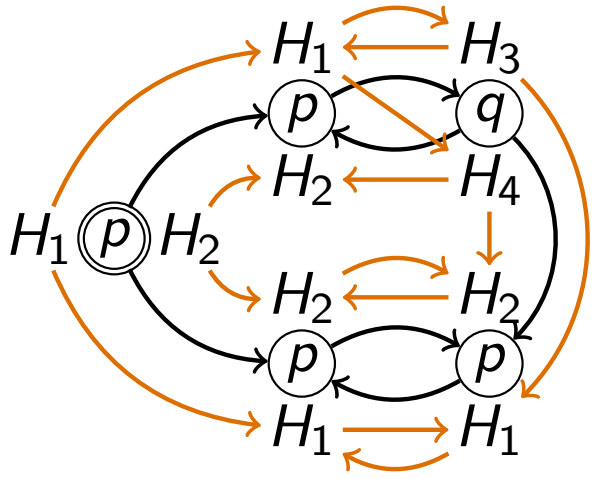
$H_1$ and $H_2$ are all internally consistent $(p)$-conjectures

$H_3$ and $H_4$ are all internally consistent $(q)$-conjectures

# Tableau-based model checking algorithm for LTL

## Example

$\varphi: p\mathbf{U}q$




$H_1 = \{p, \neg q, \quad \mathbf{X}(p\mathbf{U}q), \quad p\mathbf{U}q\ \}$ $\quad H_3 = \{\neg p, q, \quad \mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$

$H_2 = \{p, \neg q, \neg\mathbf{X}(p\mathbf{U}q), \neg(p\mathbf{U}q)\}$ $\quad H_4 = \{\neg p, q, \neg\mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$

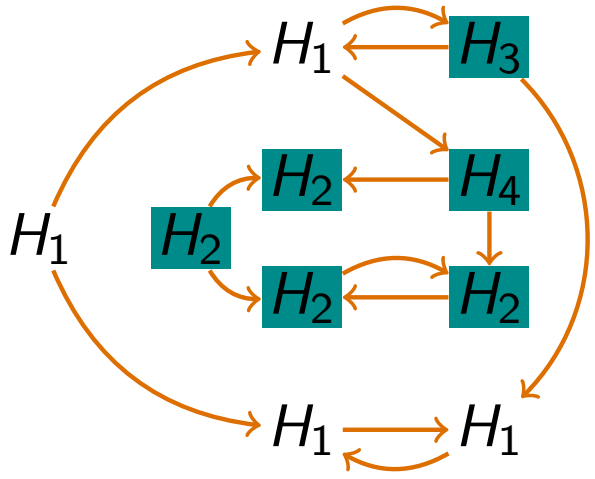
Local consistency of conjectures $H_i$, $H_j$:

$$\mathbf{X}(p\mathbf{U}q) \in H_i \Leftrightarrow p\mathbf{U}q \in H_j$$

# Tableau-based model checking algorithm for LTL

## Example

$\varphi: p\mathbf{U}q$




$H_1 = \{p, \neg q, \quad \mathbf{X}(p\mathbf{U}q), \quad p\mathbf{U}q\ \}$ $\quad H_3 = \{\neg p, q, \quad \mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$

$H_2 = \{p, \neg q, \neg\mathbf{X}(p\mathbf{U}q), \neg(p\mathbf{U}q)\}$ $\quad H_4 = \{\neg p, q, \neg\mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$

$H_3$ and $H_4$ conclude $p\mathbf{U}q$: $q \in H_3 \cap H_4$

$H_2$ concludes $p\mathbf{U}q$: $\mathbf{X}(p\mathbf{U}q) \notin H_2$

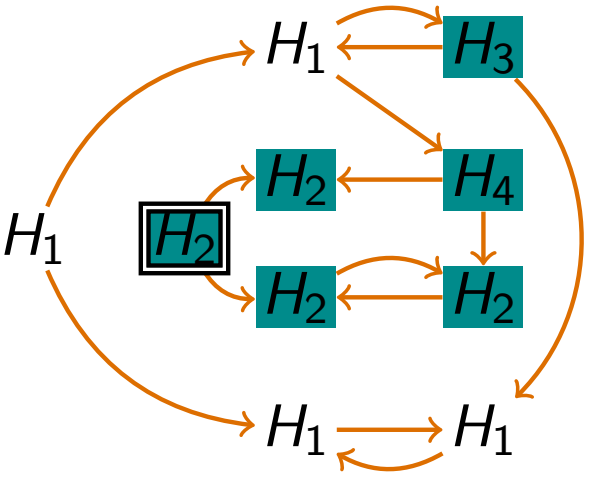
$H_1$ **does not** conclude $p\mathbf{U}q$:

- $q \notin H_1$
- $\mathbf{X}(p\mathbf{U}q) \in H_1$

# Tableau-based model checking algorithm for LTL

## Example

$\varphi$: $p\mathbf{U}q$ $\quad$ $\mathcal{HS}(M,\varphi)$:




$H_1 = \{p, \neg q, \quad \mathbf{X}(p\mathbf{U}q), \quad p\mathbf{U}q\ \}$ $\quad$ $H_3 = \{\neg p, q, \quad \mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$

$H_2 = \{p, \neg q, \neg\mathbf{X}(p\mathbf{U}q), \neg(p\mathbf{U}q)\}$ $\quad$ $H_4 = \{\neg p, q, \neg\mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$

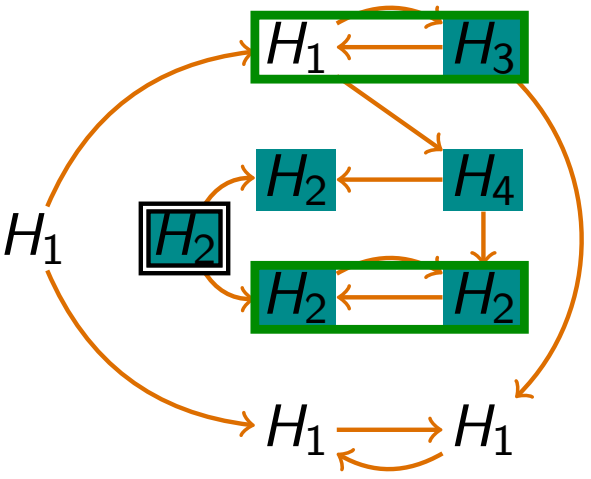
The only initial vertex $(s, H)$:

- $s$ is the initial state of $M$
- $p\mathbf{U}q \notin H$

# Tableau-based model checking algorithm for LTL

## Example

$\varphi: p\mathbf{U}q$ $\quad$ $\mathcal{HS}(M,\varphi)$:




$H_1 = \{p, \neg q, \quad \mathbf{X}(p\mathbf{U}q), \quad p\mathbf{U}q\ \}$ $\quad$ $H_3 = \{\neg p, q, \quad \mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$

$H_2 = \{p, \neg q, \neg\mathbf{X}(p\mathbf{U}q), \neg(p\mathbf{U}q)\}$ $\quad$ $H_4 = \{\neg p, q, \neg\mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$

**Theorem.** $M \not\models \varphi \Leftrightarrow$

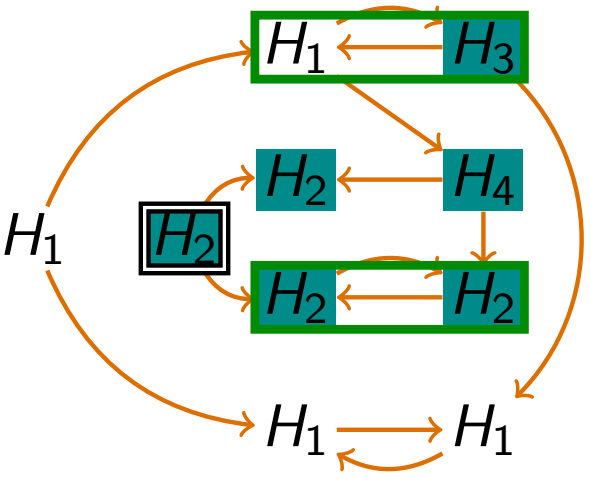
**$\mathcal{H}(M,\varphi)$ contains a path $\pi$ from an initial vertex such that each color occurs inifinitely often in $\pi$ $\Leftrightarrow$**

**at least one nontrivial (with at least one arc) strongly connected component containing all colors is reachable from an initial vertex in $\mathcal{H}(M,\varphi)$**

# Tableau-based model checking algorithm for LTL

## Example

$\varphi: p\mathbf{U}q$ $\quad$ $\mathcal{HS}(M,\varphi)$:




$H_1 = \{p, \neg q, \quad \mathbf{X}(p\mathbf{U}q), \quad p\mathbf{U}q\ \}$ $\quad$ $H_3 = \{\neg p, q, \quad \mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$

$H_2 = \{p, \neg q, \neg\mathbf{X}(p\mathbf{U}q), \neg(p\mathbf{U}q)\}$ $\quad$ $H_4 = \{\neg p, q, \neg\mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$

**Conclusion:** $M \not\models \varphi$

# Interlude

*Tableau-based model checking algorithm for CTL* is inefficient, but ideologically underlies other efficient algorithms (e.g. the *symbolic* algorithm)

The same holds for LTL: tableau-based model checking algorithm for LTL is inefficient, but ideologically underlies other efficient algorithms

The most popular efficient model checking algorithms for LTL used in practice are automata-based

# Automata-based algorithm: general scheme

**Given:** a Kripke structure $M$, an ltl-formula $\varphi$

**Check:** $M \models \varphi$?

**General scheme:**

1. Construct an automaton $A_M$ which recognizes the set of (**inifinite**!) traces $Tr(M)$ $\qquad (L(A_M) = Tr(M))$
2. Construct an automaton $A_{\neg\varphi}$ which recognizes the set of traces $Tr(\neg\varphi)$ $\qquad (L(A_{\neg\varphi}) = Tr(\neg\varphi))$
3. Combine $A_M$ and $A_{\neg\varphi}$ into an automaton $A$ recognizing the set $Tr(M) \cap Tr(\neg\varphi)$ $\qquad (L(A) = L(A_M) \cap L(A_{\neg\varphi}))$
4. Check whether $A$ recognizes the **empty** set of traces: $\qquad (L(A) = \emptyset)$
   - if $L(A) = \emptyset$, then $M \models \varphi$
   - if $L(A) \neq \emptyset$, then $M \not\models \varphi$

# Büchi automata

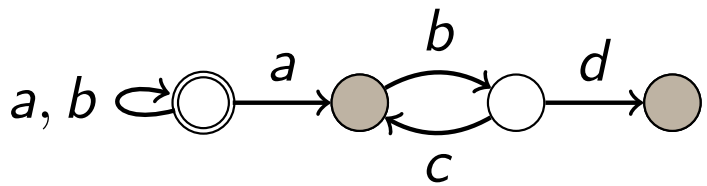
A Büchi automaton (BA) over an alphabet $\Sigma$ is a tuple $A = (S, S_0, \rightarrow, F)$, where:

- $S$ is a finite set of states
- $S_0 \subseteq S$ is a set of initial states
- $\rightarrow \subseteq S \times \Sigma \times S$ is a transition relation
- $F \subseteq S$ is a set of accepting states

$s \xrightarrow{\sigma_1,\ldots,\sigma_k} s'$ is a graph-related representation of a transition

$$(s, \sigma_1, s'), \quad \ldots, \quad (s, \sigma_k, s')$$

**Example**




◎ is an initial state

● is an accepting state

# Büchi automata

An $\omega$-word is an infinite word

A run of a BA $A = (S, S_0, \rightarrow, F)$ (on an $\omega$-word $\sigma_1\sigma_2\sigma_3\ldots$) is an infinite sequence of states of the form

$$s_0 \xrightarrow{\sigma_1} s_1 \xrightarrow{\sigma_2} s_2 \xrightarrow{\sigma_3} \ldots$$

such that $s_0 \in S_0$

$R(A, w)$ is the set of all runs of $A$ on an $\omega$-word $w$

$inf(\rho)$ is the set of all states occuring infinitely often in a run $\rho$

A run $\rho$ of $A$ is accepting iff $inf(\rho) \cap F \neq \emptyset$

The BA $A$ accepts an $\omega$-word $w$ iff there exists at least one accepting run of $A$ on $w$, i.e.:

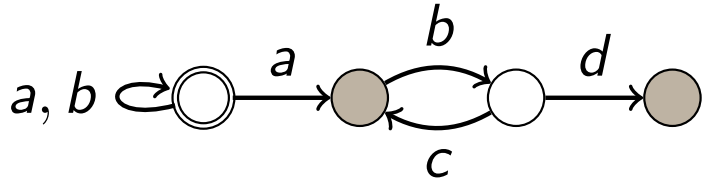
$$\exists \rho \in R(A, w) : inf(\rho) \cap F \neq \emptyset$$

$L(A)$ is the language accepted by $A$: the set of all $\omega$-words accepted by $A$

# Büchi automata

## Example

Consider the following BA $A$:




Then

$$L(A) = \{habcbcbc \dots bc \cdots \mid h \in \{a, b\}^*\}$$

# Büchi automata and Kripke structures

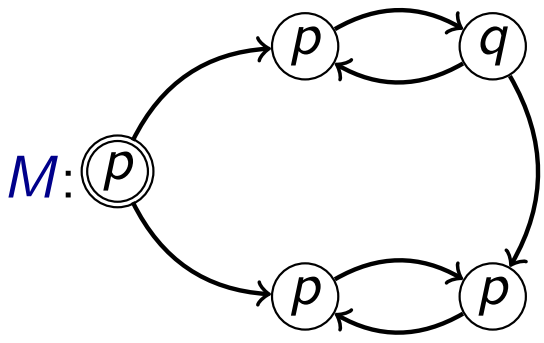
**Given:** a Kripke structure $M = (S, S_0, \to, L)$

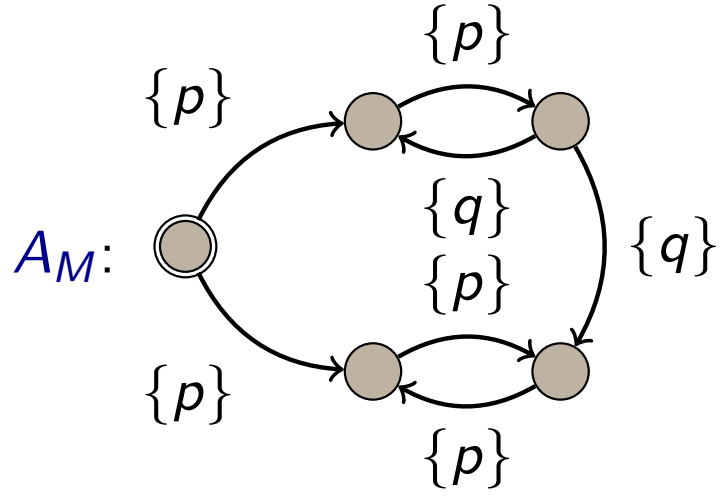
**Compute:** a BA $A_M = (S', S'_0, \mapsto, F)$ over $2^{AP}$ such that $L(A_M) = Tr(M)$

**Solution:**

- $S' = F = S$
- $S'_0 = S_0$
- $s_1 \overset{\sigma}{\mapsto} s_2 \Leftrightarrow s_1 \to s_2$ и $\sigma = L(s_1)$

**Example**



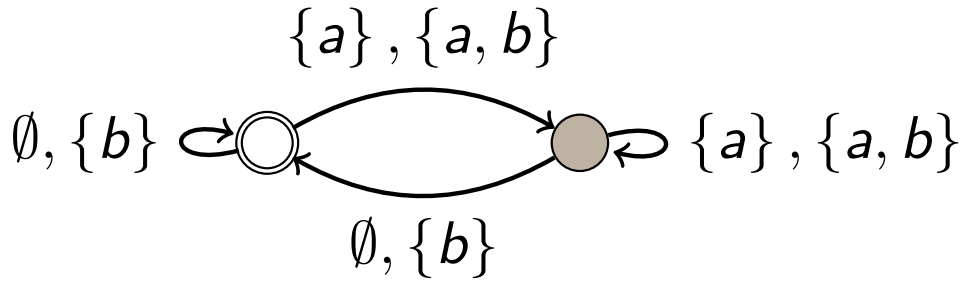



# Büchi automata and ltl-formulae

**Given:** an ltl-formula $\varphi$

**Compute:** a BA $A_\varphi = (S, S_0, \to, F)$ over $2^{AP}$ such that $L(A_\varphi) = Tr(\varphi)$

**Let us start with some examples** $(AP = \{a, b\})$



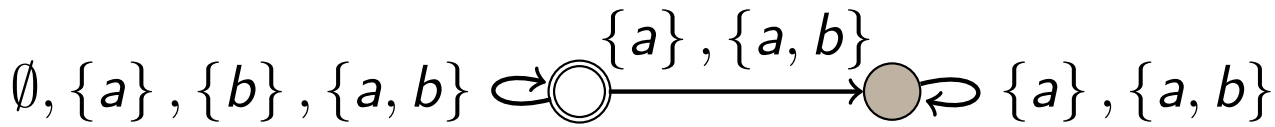

$$L(A) = Tr(\mathbf{GF}a)$$

# Büchi automata and ltl-formulae

**Given:** an ltl-formula $\varphi$

**Compute:** a BA $A_\varphi = (S, S_0, \to, F)$ over $2^{AP}$ such that $L(A_\varphi) = Tr(\varphi)$

**Let us start with some examples** $(AP = \{a, b\})$



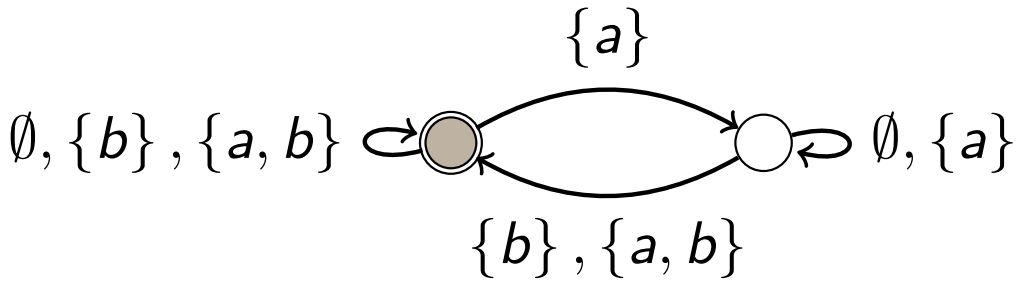

$$L(A) = Tr(\mathbf{FG}a)$$

# Büchi automata and ltl-formulae

**Given:** an ltl-formula $\varphi$

**Compute:** a BA $A_\varphi = (S, S_0, \to, F)$ over $2^{AP}$ such that $L(A_\varphi) = Tr(\varphi)$

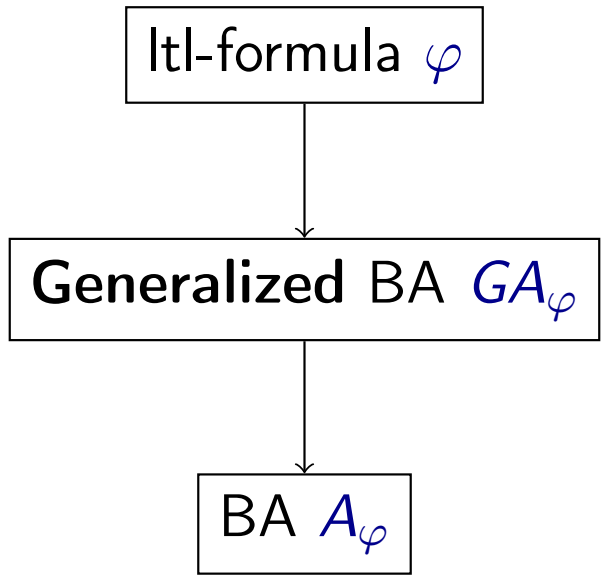
**Let us start with some examples** $(AP = \{a, b\})$




$$L(A) = Tr(\mathbf{G}(a \to \mathbf{F}b))$$

# Büchi automata and ltl-formulae

General scheme of a translation of an ltl-formula into a BA:



# Generalized Büchi automata

A Generalized Büchi automaton (GBA) over an alphabet $\Sigma$ is a tuple $GA = (S, S_0, \rightarrow, \mathcal{F})$, where:

- $S$ is a finite set of states
- $S_0 \subseteq S$ is a set of initial states
- $\rightarrow \subseteq S \times \Sigma \times S$ is a transition relation
- $\mathcal{F} \subseteq 2^S$ is a collection of accepting sets

The only basic notion for GBA different from the same notion for BA is the definition of an *accepting run*

A run $\rho$ of a GBA $GA$ is accepting iff for each accepting set $F$ of $GA$ the inequality $inf(\rho) \cap F \neq \emptyset$ holds

Corresponding acceptance condition for $GA$ can be stated as follows:

$$\exists \rho \in R(GA, w) : \forall F \in \mathcal{F} : inf(\rho) \cap F \neq \emptyset$$

# Generalized Büchi automata

**Theorem**

**For any GBA $GA$ there exists a BA $A$ such that $L(A) = L(GA)$**

Proof.

Let us assume for clarity that $GA = (S, S_0, \to, \mathcal{F})$, where

$$\mathcal{F} = \{F_0, \ldots, F_{k-1}\}$$

Consider the following BA $A = (S', S'_0, \mapsto, F)$:

- $S' = \{(s, i) \mid s \in S; i \in \{0, \ldots, k-1\}\}$
- $S'_0 = \{(s, 0) \mid s \in S_0\}$
- $F = \{(s, i) \mid s \in F_i; i \in \{0, \ldots, k-1\}\}$
- if $s \notin F_i$, then $(s, i) \overset{\delta}{\mapsto} (s', i) \Leftrightarrow s \overset{\delta}{\to} s'$
- if $s \in F_i$, then $(s, i) \overset{\delta}{\mapsto} (s', (i+1) \bmod k) \Leftrightarrow s \overset{\delta}{\to} s'$

It is sufficient to show that $L(GA) = L(A)$

# Generalized Büchi automata

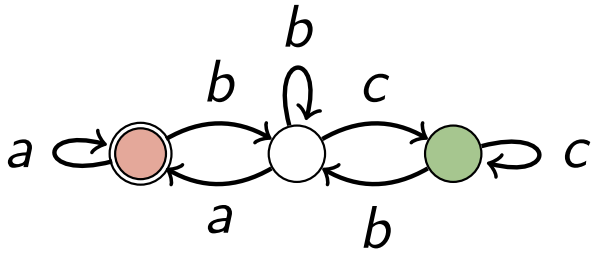
**Theorem**

**For any GBA $GA$ there exists a BA $A$ such that $L(A) = L(GA)$**

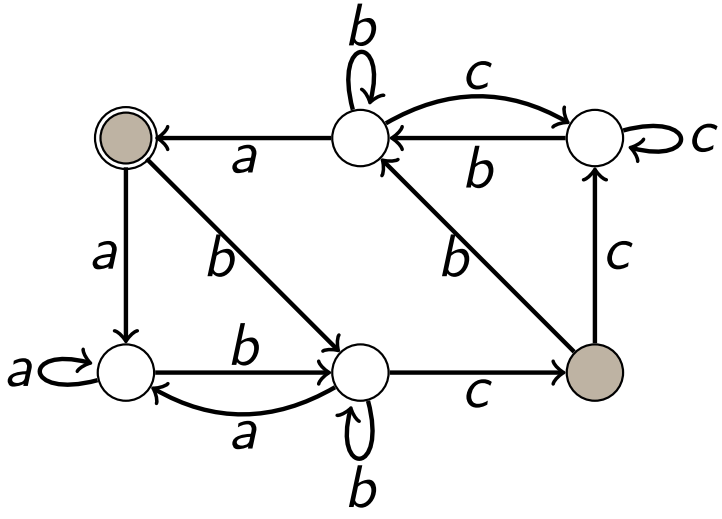
Proof.

**Illustration:**

$GA$:




$A$:



# Generalized Büchi automata

**Theorem**

**For any GBA $GA$ there exists a BA $A$ such that $L(A) = L(GA)$**

Proof. $(L(A) \subseteq L(GA))$

Consider an $\omega$-word $\delta_0\delta_1\ldots$ accepted by $A$, and an accepting run $\rho$ of $A$ on this word: $(s_0, i_0) \overset{\delta_0}{\longmapsto} (s_1, i_1) \overset{\delta_1}{\longmapsto} \ldots$

Then the sequence $\rho$ contains an infinite subsequence

$$(q_0, 0), (q_1, 1), \ldots, (q_{k-1}, k-1), (q_k, 0), (q_{k+1}, 1), \ldots,$$

such that $(q_i, i \bmod k) \in F$, $i \geq 0$

By defition of $A$:

- $\rho' = (s_0 \overset{\delta_0}{\longrightarrow} s_1 \overset{\delta_1}{\longrightarrow} \ldots)$ is a run of $GA$
- $q_i \in F_{i \bmod k}$, which means that for each accepting set $F_j$ of $GA$ at least one state of $F_j$ occurs in $\rho'$ infinitely often

Thus, $\rho'$ is an accepting run, and the word $\delta_0\delta_1\ldots$ is accepted by $GA$

# Generalized Büchi automata

**Theorem**

**For any GBA $GA$ there exists a BA $A$ such that $L(A) = L(GA)$**

Proof. $(L(GA) \subseteq L(A))$

Consider an $\omega$-word $\delta_0\delta_1\ldots$ accepted by $GA$, an accepting run $\rho$ of $GA$ on this word: $s_0 \xrightarrow{\delta_0} s_1 \xrightarrow{\delta_1} \ldots$

Consider the following subsequence $q_0, q_1, q_2, \ldots$ of $\rho$:

- $q_0$ is the first state of $\tau$ such that $q_0 \in F_0$
- $q_{i+1}$ is the first state of $\tau$ following $q_i$ such that $q_{i+1} \in F_{i+1 \bmod k}$, $i \geq 0$

By definition of $A$, the following run is an accepting run of $A$ on $\delta_0\delta_1\ldots$:

$$(s_0, 0) \mapsto \cdots \mapsto (q_0, 0) \mapsto (s_{i_1}, 1) \mapsto \cdots \mapsto (q_1, 1) \mapsto \cdots \mapsto (s_{i_2}, 2) \mapsto \cdots \mapsto \cdots \mapsto (q_{k-1}, k-1) \mapsto (s_{i_k}, 0) \mapsto \cdots \mapsto (q_k, 0) \mapsto (s_{k+1}, 1) \mapsto \ldots$$

# Generalized Büchi automata and ltl-formulae

**Given:** an ltl-formula $\varphi$

**Construct:** a GBA $GA_\varphi = (S, S_0, \to, \mathcal{F})$ over $2^{AP}$ such that $L(GA_\varphi) = Tr(\varphi)$

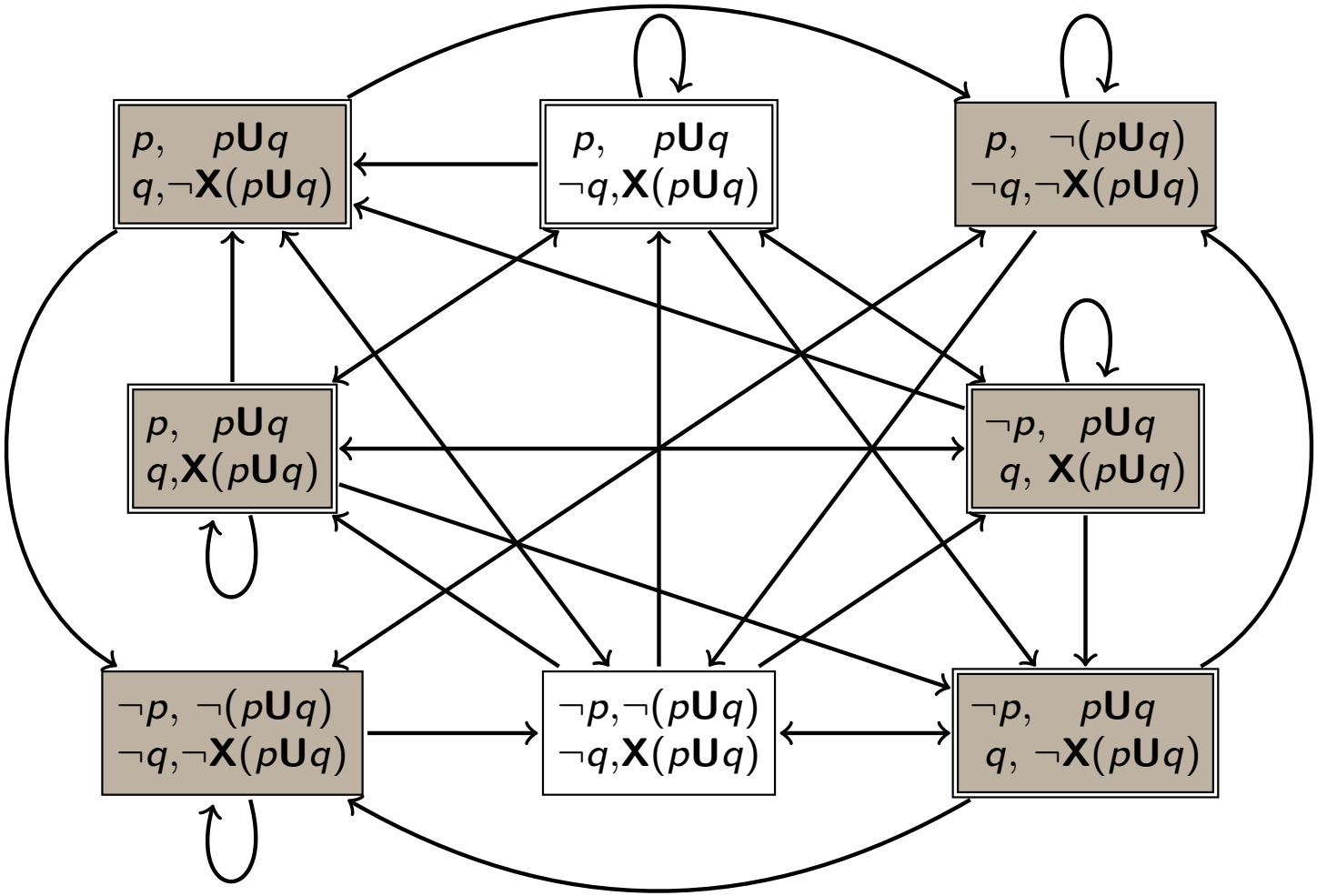
**Solution (*not the best one, but the simplest one*):**

- $S$ is the set of all internally consistent conjectures for $\varphi$
- $H \in S_0 \Leftrightarrow \varphi \in H$
- $H_1 \xrightarrow{X} H_2 \Leftrightarrow X = H_1 \cap AP$, and the conjectures $H_1$, $H_2$ are locally consistent
- $\mathcal{F} = \{F_1, \ldots, F_k\}$, where
  - $\psi_1, \ldots, \psi_k$ are all subformulae of $\varphi$ of the form $\chi_1 \mathbf{U} \chi_2$ (arbitrarily numbered)
  - $H \in F_i \Leftrightarrow$ the conjecture $H$ concludes $\psi_i$

The proposed GBA can be alternatively defined as an arc-labeled Hintikka system for $\neg\varphi$ and a smallest Kripke model containing all possible traces ($Tr(M') = (2^{2^{AP}})^\omega$), and correctness of the GBA follows from Hintikka system properties

# Generalized Büchi automata and ltl-formulae

**Example:** $GA_{p\mathbf{U}q}$




(*arc labels are omitted: an arc label equals to a state label for the source state of the arc*)

# Büchi automata intersection

## Theorem

**For any BA $A'$, $A''$ there exists a GBA $GA$ such that $L(GA) = L(A') \cap L(A'')$**

Proof.

Let us assume for clarity that $A' = (S', S'_0, \to, F')$ и $A'' = (S'', S''_0, \mapsto, F'')$

Consider the following GBA $GA = (S, S_0, \Rightarrow, \mathcal{F})$
(*a Cartesian product of $A'$ and $A''$*):

- $S = \{(s', s'') \mid s' \in S'; s'' \in S''\}$
- $S_0 = \{(s', s'') \mid s' \in S'_0; s'' \in S''_0\}$
- $(s'_1, s''_1) \stackrel{\delta}{\Rightarrow} (s'_2, s''_2) \Leftrightarrow s'_1 \stackrel{\delta}{\to} s'_2$ and $s''_1 \stackrel{\delta}{\mapsto} s''_2$
- $\mathcal{F} = \{F_1, F_2\}$, where
  $F_1 = \{(s', s'') \mid s' \in F'; s'' \in S''\}$ and
  $F_2 = \{(s', s'') \mid s' \in S'; s'' \in F''\}$

It is sufficient to show that $L(GA) = L(A') \cap L(A'')$

# Büchi automata intersection

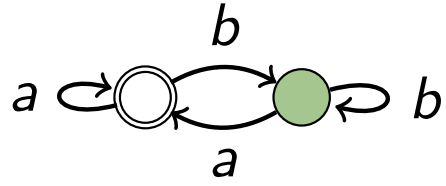
## Theorem

**For any BA $A'$, $A''$ there exists a GBA $GA$ such that $L(GA) = L(A') \cap L(A'')$**

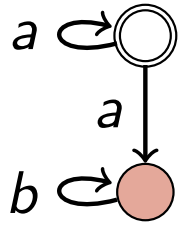
Proof.

## Illustration

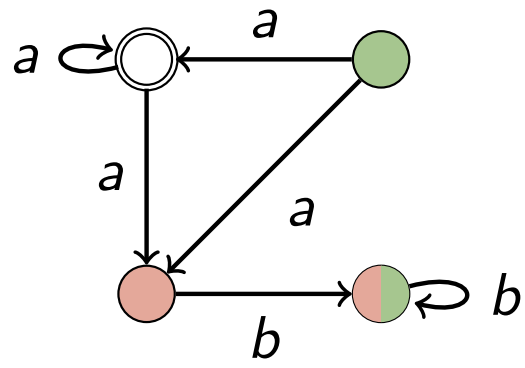
$A'$:




$A''$:




$GA$:



# Büchi automata intersection

**Theorem**

**For any BA $A'$, $A''$ there exists a GBA $GA$ such that $L(GA) = L(A') \cap L(A'')$**

Proof. $(L(GA) \subseteq L(A') \cap L(A''))$

Consider an accepting run $\rho$ of $GA$:

$$(s'_0, s''_0) \overset{\delta_0}{\Rightarrow} (s'_1, s''_1) \overset{\delta_1}{\Rightarrow} \dots$$

By definition of $GA$:

- $\rho' = (s'_0 \overset{\delta_0}{\longrightarrow} s'_1 \overset{\delta_1}{\longrightarrow} \dots)$ is a run of $A'$
- $\rho'' = (s''_0 \overset{\delta_0}{\longmapsto} s''_1 \overset{\delta_1}{\longmapsto} \dots)$ is a run of $A''$
- states from $F_1$ occur infinitely often in $\rho$, and, therefore, in $\rho'$
  - thus, $\rho'$ is an accepting run of $A_1$
- states from $F_1$ occur infinitely often in $\rho$, and, therefore, in $\rho''$
  - thus, $\rho''$ is an accepting run of $A_2$

# Büchi automata intersection

**Theorem**

**For any BA $A'$, $A''$ there exists a GBA $GA$ such that $L(GA) = L(A') \cap L(A'')$**

Proof. $(L(A') \cap L(A'') \subseteq L(GA))$

Consider accepting runs $\rho'$, $\rho''$ of $A'$, $A''$ respectively:

$$s'_0 \xrightarrow{\delta_0} s'_1 \xrightarrow{\delta_1} \dots$$
$$s''_0 \xmapsto{\delta_0} s''_1 \xmapsto{\delta_1} \dots$$

By definition of $GA$:

- $\rho = ((s'_0, s''_0) \overset{\delta_0}{\Rightarrow} (s'_1, s''_1) \overset{\delta_1}{\Rightarrow} \dots)$ is a run of $GA$
- states of в $F_1$ occur infinitely often in $\rho'$, and, therefore, in $\rho$
- states of в $F_2$ occur infinitely often in $\rho''$, and, therefore, in $\rho$

Thus, $\rho$ is an accepting run of $GA$ ▼

# Emptiness checking for Büchi automata

**Theorem**

**For any BA $A$:**

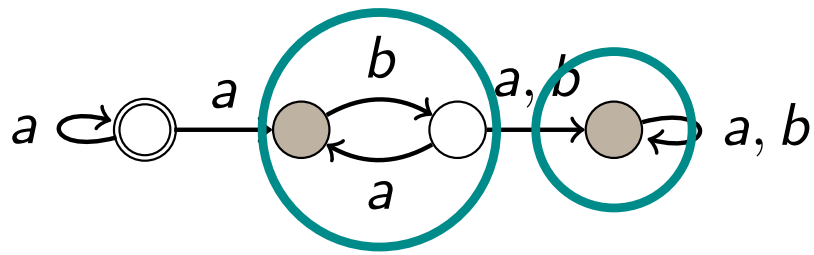
$L(A) = \emptyset \Leftrightarrow$

**at least one nontrivial strongly connected component containing an accepting state is reachable from an initial state**

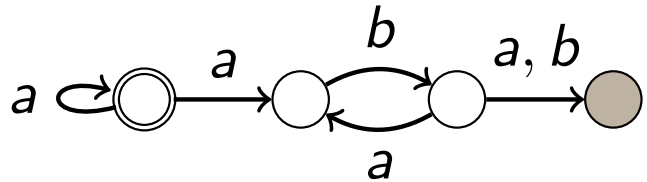
Proof.Obvious?

**Example**

$A_1$:




$A_2$:




$L(A_1) \neq \emptyset$

$L(A_2) = \emptyset$

# Automata-based model checking algorithm for LTL

**Given:** a Kripke structure $M$, and an ltl-formula $\varphi$

**Check:** $M \models \varphi$?

**Solution:**