Computer aided verification

Lecture 3: ω-automata

Model checking

PSPACE-complete

- input: M, ϕ
- question: $M \models \phi$?

Satisfiability

PSPACE-complete

- input: ϕ
- question: $\exists M.\ M \models \phi$?

Complexity of model checking:

$$|M| \cdot 2^{\mathcal{O}(|\phi|)}$$

$2^{\mathcal{O}(|\phi|)}$ OK

$|M|$ too much!

(1) $M \mapsto \mathcal{A}_M$

(2) $\neg\phi \mapsto \mathcal{A}_{\neg\phi}$ LTL $\rightarrow$ ω-automata

(3) $L(\mathcal{A}_M \times \mathcal{A}_{\neg\phi}) = \emptyset?$

yes $\rightarrow M \models \phi$

no $\rightarrow \neg(M \models \phi)$, counterexample = a path in M

Algorithm

(1) $M \mapsto \mathcal{A}_M$

(2) $\neg\phi \mapsto \mathcal{A}_{\neg\phi}$

LTL $\rightarrow$ ω-automata

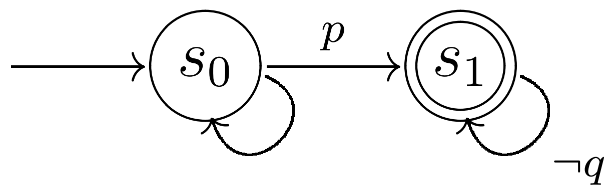
(3) $L(\mathcal{A}_M \times \mathcal{A}_{\neg\phi}) = \emptyset?$

yes $\rightarrow$ $M \vDash \phi$

no $\rightarrow$ $\neg(M \vDash \phi)$, counterexample = a path in M

$\phi = \mathsf{G}\,(p \implies \mathsf{X}\,\mathsf{F}\,q)$

$A_{\neg\phi} =$



I. ω-automata

Def.: ω-automaton (**Büchi** automaton) $\mathcal{A} = \langle \Sigma, S, S_{\text{init}}, \sigma, F \rangle$

– $S_{\text{init}} \subseteq S$ nonempty subset of initial states

– $\sigma \subseteq S \times \Sigma \times S$ transition relation

– $F \subseteq S$ nonempty subset of accepting states

$\mathcal{A}$ is deterministic when $|S_{\text{init}}| = 1$ and $\forall s, a.\ |\sigma(s, a)| \leq 1$.

Def.: ω-automaton (**Büchi** automaton) $\mathcal{A} = \langle \Sigma, S, S_{\text{init}}, \sigma, F \rangle$

- $S_{\text{init}} \subseteq S$ nonempty subset of initial states
- $\sigma \subseteq S \times \Sigma \times S$ transition relation
- $F \subseteq S$ nonempty subset of accepting states

$\mathcal{A}$ is deterministic when $|S_{\text{init}}| = 1$ and $\forall s, a.\ |\sigma(s, a)| \leq 1$.

ω-words: $w = a_0\, a_1\, a_2\, \ldots$

Def.: For $w = a_0\ a_1\ a_2\ \ldots$, a run of an automaton $\mathcal{A}$ is $r = s_0\ s_1\ s_2\ \ldots$ such that $\forall i.\ (s_i, a_i, s_{i+1}) \in \sigma$.

Def.: For $w = a_0\ a_1\ a_2\ \ldots$, a run of an automaton $\mathcal{A}$ is $r = s_0\ s_1\ s_2\ \ldots$ such that $\forall i.\ (s_i, a_i, s_{i+1}) \in \sigma$.

A run is accepting when $s_i \in F$ for infinitely many i.

Def.: For $w = a_0\, a_1\, a_2\, \ldots$, a run of an automaton $\mathcal{A}$ is $r = s_0\, s_1\, s_2\, \ldots$ such that $\forall i.\ (s_i, a_i, s_{i+1}) \in \sigma$.

A run is accepting when $s_i \in F$ for infinitely many i.

Let $\inf(r) = \{s \in S : s = s_i$ for infinitely many $i\}$.
A run is accepting when $\inf(r) \cap F \neq \emptyset$.

Def.: For $w = a_0\ a_1\ a_2\ \ldots$, a run of an automaton $\mathcal{A}$ is $r = s_0\ s_1\ s_2\ \ldots$ such that $\forall i.\ (s_i, a_i, s_{i+1}) \in \sigma$.

A run is accepting when $s_i \in F$ for infinitely many i.

Let $\inf(r) = \{s \in S : s = s_i$ for infinitely many $i\}$.
A run is accepting when $\inf(r) \cap F \neq \emptyset$.

$L_\omega(\mathcal{A}) := \{w : \mathcal{A}$ has an accepting run for $w\}$.

Def.: For $w = a_0\, a_1\, a_2\, \ldots$, a run of an automaton $\mathcal{A}$ is $r = s_0\, s_1\, s_2\, \ldots$ such that $\forall i.\ (s_i, a_i, s_{i+1}) \in \sigma$.

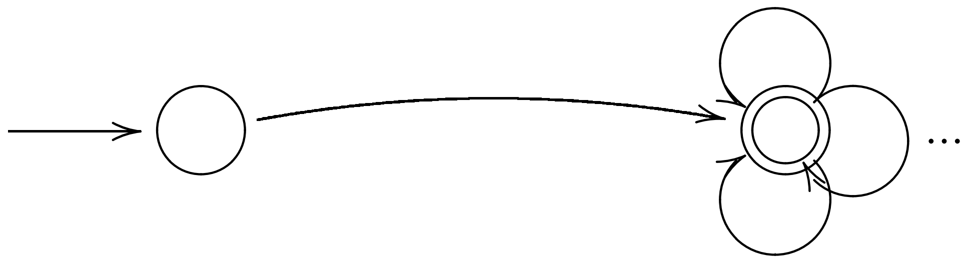
A run is accepting when $s_i \in F$ for infinitely many i.

Let $\inf(r) = \{s \in S : s = s_i$ for infinitely many $i\}$.
A run is accepting when $\inf(r) \cap F \neq \emptyset$.

$L_\omega(\mathcal{A}) := \{w : \mathcal{A}$ has an accepting run for $w\}$.

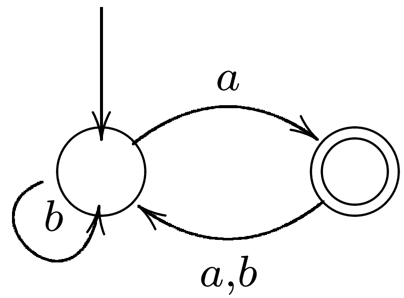
Def.: A language is $L \subseteq \Sigma^\omega$ is ω-regular if $L = L_\omega(\mathcal{A})$ for some $\mathcal{A}$.

An accepting run looks like:

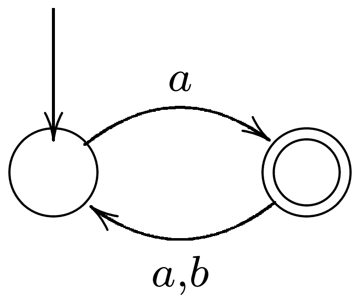


$\Sigma = \{a, b\}$

infinitely often a $(b^* \ a)^\omega$



odd(a) $(a \ (a + b))^\omega$



Corollary: $LTL \subsetneq \omega$-automata

Exercise

– infinitely often a and b

– between any two consecutive a’s
even number of b’s

Exercise

– infinitely often a and b

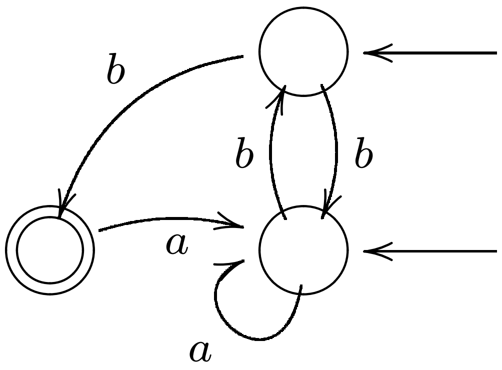
$$b^* \ (aa^* \ bb(bb)^*)^\omega$$

– between any two consecutive a's even number of b's

– infinitely often a and b

$$b^* \ (aa^* \ bb(bb)^*)^\omega$$

– between any two consecutive a's even number of b's



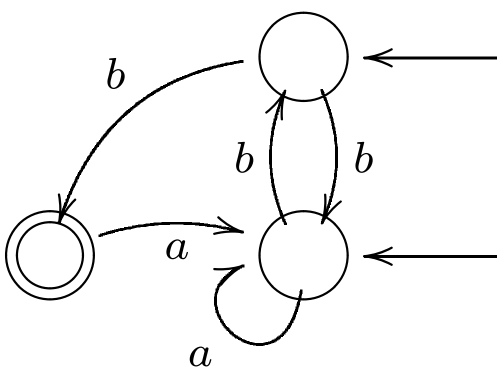
and what about deterministic ?

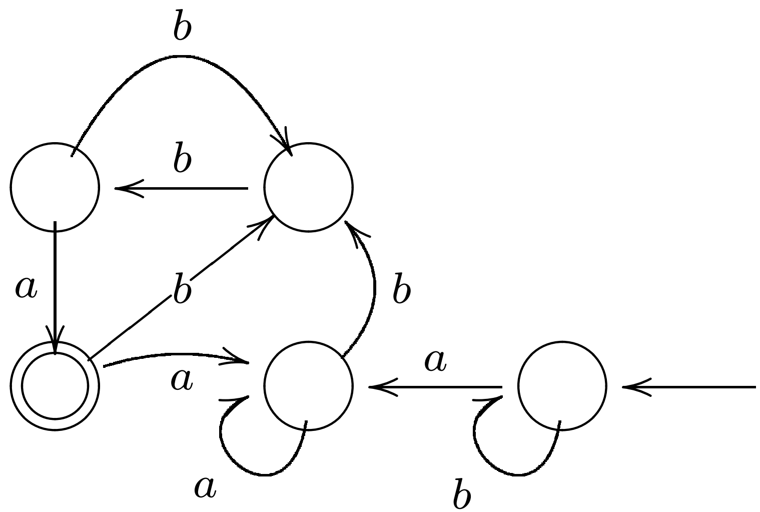
Exercise

– infinitely often a and b

$$b^* \ (aa^* \ bb(bb)^*)^\omega$$

– between any two consecutive a's
even number of b's





Exercise

$\Sigma = \{a, b\}$

finitely often a

Exercise

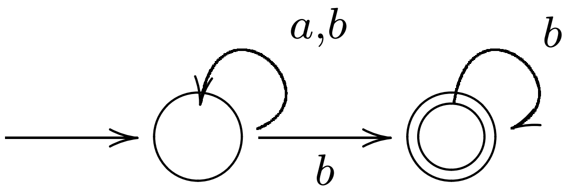
$\Sigma = \{a, b\}$

finitely often a $(b + a)^* b^\omega$

Exercise

$\Sigma = \{a, b\}$

finitely often a $\quad (b + a)^* b^\omega$



and what about deterministic ?

Tw: ω-regular languages are closed under $\cup$, $\cap$ and complementation.

$\vee$, $\wedge$ and $\neg$

$\mathcal{A}_1, \mathcal{A}_2 \;\mapsto\; \mathcal{A}$

(1) $L_\omega(\mathcal{A}) \;=\; L_\omega(\mathcal{A}_1) \;\cup\; L_\omega(\mathcal{A}_2)$

(2) $L_\omega(\mathcal{A}) \;=\; L_\omega(\mathcal{A}_1) \;\cap\; L_\omega(\mathcal{A}_2)$

$\mathcal{A} \;\mapsto\; \bar{\mathcal{A}}$

(3) $L_\omega(\bar{\mathcal{A}}) \;=\; \Sigma^\omega \;\setminus\; L_\omega(\mathcal{A})$

(2) $\mathcal{A}_1, \mathcal{A}_2 \mapsto \mathcal{A}$ $\qquad L_\omega(\mathcal{A}) = L_\omega(\mathcal{A}_1) \cap L_\omega(\mathcal{A}_2)$

?

(2) $\mathcal{A}_1, \mathcal{A}_2 \ \mapsto \ \mathcal{A}$ $\qquad L_\omega(\mathcal{A}) \ = \ L_\omega(\mathcal{A}_1) \ \cap \ L_\omega(\mathcal{A}_2)$

$$S \ = \ S_1 \times S_2 \times \{1, 2\}$$

$$S_{\text{init}} \ = \ S_{1,\text{init}} \times S_{2,\text{init}} \times \{1\}$$

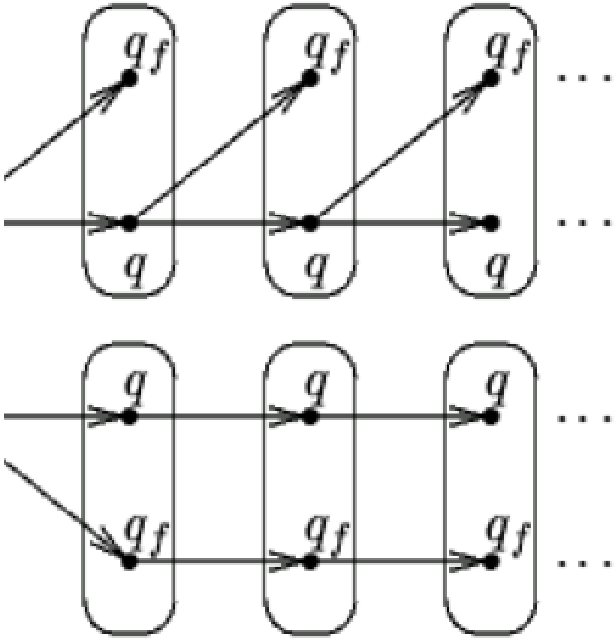
$$F \ = \ F_1 \times S_2 \times \{1\}$$

$$((s, t, i), a, (s', t', j)) \in \sigma \iff (s, a, s') \in \sigma_1, \ (t, a, t') \in \sigma_2,$$

$$j \ = \ \begin{cases} 2 & \text{if } i = 1, \ s \in F_1 \\ 1 & \text{if } i = 2, \ t \in F_2 \\ i & \text{otherwise} \end{cases}$$

(3) $\mathcal{A} \mapsto \bar{\mathcal{A}}$ $\qquad L_\omega(\bar{\mathcal{A}}) = \Sigma^\omega \setminus L_\omega(\mathcal{A})$

– no determinization!



No determinization!

Thm:
$(a+b)^*b^\omega$ is not accepted by a deterministic automaton.

Thm:
$(a+b)^*b^\omega$ is not accepted by a deterministic automaton.

Dowód: Assume $L_\omega(\mathcal{A}) = (a+b)^*b^\omega$, $\mathcal{A}$ deterministic.

No determinization!

Thm:
$(a+b)^*b^\omega$ is not accepted by a deterministic automaton.

Dowód: Assume $L_\omega(\mathcal{A}) = (a+b)^*b^\omega$, $\mathcal{A}$ deterministic.

$w_0 = b^\omega$. For some k_0, $\sigma(s_0, b^{k_0}) \in F$.

No determinization!

Thm:
$(a+b)^*b^\omega$ is not accepted by a deterministic automaton.

Dowód: Assume $L_\omega(\mathcal{A}) = (a+b)^*b^\omega$, $\mathcal{A}$ deterministic.

$w_0 = b^\omega$. For some k_0, $\sigma(s_0, b^{k_0}) \in F$.
$w_1 = b^{k_0}ab^\omega$. For some k_1, $\sigma(s_0, b^{k_0}ab^{k_1}) \in F$.
...

No determinization!

Thm:
$(a+b)^*b^\omega$ is not accepted by a deterministic automaton.

Dowód: Assume $L_\omega(\mathcal{A}) = (a+b)^*b^\omega$, $\mathcal{A}$ deterministic.

$w_0 = b^\omega$. For some k_0, $\sigma(s_0, b^{k_0}) \in F$.
$w_1 = b^{k_0}ab^\omega$. For some k_1, $\sigma(s_0, b^{k_0}ab^{k_1}) \in F$.
...

$\exists i < j$ such that $\sigma(s_0, b^{k_0}ab^{k_1}\ldots ab^{k_i}) = \sigma(s_0, b^{k_0}ab^{k_1}\ldots ab^{k_j})$

No determinization!

Thm:
$(a+b)^*b^\omega$ is not accepted by a deterministic automaton.

Dowód: Assume $L_\omega(\mathcal{A}) = (a+b)^*b^\omega$, $\mathcal{A}$ deterministic.

$w_0 = b^\omega$. For some k_0, $\sigma(s_0, b^{k_0}) \in F$.
$w_1 = b^{k_0}ab^\omega$. For some k_1, $\sigma(s_0, b^{k_0}ab^{k_1}) \in F$.
...

$\exists i < j$ such that $\sigma(s_0, b^{k_0}ab^{k_1}\ldots ab^{k_i}) = \sigma(s_0, b^{k_0}ab^{k_1}\ldots ab^{k_j})$

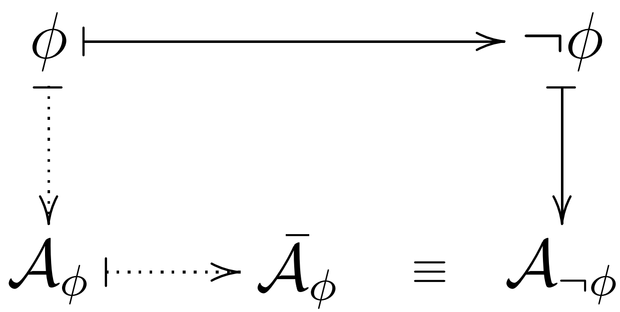
Thus $\mathcal{A}$ accepts $b^{k_0}ab^{k_1}\ldots ab^{k_i}(a\ldots ab^{k_j})^\omega$

contradiction!

(3) $\mathcal{A} \mapsto \bar{\mathcal{A}}$ $\qquad L_\omega(\bar{\mathcal{A}}) = \Sigma^\omega \setminus L_\omega(\mathcal{A})$

- no determinization
- a complex construction
- $|\bar{\mathcal{A}}| = 2^{\mathcal{O}(n \cdot \log n)}$, where $n = |\mathcal{A}|$

Moral: Better to avoid complementation



Question:

How complementation is done if $\mathcal{A}$ is deterministic?

$$\mathcal{A} \mapsto \bar{\mathcal{A}}$$

$$F \mapsto \bar{F} = Q \backslash F$$

$L_\omega(\bar{\mathcal{A}}) \;=\; \Sigma^\omega \;\backslash\; L_\omega(\mathcal{A})$?

Question:

How complementation is done if $\mathcal{A}$ is deterministic?

$$\mathcal{A} \mapsto \bar{\mathcal{A}}$$

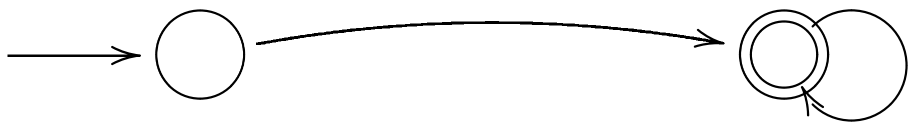
$$F \mapsto \bar{F} = Q \setminus F$$

$L_\omega(\bar{\mathcal{A}}) \; = \; \Sigma^\omega \; \setminus \; L_\omega(\mathcal{A})$? **NO!**

co-Büchi: a run $r \; = \; s_0 \; s_1 \; s_2 \; \ldots$ is accepting when $s_i \in \bar{F}$ for almost all i ($\inf(r) \subseteq \bar{F}$).

problem for finite automata	problem for ω-automata	complexity	cost of algorithm
$L(A) \neq \emptyset$	$L_\omega(A) \neq \emptyset$	NLOGSPACE	$\mathcal{O}(n)$
$L(A) = \Sigma^*$	$L_\omega(A) = \Sigma^\omega$	PSPACE	$2^{\mathcal{O}(n \cdot \log n)}$
$L(A) \subseteq L(B)$	$L_\omega(A) \subseteq L_\omega(B)$	PSPACE	$2^{\mathcal{O}(n \cdot \log n)}$

Lasso



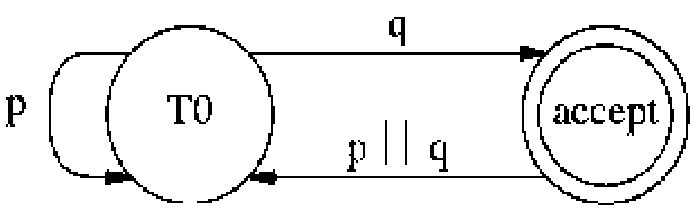
Thm: $L_\omega(A) \neq \emptyset$ iff $\mathcal{A}$ has a lasso.

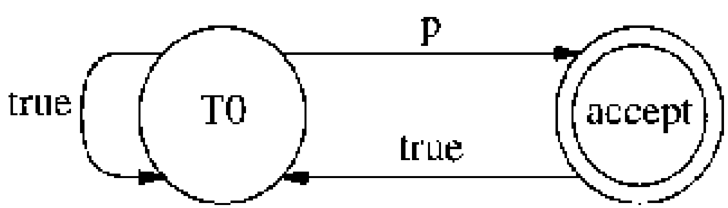
II. LTL $\longmapsto$ BA

SPIN – examples

```
$ spin -f "[](p U q)"
never {
T0:
        if
        :: (p) -> goto T0
        :: (q) -> goto accept
        fi;
accept:
        if
        :: ((p) || (q)) -> goto T0
        fi
}
```

```
$ spin -f "[]<>p"
never {
T0:
        if
        :: (true) -> goto T0
        :: (p) -> goto accept
        fi;
accept:
        if
        :: (true) -> goto T0
        fi
}
```





SPIN's doc

Generalized ω-automata (GBA)

- $\{F_1, \ldots, F_n\}$ instead of F
- a run r is accepting when $\forall i.\ \inf(r) \cap F_i \neq \emptyset$

Question: Are generalized automata more expressive?

Generalized ω-automata (GBA)

- $\{F_1, \ldots, F_n\}$ instead of F
- a run r is accepting when $\forall i.\ \inf(r) \cap F_i \neq \emptyset$

Question: Are generalized automata more expressive?

$$\mathcal{A}_{F_1 \ldots F_n} \mapsto \mathcal{A}_F$$

$$L_\omega(\mathcal{A}_{F_1 \ldots F_n}) = L_\omega(\mathcal{A}_F) \subseteq L_\omega(\mathcal{A}_{F_1}) \cap \ldots \cap L_\omega(\mathcal{A}_{F_n})$$

$$|\mathcal{A}_F| = \mathcal{O}(|\mathcal{A}_{F_1,\ldots,F_n}| \cdot n)$$

- **SPIN:** LTL $\mapsto$ GBA $\mapsto$ BA
- **LTL2BA:** LTL $\mapsto$ ABA $\mapsto$ GBA' $\mapsto$ BA

- On-the-fly verification

LTL$^+$:

$$\phi \;:=\; p \mid \neg p \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \mathsf{X}\,\phi \mid \phi_1\,\mathsf{U}\,\phi_2 \mid \phi_1\,\mathsf{R}\,\phi_2 \mid$$

$$\texttt{true} \mid \texttt{false}$$

Intuition: $\phi \;\equiv\; \mathsf{now}(\phi) \overset{\wedge}{\underset{\vee}{-}} \mathsf{later}(\phi)$

$$\phi\,\mathsf{U}\,\psi \;\equiv\; \psi \;\vee\; (\phi \wedge \mathsf{X}\,(\phi\,\mathsf{U}\,\psi))$$

$$\phi\,\mathsf{R}\,\psi \;\equiv\; \psi \;\wedge\; (\phi \vee \mathsf{X}\,(\phi\,\mathsf{R}\,\psi))$$

(fixed points)

. . . do not think about tomorrow! :)

$\alpha \quad \mapsto \quad$ today(α) – boolean formula over $P \cup \bar{P} \cup \{\, \mathsf{X}\,\phi : \phi \ldots \}$

$$\bar{P} = \{\neg p : p \in P\}$$

$$\begin{array}{lcl}
\text{today}(\alpha) & = & \alpha, \quad \text{for} \quad \alpha = p, \neg p, \mathsf{X}\,\beta, \texttt{true}, \texttt{false} \\
\text{today}(\alpha \vee \beta) & = & \text{today}(\alpha) \;\vee\; \text{today}(\beta) \\
\text{today}(\alpha \wedge \beta) & = & \text{today}(\alpha) \;\wedge\; \text{today}(\beta) \\
\text{today}(\alpha\,\mathsf{U}\,\beta) & = & \text{today}(\beta) \;\vee\; (\text{today}(\alpha) \wedge \mathsf{X}\,(\alpha\,\mathsf{U}\,\beta)) \\
\text{today}(\alpha\,\mathsf{R}\,\beta) & = & \text{today}(\beta) \;\wedge\; (\text{today}(\alpha) \vee \mathsf{X}\,(\alpha\,\mathsf{R}\,\beta))
\end{array}$$

$$\alpha \;\mapsto\; \text{today}(\alpha) \;\mapsto\; \text{dnf}(\alpha) \;\subseteq\; \mathcal{P}(P \cup \bar{P} \cup \{\mathsf{X}\,\phi : \phi \ldots\})$$

$$\boxed{\text{today}(\alpha) \;\equiv\; \bigvee_{X \in \text{dnf}(\alpha)} \left(\bigwedge X\right)}$$

For example:

$$\begin{array}{lcll}
\text{dnf}(\alpha) & = & \{\{\alpha\}\}, \;\; \text{gdy} \;\; \alpha = p, \neg p, \mathsf{X}\,\beta & \\
\text{dnf}(\alpha \vee \beta) & = & \text{dnf}(\alpha) \cup \text{dnf}(\beta) & \\
\text{dnf}(\alpha\,\mathsf{U}\,\beta) & = & \text{dnf}(\beta) \cup \text{dnf}(\alpha \wedge \mathsf{X}\,(\alpha\,\mathsf{U}\,\beta)) & \\
\text{dnf}(\mathtt{true}) & = & \{\emptyset\} & \bigwedge\emptyset \equiv \mathtt{true} \\
\text{dnf}(\mathtt{false}) & = & \emptyset & \bigvee\emptyset \equiv \mathtt{false}
\end{array}$$

GBA $\mathcal{A}_\phi = \langle \Sigma, S, S_{\text{init}}, \sigma, F \rangle$:

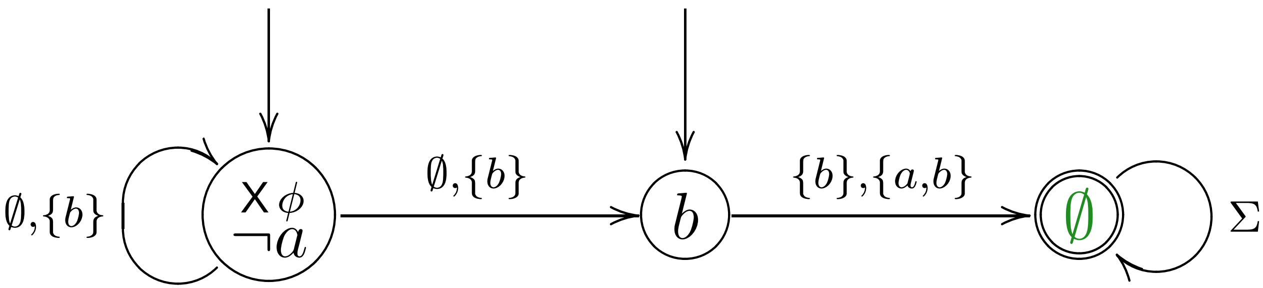
- $S = \mathcal{P}(P \cup \bar{P} \cup \{ \mathsf{X}\, \phi : \phi \ldots \})$
- $\Sigma = \mathcal{P}(P)$
- $S_{\text{init}} = \text{dnf}(\phi)$
- $X \xrightarrow{A} Y$ iff
  - $X \cap P \subseteq A$ — non-contradictory
  - $(X \cap \bar{P}) \cap A = \emptyset$ — X and A today
  - $Y \in \text{dnf}(\wedge \{\alpha \mid \mathsf{X}\, \alpha \in X\})$ — possible tomorrow
- $F = ?$

LTL $\mapsto$ GBA (example 1)

$\phi = \neg a \, \mathsf{U} \, b$

$S = \mathcal{P}(a,\ \neg a,\ b,\ \neg b,\ \mathsf{X}\,(\neg a \, \mathsf{U} \, b))$

$\Sigma = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$



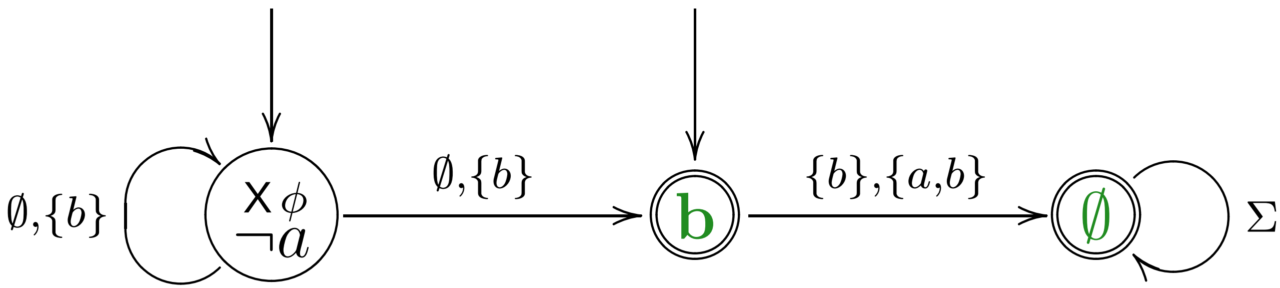
$F = ?$

LTL $\mapsto$ GBA (example 1)

$\phi = \neg a \,\mathsf{U}\, b$

$S = \mathcal{P}(a, \neg a, b, \neg b, \mathsf{X}(\neg a \,\mathsf{U}\, b))$

$\Sigma = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$



$F = \{\{\emptyset, \{b\}\}\}$

- $F_i = \{A \mid \alpha_i \,\mathsf{U}\, \beta_i \notin A \;\vee\; \beta_i \in A\}, \quad i = 1, \ldots, n$

$$\{\alpha_i \,\mathsf{U}\, \beta_i \mid i = 1, \ldots, n\} \subseteq \text{subformula}(\phi)$$

- $F_i = \{X \in S \mid \alpha_i \,\mathsf{U}\, \beta_i \notin \text{cons}(X) \;\vee\; \beta_i \in \text{cons}(X)\}$

$X \subseteq \text{cons}(X)$

$\alpha \vee \beta \in \text{cons}(X)$ jeśli $\alpha \in \text{cons}(X)$ lub $\beta \in \text{cons}(X)$

$\alpha \wedge \beta \in \text{cons}(X)$ jeśli $\alpha \in \text{cons}(X)$ i $\beta \in \text{cons}(X)$

$\alpha \,\mathsf{U}\, \beta \in \text{cons}(X)$ jeśli $\beta \;\vee\; (\alpha \wedge \mathsf{X}\,(\alpha \,\mathsf{U}\, \beta)) \in \text{cons}(X)$

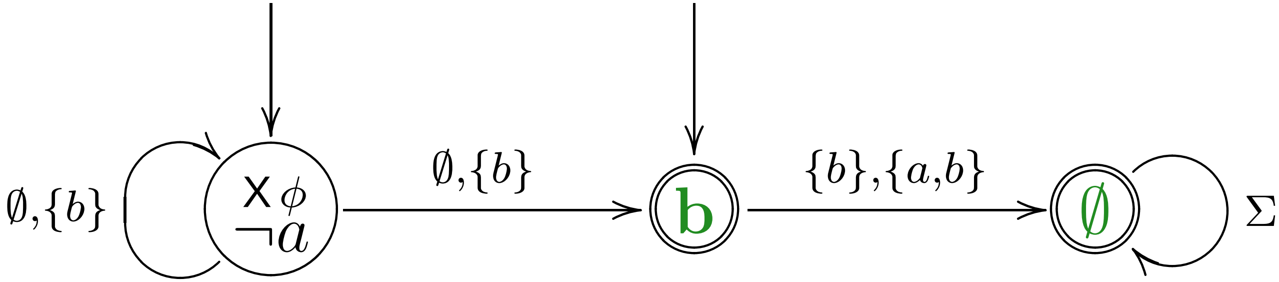
$\alpha \,\mathsf{R}\, \beta \in \text{cons}(X)$ jeśli $\beta \;\wedge\; (\alpha \vee \mathsf{X}\,(\alpha \,\mathsf{R}\, \beta)) \in \text{cons}(X)$

LTL $\mapsto$ GBA (example 1 cont.)

$\phi = \neg a \,\mathsf{U}\, b$

$S = \mathcal{P}(a,\ \neg a,\ b,\ \neg b,\ \mathsf{X}\,(\neg a \,\mathsf{U}\, b))$

$\Sigma = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$



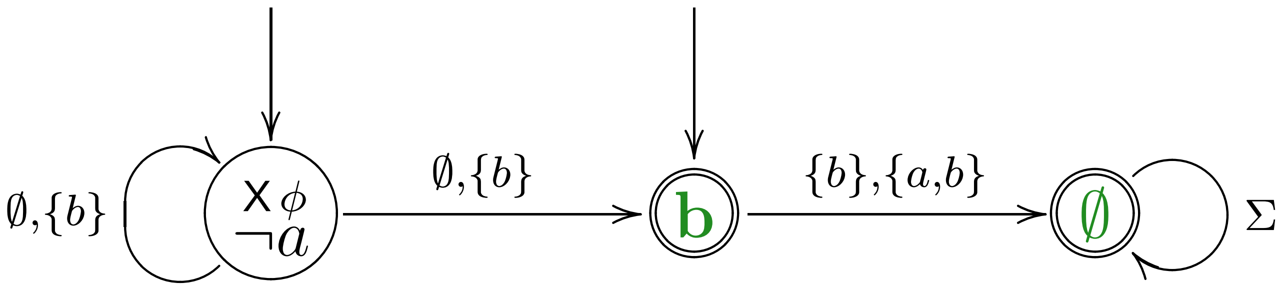
Can automaton $\mathcal{A}_\phi$ be smaller?

LTL $\mapsto$ GBA (example 1 cont.)

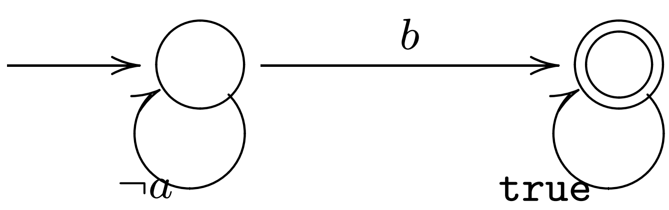
$\phi = \neg a \, \mathsf{U} \, b$

$S = \mathcal{P}(a, \neg a, b, \neg b, \mathsf{X}(\neg a \, \mathsf{U} \, b))$

$\Sigma = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$



Can automaton $\mathcal{A}_\phi$ be smaller? **YES!**



LTL $\mapsto$ GBA (example 2)

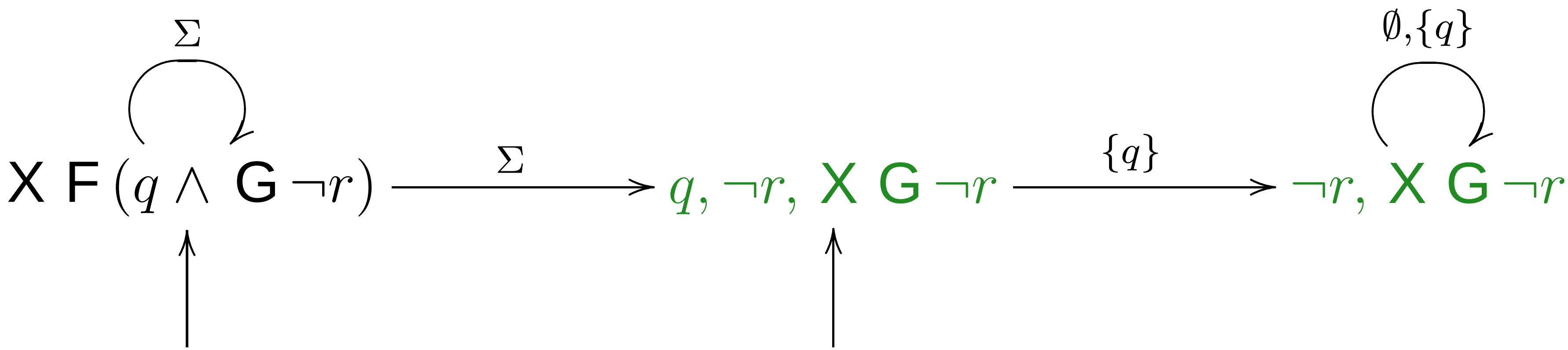
$$\theta = \neg\, \mathsf{G}\,(q \implies \mathsf{F}\,r) \equiv \mathsf{F}\,(q \wedge \mathsf{G}\,\neg r)$$

$$\text{dnf}(\mathsf{F}\,\alpha) = \text{dnf}(\alpha) \cup \{\{\mathsf{X}\,\mathsf{F}\,\alpha\}\} \qquad \mathsf{F}\,\alpha \equiv \alpha \vee \mathsf{X}\,\mathsf{F}\,\alpha$$

$$\text{dnf}(\mathsf{G}\,\alpha) = \text{dnf}(\alpha \wedge \mathsf{X}\,\mathsf{G}\,\alpha) \qquad \mathsf{G}\,\alpha \equiv \alpha \wedge \mathsf{X}\,\mathsf{G}\,\alpha$$

$$S = \mathcal{P}(q,\ \neg q,\ r,\ \neg r,\ \mathsf{X}\,(\mathsf{F}\,(q \wedge \mathsf{G}\,\neg r)),\ \mathsf{X}\,\mathsf{G}\,\neg r) \qquad \mathsf{F} = ?$$

$$\text{dnf}(\mathsf{F}\,(q \wedge \mathsf{G}\,\neg r)) = \{\{\mathsf{X}\,\mathsf{F}\,(q \wedge \mathsf{G}\,\neg r)\},\ \{q, \neg r, \mathsf{X}\,\mathsf{G}\,\neg r\}\}$$



LTL $\mapsto$ GBA (example 2)

$$\theta = \neg(\mathsf{G}\,\mathsf{F}\,p \implies \mathsf{G}\,(q \implies \mathsf{F}\,r)) \equiv \mathsf{G}\,\mathsf{F}\,p \wedge \mathsf{F}\,(q \wedge \mathsf{G}\,\neg r)$$

$$\mathsf{dnf}(\mathsf{F}\,(q \wedge \mathsf{G}\,\neg r)) = \mathsf{X}\,\mathsf{F}\,(q \wedge \mathsf{G}\,\neg r) \vee (q \wedge \neg r \wedge \mathsf{X}\,\mathsf{G}\,\neg r)$$

$$\mathsf{dnf}(\mathsf{G}\,\mathsf{F}\,p) = \mathsf{dnf}((p \vee \mathsf{X}\,\mathsf{F}\,p) \wedge \mathsf{X}\,\mathsf{G}\,\mathsf{F}\,p) = (p \wedge \mathsf{X}\,\mathsf{G}\,\mathsf{F}\,p) \vee (\mathsf{X}\,\mathsf{F}\,p \wedge \mathsf{X}\,\mathsf{G}\,\mathsf{F}\,p)$$

$$\mathsf{dnf}(\mathsf{G}\,\mathsf{F}\,p \wedge \mathsf{F}\,(q \wedge \mathsf{G}\,\neg r)) = \ldots \vee \ldots \vee \ldots \vee \ldots$$

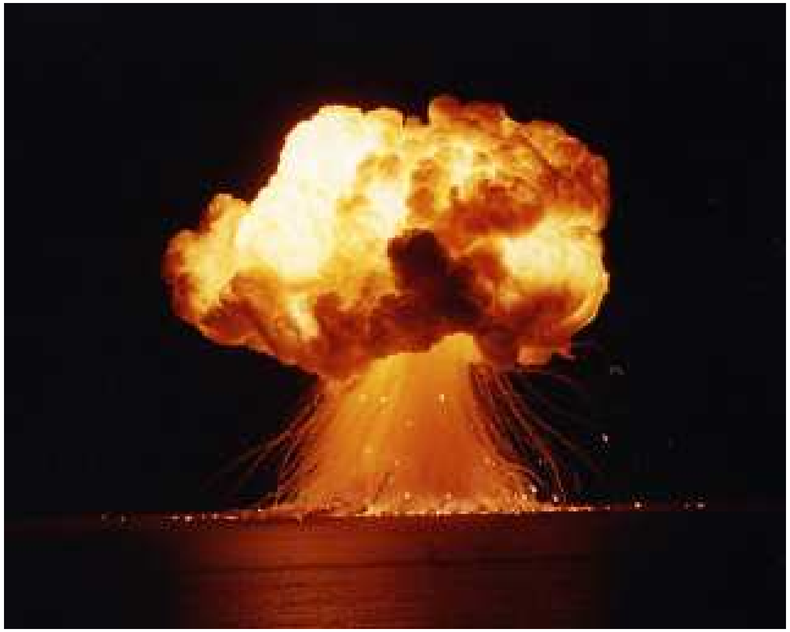
$\mathsf{X}\,\mathsf{F}\,(q \wedge \mathsf{G}\,\neg r), p, \mathsf{X}\,\mathsf{G}\,\mathsf{F}\,p$ $\qquad$ $q, \neg r, \mathsf{X}\,\mathsf{G}\,\neg r, p, \mathsf{X}\,\mathsf{G}\,\mathsf{F}\,p$

$\mathsf{X}\,\mathsf{F}\,(q \wedge \mathsf{G}\,\neg r), \mathsf{X}\,\mathsf{F}\,p, \mathsf{X}\,\mathsf{G}\,\mathsf{F}\,p$ $\qquad$ $q, \neg r, \mathsf{X}\,\mathsf{G}\,\neg r, \mathsf{X}\,\mathsf{F}\,p, \mathsf{X}\,\mathsf{G}\,\mathsf{F}\,p$

LTL $\mapsto$ GBA (example 2)

$$\theta_n = \neg((\mathsf{G}\,\mathsf{F}\,p_1 \wedge \ldots \wedge \mathsf{G}\,\mathsf{F}\,p_n) \implies \mathsf{G}\,(q \implies \mathsf{F}\,r)) \equiv$$

$$\mathsf{G}\,\mathsf{F}\,p_1 \wedge \ldots \wedge \mathsf{G}\,\mathsf{F}\,p_n \wedge \mathsf{F}\,(q \wedge \mathsf{G}\,\neg r)$$



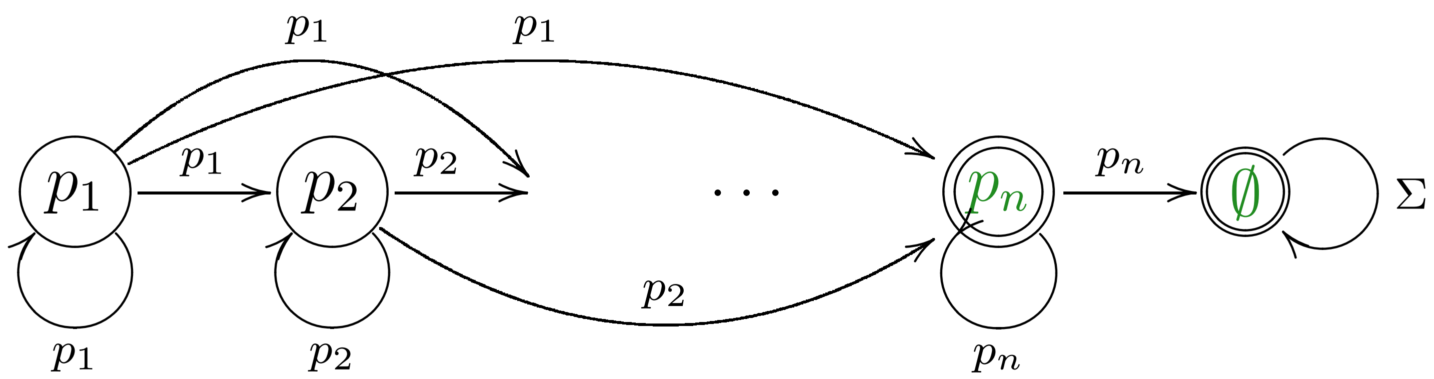
LTL $\mapsto$ GBA (example 2)

$$\theta_n = \neg((\mathsf{G}\,\mathsf{F}\,p_1 \wedge \ldots \wedge \mathsf{G}\,\mathsf{F}\,p_n) \implies \mathsf{G}\,(q \implies \mathsf{F}\,r))$$

	Spin		Wring		EQLTL	LTL2BA–		LTL2BA	
	time	space	time	space	time	time	space	time	space
θ_1	0.18	460	0.56	4,100	16	0.01	9	0.01	9
θ_2	4.6	4,200	2.6	4,100	16	0.01	19	0.01	11
θ_3	170	52,000	16	4,200	18	0.01	86	0.01	19
θ_4	9,600	970,000	110	4,700	25	0.07	336	0.06	38
θ_5			1,000	6,500	135	0.70	1,600	0.37	48
θ_6			8,400	13,000	N/A	12	8,300	4.0	88
θ_7			72,000†	43,000†		220	44,000	32	175
θ_8						4,200	260,000	360	250
θ_9						97,000	1,600,000	3,000	490
θ_{10}								36,000	970

[Gastin,Oddoux 2001]

$$\phi_n \; = \; p_1 \, \mathsf{U} \, (p_2 \, \mathsf{U} \, (\ldots \, \mathsf{U} \, p_n) \ldots)$$



$$\theta_n \; = \; \neg\big(p_1 \, \mathsf{U} \, (p_2 \, \mathsf{U} \, (\ldots \, \mathsf{U} \, p_n) \ldots)\big)$$