

Chapter 4

LTL model checking

Course “Specification and Verification of Parallel Systems”

Juliane Päßler, Tobias John

Spring 2024

Overview

- we continue with LTL model checking
- LTL to BA
  - LTL to GBA
  - GBA to BA

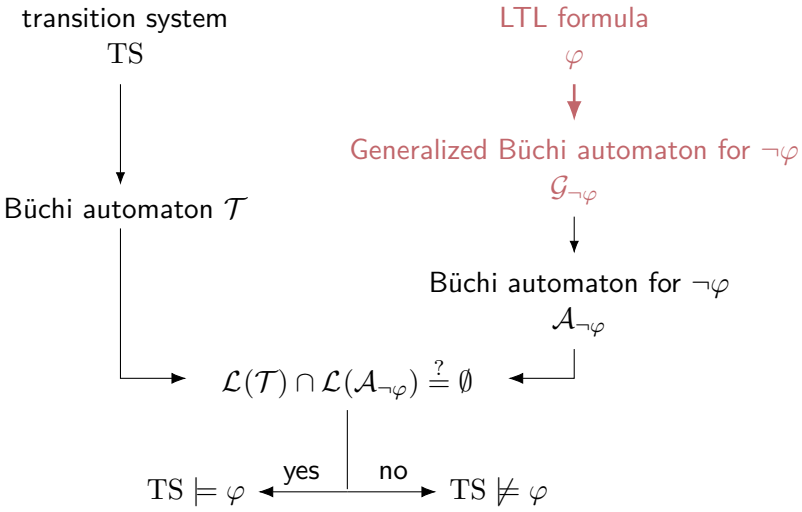




Section
LTL to BA

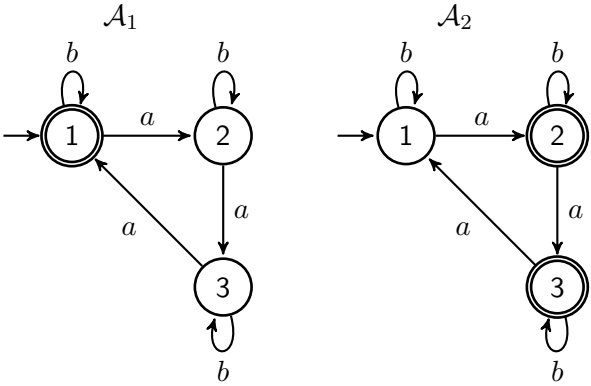
Spring 2024

Overview

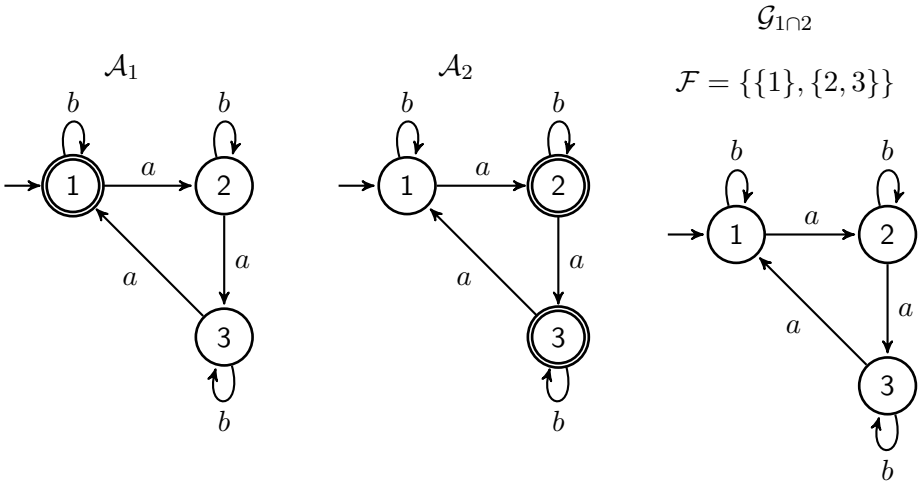

transition system
TS
Büchi automaton $\mathcal{T}$
LTL formula
φ
Generalized Büchi automaton for $\neg\varphi$
$\mathcal{G}_{\neg\varphi}$
Büchi automaton for $\neg\varphi$
$\mathcal{A}_{\neg\varphi}$
$\mathcal{L}(\mathcal{T}) \cap \mathcal{L}(\mathcal{A}_{\neg\varphi}) \stackrel{?}{=} \emptyset$
yes
no
TS $\models \varphi$
TS $\not\models \varphi$

GBA: motivating example (language intersection)

$\mathcal{G}_{1\cap 2}$



GBA: motivating example (language intersection)



- note: $\mathcal{G}_{1\cap 2}$ is different from BA with $F = \{1,2,3\}$ (e.g. consider word b^ω)

Generalized Büchi automata (GBA / GNBA)

Definition (Generalized Büchi automaton)

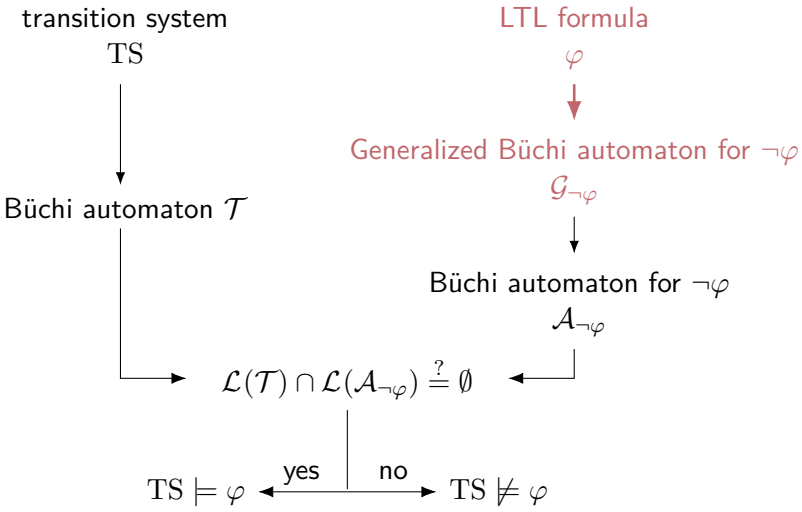
A *generalized Büchi automaton* is an automaton $\mathcal{A} = (Q, q_0, \Sigma, \mathcal{F}, \rightarrow)$, where $\mathcal{F} \subseteq 2^Q$.
Let $\mathcal{F} = \{F_1, \ldots, F_k\}$ and $F_i \subseteq Q$. A run σ of $\mathcal{A}$ is *accepting* if

$$\text{for each } F_i \in \mathcal{F},\ inf(\sigma) \cap F_i \neq \emptyset.$$

- $inf(\sigma)$: states visited infinitely often in σ
- generalized Büchi automaton: multiple accepting sets instead of only one (as in "original" Büchi Automata)
- generalized Büchi automata: *equally* expressive as Büchi automata (we will see later why...)



Overview


transition system
TS
Büchi automaton $\mathcal{T}$
LTL formula
φ
Generalized Büchi automaton for $\neg\varphi$
$\mathcal{G}_{\neg\varphi}$
Büchi automaton for $\neg\varphi$
$\mathcal{A}_{\neg\varphi}$
$\mathcal{L}(\mathcal{T}) \cap \mathcal{L}(\mathcal{A}_{\neg\varphi}) \stackrel{?}{=} \emptyset$
yes
no
TS $\models \varphi$
TS $\not\models \varphi$

Assumptions

- given: LTL formula φ that only uses the following operators:

  $$\wedge, \neg, U, \bigcirc$$

  (remember: this set is complete for LTL)

- goal: construct GBA $\mathcal{G}_\varphi$ for φ
  i.e. $\mathcal{L}(\mathcal{G}_\varphi) = \{w \in \Sigma^\omega \mid w \models \varphi\}$

- remark: we consider φ instead of $\neg\varphi$ as the latter is included in the first case



Basic idea for $\mathcal{G}_\varphi$

- States of automaton: sets of formulas B_i where
  - $A_i \in \Sigma$, sets of atomic props
  - B_i : "extention of A_i" (by sub-formulas of φ), i.e., $B_i \supseteq A_i$.

States as sets of formulas

States capture formulas that are intended to be in the "language of that state". I.e., the B_i's form the states of $\mathcal{G}_\varphi$.

Given: run $\sigma = A_0 A_1 A_2 \ldots$.
Extension to $\hat{\sigma} = B_0 B_1 B_2 \ldots$

$$\psi \in B_i \qquad \text{iff} \qquad \underbrace{A_i, A_{i+1} A_{i+2} \ldots}_{\sigma^i} \models \psi$$

$\hat{\sigma}$ = run (ultimately: state-sequence) in $\mathcal{G}_\varphi$



Example: Extension of states

- we consider the formula $\varphi = \bigcirc b$
- we consider the run $ab\{a,b\}a^\omega$

$$
\begin{array}{ll}
A_0 = \{a\} & B_0 = \\
A_1 = \{b\} & B_1 = \\
A_2 = \{a,b\} & B_2 = \\
A_3 = \{a\} & B_3 = \\
A_4 = \{a\} & B_4 = \\
\vdots & \vdots
\end{array}
$$



Example: Extension of states

- we consider the formula $\varphi = \bigcirc b$
- we consider the run $ab\{a,b\}a^\omega$

$$
\begin{array}{ll}
A_0 = \{a\} & B_0 = \{a, \neg b, \bigcirc b\} \\
A_1 = \{b\} & B_1 = \{\neg a, b, \bigcirc b\} \\
A_2 = \{a, b\} & B_2 = \{a, b, \neg \bigcirc b\} \\
A_3 = \{a\} & B_3 = \{a, \neg b, \neg \bigcirc b\} \\
A_4 = \{a\} & B_4 = \{a, \neg b, \neg \bigcirc b\} \\
\vdots & \vdots
\end{array}
$$



Closure of φ

- Goal: restrict B_is to finite sets (to obtain finite set of states)
- Solution: only capture relevant formulas w.r.t. φ

Definition

The *closure* of an LTL formula φ is the set $closure(\varphi)$ consisting of all subformulas ψ of φ and their negation $\neg\psi$ (where ψ and $\neg\neg\psi$ are identified).

Example

$\varphi = a \ U \ (\neg a \wedge b)$
$closure(\varphi) =$



Closure of φ

- Goal: restrict B_is to finite sets (to obtain finite set of states)
- Solution: only capture relevant formulas w.r.t. φ

Definition

The *closure* of an LTL formula φ is the set $closure(\varphi)$ consisting of all subformulas ψ of φ and their negation $\neg\psi$ (where ψ and $\neg\neg\psi$ are identified).

Example

$\varphi = a \ U \ (\neg a \wedge b)$
$closure(\varphi) = \{a, b, \neg a, \neg b, \neg a \wedge b, \neg(\neg a \wedge b), \varphi, \neg\varphi\}$



Maximal consistent sets of formulas I

- Goal: only use "meaningful" sets of formulas for B_is
  - remember: sets are intended to capture formulas that are satisfied for the state / the outgoing paths
- Solution: use *consistent* and *maximal* sets



Maximal consistent sets of formulas I

- Goal: only use "meaningful" sets of formulas for B_is
  - remember: sets are intended to capture formulas that are satisfied for the state / the outgoing paths
- Solution: use *consistent* and *maximal* sets

Definition (Propositional consistency)

A set B is consistent wrt. propositional logic if

- $\psi \in B$ implies $\neg\psi \notin B$.



Maximal consistent sets of formulas I

- Goal: only use "meaningful" sets of formulas for B_is
  - remember: sets are intended to capture formulas that are satisfied for the state / the outgoing paths
- Solution: use *consistent* and *maximal* sets

Definition (Propositional consistency)

A set B is consistent wrt. propositional logic if

- $\psi \in B$ implies $\neg\psi \notin B$.

Definition (Closed wrt. propositional entailment)

A set B is closed under propositional entailment if

- $\varphi_1 \wedge \varphi_2 \in B$ iff $\varphi_1 \in B$ and $\varphi_2 \in B$.



Maximal consistent sets of formulas II

Definition (Local consequences of until)

A set B is closed under local entailments wrt. the until operator if

1. $\varphi_2 \in B$ implies $\varphi_1 \; U \; \varphi_2 \in B$
2. $\varphi_1 \; U \; \varphi_2 \in B$ and $\varphi_2 \notin B$ implies $\varphi_1 \in B$.



Maximal consistent sets of formulas II

Definition (Local consequences of until)

A set B is closed under local entailments wrt. the until operator if

1. $\varphi_2 \in B$ implies $\varphi_1 \; U \; \varphi_2 \in B$
2. $\varphi_1 \; U \; \varphi_2 \in B$ and $\varphi_2 \notin B$ implies $\varphi_1 \in B$.

Definition (Maximality)

A set is maximal, if for all $\psi \in closure(\varphi)$

$$\psi \notin B \qquad \text{implies} \qquad \neg\psi \in B \; .$$



Maximal consistent sets of formulas III

Definition

Given a LTL-formula φ. A set $B \subseteq closure(\varphi)$ is *maximally consistent* (or elementary) wrt. φ if it is propositionally consistent, closed under propositonal entailment and locally entailed formulas wrt. until, and if it is maximal.



Example: maximal consistent sets

- we consider $\varphi = a \ U \ b$
- $closure(\varphi) =$



Example: maximal consistent sets

- we consider $\varphi = a \ U \ b$
- $closure(\varphi) = \{a, b, \neg a, \neg b, a \ U \ b, \neg(a \ U \ b)\}$

Which of these sets are maximal consistent?

$$\{a, b\} \qquad \{a, b, a \ U \ b\} \qquad \{a, \neg a, a \ U \ b\}$$
$$\{a, b, a \wedge b\} \qquad \{a, b, \neg(a \ U \ b)\} \qquad \{a, \neg b, \neg(a \ U \ b)\}$$

Example: maximal consistent sets

- we consider $\varphi = a \ U \ b$
- $closure(\varphi) = \{a, b, \neg a, \neg b, a \ U \ b, \neg(a \ U \ b)\}$

Which of these sets are maximal consistent?

$\{a, b\}$ $\quad$ $\{a, b, a \ U \ b\}$ $\quad$ $\{a, \neg a, a \ U \ b\}$

$\{a, b, a \wedge b\}$ $\quad$ $\{a, b, \neg(a \ U \ b)\}$ $\quad$ $\{a, \neg b, \neg(a \ U \ b)\}$

What are all the maximal consistent sets?



Example: maximal consistent sets

- we consider $\varphi = a \ U \ b$
- $closure(\varphi) = \{a, b, \neg a, \neg b, a \ U \ b, \neg(a \ U \ b)\}$

Which of these sets are maximal consistent?

$\{a, b\}$ $\quad$ $\{a, b, a \ U \ b\}$ $\quad$ $\{a, \neg a, a \ U \ b\}$

$\{a, b, a \wedge b\}$ $\quad$ $\{a, b, \neg(a \ U \ b)\}$ $\quad$ $\{a, \neg b, \neg(a \ U \ b)\}$

What are all the maximal consistent sets?

$\{a, b, a \ U \ b\}$ $\quad$ $\{a, \neg b, a \ U \ b\}$ $\quad$ $\{a, \neg b, \neg(a \ U \ b)\}$

$\{\neg a, b, a \ U \ b\}$ $\quad$ $\{\neg a, \neg b, \neg(a \ U \ b)\}$



Construction of GNBA: general

- remember
  - given: AP and φ
  - goal: construct an GNBA $\mathcal{G}_\varphi$ such that
$$\mathcal{L}(\mathcal{G}_\varphi) = \{w \in \Sigma^\omega \mid w \models \varphi\}$$
- 3 core ingredients
  1. states: sets of formulas which "hold" for the outgoing paths
  2. transition relation: connects the states appropriately
  3. transitions labeled by sets of AP.

Definition ($\mathcal{G}_\varphi$)

Given AP and φ, we define $\mathcal{G}_\varphi = (Q, I, 2^{\mathrm{AP}}, \mathcal{F}, \delta)$ where

- Q is the set of all maximal consistent sets of formulas $B \subseteq \mathit{closure}(\varphi)$
- $I = \{B \in Q \mid \varphi \in B\}$
- $\mathcal{F}$ and δ as on the next slides defined



Construction of GNBA: transition relation

$$\delta : Q \times 2^{\mathrm{AP}} \to 2^{Q}$$

- if $A \neq B \cap \mathrm{AP}$, then $\delta(B, A) = \emptyset$
- if $A = B \cap \mathrm{AP}$, then $\delta(B, A)$ is the set of all sets $B' \in Q$ such that
  - for every $\bigcirc\psi \in \mathit{closure}(\varphi)$:

$$\bigcirc\psi \in B \quad \text{iff} \quad \psi \in B'$$

  - for every $\varphi_1 \; U \; \varphi_2 \in \mathit{closure}(\varphi)$:

$$\varphi_1 \; U \; \varphi_2 \in B \quad \text{iff} \quad \big(\varphi_2 \in B \quad \text{or} \quad (\varphi_1 \in B \quad \text{and} \quad \varphi_1 \; U \; \varphi_2 \in B')\big)$$



Construction of GNBA: transition relation

$$\delta : Q \times 2^{\mathrm{AP}} \to 2^{Q}$$

- if $A \neq B \cap \mathrm{AP}$, then $\delta(B, A) = \emptyset$
- if $A = B \cap \mathrm{AP}$, then $\delta(B, A)$ is the set of all sets $B' \in Q$ such that
  - for every $\bigcirc\psi \in \mathit{closure}(\varphi)$:

$$\bigcirc\psi \in B \quad \text{iff} \quad \psi \in B'$$

  - for every $\varphi_1 \ U \ \varphi_2 \in \mathit{closure}(\varphi)$:

$$\varphi_1 \ U \ \varphi_2 \in B \quad \text{iff} \quad \big(\varphi_2 \in B \quad \text{or} \quad (\varphi_1 \in B \quad \text{and} \quad \varphi_1 \ U \ \varphi_2 \in B')\big)$$

Example

What are the transitions for the GBA for the formula $\varphi = a \ U \ b$ with the following states?

$$\{a, b, a \ U \ b\} \qquad \{a, \neg b, a \ U \ b\} \qquad \{a, \neg b, \neg(a \ U \ b)\}$$
$$\{\neg a, b, a \ U \ b\} \qquad \{\neg a, \neg b, \neg(a \ U \ b)\}$$



Construction: final states

- idea: introduce one set of final states for each until formula
- acceptance condition ensures that φ_2 is *eventually* true

$$\mathcal{F} = \{F_{\varphi_1 U \varphi_2} \mid \varphi_1 \ U \ \varphi_2 \in closure(\varphi)\}$$

where

$$F_{\varphi_1 U \varphi_2} = \{B \in Q \mid \varphi_2 \in B \text{ or } \varphi_1 \ U \ \varphi_2 \notin B\} \ .$$

- remark: special case of no until formula: $\mathcal{F} = \emptyset$, i.e., each run is accepting



Construction: final states

- idea: introduce one set of final states for each until formula
- acceptance condition ensures that φ_2 is *eventually* true

$$\mathcal{F} = \{F_{\varphi_1 U \varphi_2} \mid \varphi_1 \ U \ \varphi_2 \in closure(\varphi)\}$$

where

$$F_{\varphi_1 U \varphi_2} = \{B \in Q \mid \varphi_2 \in B \text{ or } \varphi_1 \ U \ \varphi_2 \notin B\} \ .$$

- remark: special case of no until formula: $\mathcal{F} = \emptyset$, i.e., each run is accepting

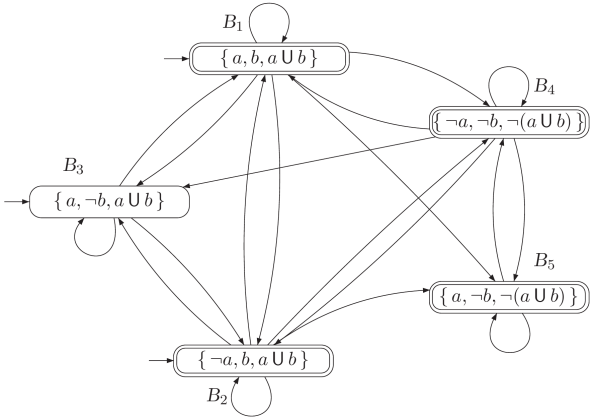
Example

What are the final states for the GBA for the formula $\varphi = a \ U \ b$ with the following states?

$$\{a, b, a \ U \ b\} \quad \{a, \neg b, a \ U \ b\} \quad \{a, \neg b, \neg(a \ U \ b)\}$$
$$\{\neg a, b, a \ U \ b\} \quad \{\neg a, \neg b, \neg(a \ U \ b)\}$$

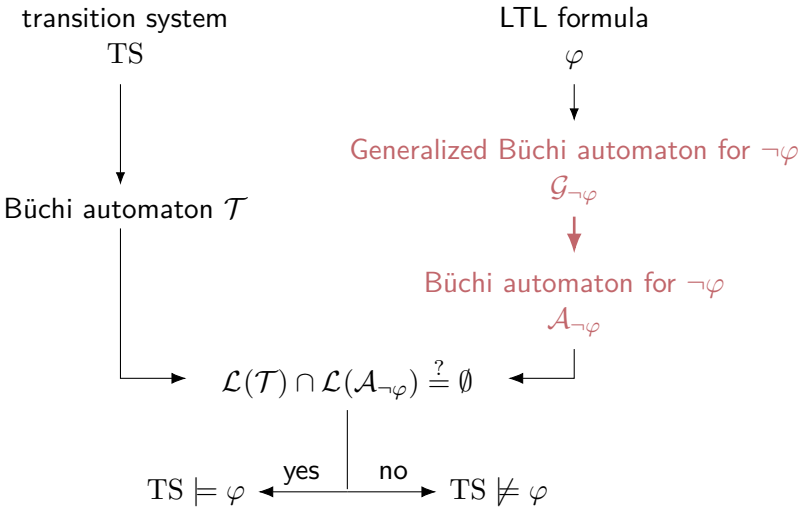


Example: $\varphi = a \ U \ b$



- note: labels of transitions are the atomic propositions in source states

Overview


transition system
TS
LTL formula
φ
Generalized Büchi automaton for $\neg\varphi$
$\mathcal{G}_{\neg\varphi}$
Büchi automaton $\mathcal{T}$
Büchi automaton for $\neg\varphi$
$\mathcal{A}_{\neg\varphi}$
$\mathcal{L}(\mathcal{T}) \cap \mathcal{L}(\mathcal{A}_{\neg\varphi}) \stackrel{?}{=} \emptyset$
yes
no
$\mathrm{TS} \models \varphi$
$\mathrm{TS} \not\models \varphi$

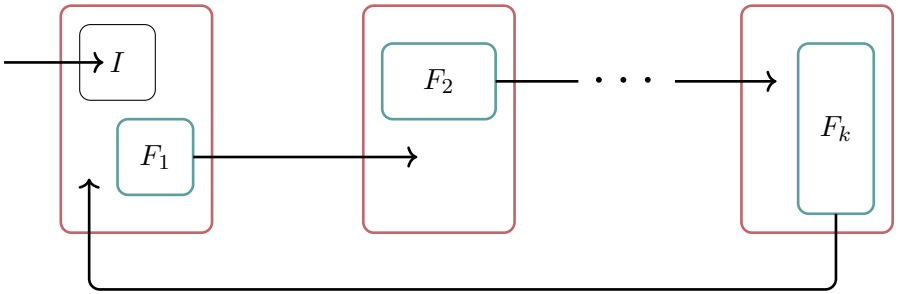
GBA to BA (idea)

- idea: chain automata multiple times together by creating as many copies as final sets
- Let $\mathcal{G} = (\boldsymbol{Q}, I, \Sigma, \mathcal{F}, \delta)$ with $\mathcal{F} = \{F_1, \ldots, F_k\}$ for $k \geq 1$ be an GBA.



GBA to BA (idea)

- idea: chain automata multiple times together by creating as many copies as final sets
- Let $\mathcal{G} = (\boldsymbol{Q}, I, \Sigma, \mathcal{F}, \delta)$ with $\mathcal{F} = \{F_1, \ldots, F_k\}$ for $k \geq 1$ be an GBA.



GBA to BA (formally)

Definition

Let $\mathcal{G} = (Q, I, \Sigma, \mathcal{F}, \delta)$ with $\mathcal{F} = \{F_1, \ldots, F_k\}$ for $k \geq 1$ be an GBA. We define the equivalent BA $\mathcal{A} = (Q', I', \Sigma, F', \delta')$ with

- $Q' = Q \times \{1, \ldots, k\}$
- $I' = I \times \{1\} = \{\langle q_0, 1\rangle \mid q_o \in Q\}$
- $F' = F_1 \times \{1\} = \{\langle q_F, 1\rangle \mid q_F \in F_1\}$
- $$\delta'(\langle q, i\rangle, A) = \begin{cases} \{\langle q', i\rangle \mid q' \in \delta(q, A)\} & \text{if } q \notin F_i \\ \{\langle q', i+1\rangle \mid q' \in \delta(q, A)\} & \text{if } q \in F_i \text{ and } i < k \\ \{\langle q', 1\rangle \mid q' \in \delta(q, A)\} & \text{if } q \in F_k \end{cases}$$

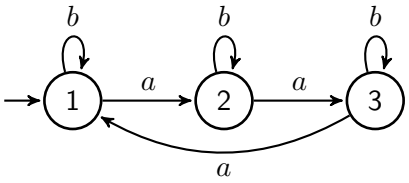


Example: GBA to BA

$\mathcal{A}$

$\mathcal{G}$

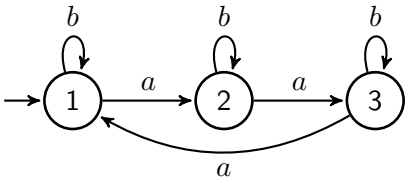
$\mathcal{F} = \{\{1\}, \{2, 3\}\}$



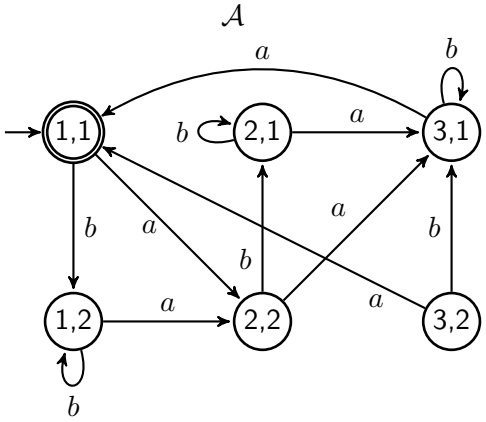
Example: GBA to BA

$\mathcal{G}$

$\mathcal{F} = \{\{1\}, \{2, 3\}\}$



$\mathcal{A}$



Complexity analysis: Size of $\mathcal{A}\varphi$

- number of subformulas of φ is linear in formula length
  $\Rightarrow |closure(\varphi)| \in O(|\varphi|)$
- maximal consistent sets are subsets of $closure(\varphi)$
  $\Rightarrow$ number of states is in $2^{O(|\varphi|)}$
- there are at most $|\varphi|$ many acceptance sets
  $\Rightarrow$ transforming the GBA to BA results in an automaton with at most $|\varphi| \cdot 2^{O(|\varphi|)} = 2^{O(|\varphi|)}$ many states.
- therefore: size of BA (and computation time) is exponential in length of formula
- it can be proven, that the exponential growth is unavoidable



Expressiveness: LTL vs. ω-regular languages

Temporal property

p is always false after an *odd* number of steps



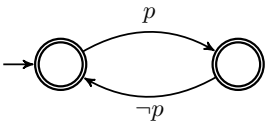
Expressiveness: LTL vs. ω-regular languages

Temporal property

p is always false after an *odd* number of steps

- naive (wrong) approach

$$p \wedge \Box(p \rightarrow \bigcirc\neg p) \wedge \Box(\neg p \rightarrow \bigcirc p)$$





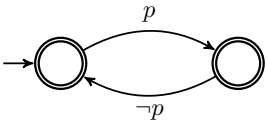
Expressiveness: LTL vs. ω-regular languages

Temporal property

p is always false after an *odd* number of steps

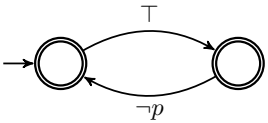
- naive (wrong) approach

$$p \wedge \Box(p \rightarrow \bigcirc\neg p) \wedge \Box(\neg p \rightarrow \bigcirc p)$$

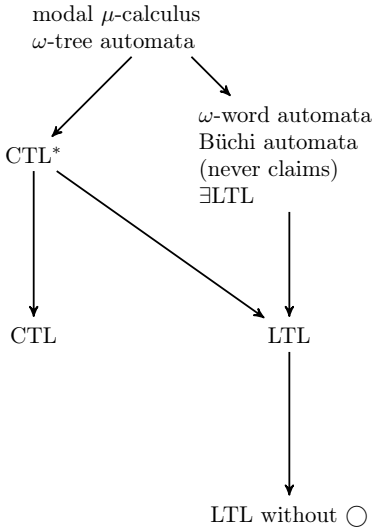


- correct approach (outside LTL)

$$\exists t.\ t \wedge \Box(t \rightarrow \bigcirc\neg t) \wedge \Box(\neg t \rightarrow \bigcirc t) \wedge \Box(\neg t \rightarrow \neg p)$$

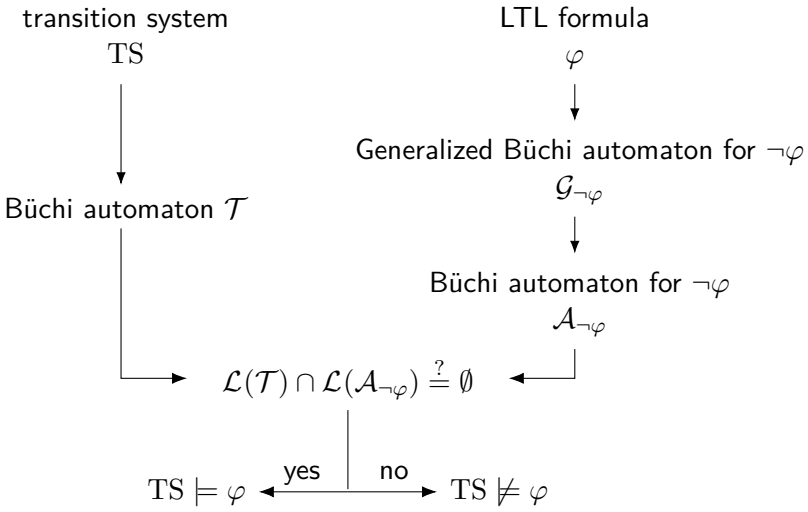


Expressiveness





Overview


transition system
TS
LTL formula
φ
Generalized Büchi automaton for $\neg\varphi$
$\mathcal{G}_{\neg\varphi}$
Büchi automaton $\mathcal{T}$
Büchi automaton for $\neg\varphi$
$\mathcal{A}_{\neg\varphi}$
$\mathcal{L}(\mathcal{T}) \cap \mathcal{L}(\mathcal{A}_{\neg\varphi}) \stackrel{?}{=} \emptyset$
yes
no
$\mathrm{TS} \models \varphi$
$\mathrm{TS} \not\models \varphi$



Summary: Automata-based LTL model checking

- translate system to automaton
- translate LTL property to automaton
  - LTL to GBA
  - GBA to BA
- BAs are more expressive than LTL
- next: operations on automaton for
  - language intersection
  - (efficient) emptiness check



References I

Bibliography

