

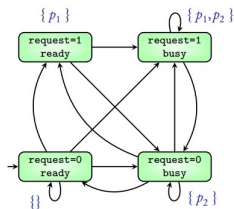
Automata on Infinite words and LTL Model Checking

Rodica Condurache

Lecture 5

Labeled Transition Systems

- Let \mathcal{AP} be the (finite) set of **atomic propositions**
- $2^{\mathcal{AP}}$ is the (finite) **alphabet**



Definition

A **Labeled Transition System (LTS)** is a tuple $\mathcal{M} = \langle \mathcal{AP}, S, S_0, \mathcal{R}, L \rangle$ where

- \mathcal{AP} is the set of labels (atomic propositions)
- S is the finite set of states
- $S_0 \in S$ is the set of initial states
- $\mathcal{R} \subseteq S \times S$ is the transition relation
- $L : S \rightarrow 2^{\mathcal{AP}}$ is the labeling function (**each state is labeled with a set of propositions!**)

Labeled Transition Systems

- A (finite or infinite) **run** ρ in \mathcal{M} is a sequence $\rho = s_0s_1s_2\dots$ where
 - $s_0 \in S_0$ is an initial state of \mathcal{M}
 - $\forall i \geq 0, (s_i, s_{i+1}) \in \mathcal{R}$
- For ρ a run in \mathcal{M} , **trace**(ρ) = $L(s_0)L(s_1)L(s_2)\dots$
- $\text{Traces}(\mathcal{M}) = \{\text{trace}(\rho) \mid \rho \text{ a run in } \mathcal{M}\}$ is the set of **traces** of \mathcal{M}

- Use Regular Expressions to express properties for finite runs (see LFA course)
- Linear-time Temporal Logic(LTL) can express properties on infinite runs

Reachability

The most basic linear property we may want to check: **Reachability**

Definition (Reachability problem)

The reachability problem is defined as follows

Problem: Reachability

Input: A Labeled Transition System \mathcal{M} , and two states s and s_f .

Question: Does there exist a finite run in \mathcal{M} starting from s and ending in s_f ?

Theorem

The reachability problem in transition systems is decidable in deterministic polynomial time. It is NLOGSPACE-complete.

Theorem

The reachability problem in transition systems is decidable in deterministic polynomial time.

Proof.

Deterministic Polynomial time algorithm:

- Let $A \subseteq S$ and

$$Pre(A) = \{s \in S \mid \exists t \in A \text{ s.t. } (s, t) \in R\}$$

- Pre is non decreasing: if $A \subseteq A'$, then $Pre(A) \subseteq Pre(A')$.
- Let build the sequence $C_0 = \emptyset$; $C_{n+1} = Pre(\{s_f\} \cup C_n)$
 - $\forall n, \forall t \in C_n$, there is a path from t to s_f
 - $\forall n, C_n \subseteq C_{n+1} \subseteq S \rightarrow \exists p \text{ s.t. } C_p = C_{p+1} = Pre^*(s_f)$ (the sequence converges)
- The fixpoint is reached after at most $|S|$ iterations \rightarrow polynomial time.



Linear-time Temporal Logic - Syntax

- LTL = propositional calculus + temporal extension
- Temporal operators: X ("next"); U ("until")

Definition (LTL syntax)

Given a set \mathcal{AP} of atomic propositions, a LTL formula over \mathcal{AP} is defined by the following syntax:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid X\varphi \mid \varphi U \varphi$$

where $p \in \mathcal{AP}$.

- We can define the following macros:

$$\varphi_1 \wedge \varphi_2 = \neg(\neg\varphi_1 \vee \neg\varphi_2) \quad (\varphi_1 \text{ and } \varphi_2)$$

$$\varphi_1 \rightarrow \varphi_2 = \neg\varphi_1 \vee \varphi_2 \quad (\varphi_1 \text{ implies } \varphi_2)$$

$$\varphi_1 \leftrightarrow \varphi_2 = (\varphi_1 \rightarrow \varphi_2) \wedge (\varphi_2 \rightarrow \varphi_1) \quad (\varphi_1 \text{ equivalent to } \varphi_2)$$

$$F\varphi = \text{true } U \varphi \quad (\text{eventually } \varphi)$$

$$G\varphi = \neg F\neg\varphi \quad (\text{always } \varphi)$$

$$\varphi_1 R \varphi_2 = G\varphi_2 \vee \varphi_2 U (\varphi_1 \wedge \varphi_2) \quad (\varphi_1 \text{ releases } \varphi_2)$$

Linear-time Temporal Logic - Semantic

- LTL formulas may be **evaluated over infinite words** $w = w_0 w_1 w_2 \dots \in (2^{\mathcal{A}\mathcal{P}})^\omega$

Definition

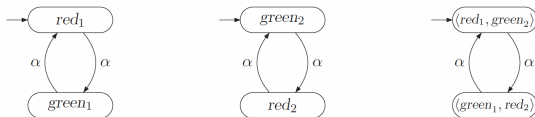
LTL Semantics Given a word $w = w_0 w_1 w_2 \dots \in (2^{\mathcal{A}\mathcal{P}})^\omega$ and a position $i \geq 0$,

- $w, i \models p$ iff $p \in w_i$
- $w, i \models \neg\varphi$ iff $w, i \not\models \varphi$
- $w, i \models \varphi_1 \vee \varphi_2$ iff $w, i \models \varphi_1$ or $w, i \models \varphi_2$
- $w, i \models X\varphi$ iff $w, i+1 \models \varphi$
- $w, i \models \varphi_1 \mathcal{U} \varphi_2$ iff $\exists j \geq i$ s.t. $w, j \models \varphi_2$ and $w, k \models \varphi_1$ for all $i \leq k < j$

- The **language of φ** : $\mathcal{L}(\varphi) = \{w \in (2^{\mathcal{A}\mathcal{P}})^\omega \mid w, 0 \models \varphi\}$

Linear Temporal Logic : Example - Traffic Light

We consider two fully synchronized traffic lights (left and middle) and their parallel composition(right).



Desirable properties:

“The first traffic light is infinitely often green”.

This property corresponds to the set of infinite words $s_0s_1s_2\dots$ such that $green_1 \in L(s_i)$ for infinitely many i .

$$\begin{aligned} & \{ red_1, green_2 \} \{ green_1, red_2 \} \{ red_1, green_2 \} \{ green_1, red_2 \} \dots, \\ & \emptyset \{ green_1 \} \emptyset \{ green_1 \} \emptyset \{ green_1 \} \emptyset \{ green_1 \} \emptyset \dots \\ & \{ red_1, green_1 \} \{ red_1, green_1 \} \{ red_1, green_1 \} \{ red_1, green_1 \} \dots \quad \text{and} \\ & \{ green_1, green_2 \} \{ green_1, green_2 \} \{ green_1, green_2 \} \{ green_1, green_2 \} \dots \end{aligned}$$

The word $\{ red_1, green_1 \} \{ red_1, green_1 \} \emptyset \emptyset \emptyset \dots$ does not correspond to specification.

LTL Model checking

- Verify that \mathcal{M} satisfies LTL formula φ :

$$\text{Traces}(\mathcal{M}) \subseteq \mathcal{L}(\varphi)$$

$$\equiv$$

$$\text{Traces}(\mathcal{M}) \cap \mathcal{L}(\neg\varphi) = \emptyset$$

- Use **automata** to encode the language of φ

We build an automaton \mathcal{A}_φ s.t. \mathcal{A}_φ accepts w iff $w \in \mathcal{L}(\varphi)$

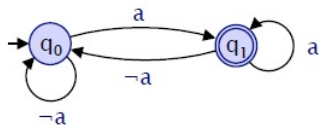
Nondeterministic Büchi word automata (NBA)

Definition (Nondeterministic Büchi word automata)

A Nondeterministic Büchi automaton accepting words over 2^{A^P} is a tuple

$\mathcal{A} = \langle 2^{A^P}, Q, Q_0, \delta, T \rangle$ where

- 2^{A^P} is the alphabet
- Q is the set of states
- $Q_0 \subseteq Q$ is the set of initial states
- $\delta \subseteq Q \times 2^{A^P} \times Q$ is the transition relation
- $T \subseteq Q$ is the set of accepting states

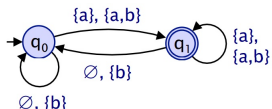


- i.e. just like a nondeterministic finite automaton (NFA) (see LFA)
- The difference is the accepting condition ...

Runs of a NBA

- Consider a Büchi automaton $\mathcal{A} = \langle 2^{AP}, Q, Q_0, \delta, T \rangle$
- A **run** of \mathcal{A} on an infinite word $w = w_0w_1w_2\dots$ is an infinite sequence $q_0q_1q_2\dots \in Q^\omega$ s.t.
 - $q_0 \in Q_0$ is an initial state of \mathcal{A} and $(q_i, w_i, q_{i+1}) \in \delta$ for all $i \geq 0$

Example



$$w = (\{a\}\{b\}\{b\})^\omega$$

$$\rho = q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_0 \xrightarrow{b} q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_0 \dots$$

- Let $\text{inf}(\rho)$ be the set of states that appear infinitely often in ρ :

$$\text{inf}(\rho) = \{q \mid \forall i \geq 0, \exists j \geq i \text{ s.t. } \rho(i) = q\}$$

- An **accepting run** is a run with $q_i \in T$ infinitely often : $\text{inf}(\rho) \cap T \neq \emptyset$

Example

$\rho = q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_0 \xrightarrow{b} q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_0 \xrightarrow{b} q_0 \xrightarrow{a} q_1 \dots$ on $w = (\{a\}\{b\}\{b\})^\omega$ is accepting

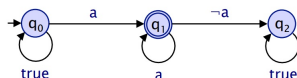
$\rho' = q_0 \xrightarrow{b} q_0 \xrightarrow{b} q_0 \xrightarrow{b} q_0 \dots$ on $w = (\{b\})^\omega$ is not accepting in \mathcal{A} from above

Language of a NBA

- A word w is **accepted** by a NBW \mathcal{A} iff there **exists** an **accepting run** on w in \mathcal{A}

Example (Eventually Globally a (FGa))

For $\mathcal{AP} = \{a, b\}$,



- For $w = \{a\}\{b\}\{b\}(\{a\})^\omega$,
 - the run $\rho = (q_0)^\omega$ is not accepting
 - **but** $\rho' = q_0 q_0 q_0 q_0 (q_1)^\omega$ is **accepting** and therefore w is accepted
- For $w = (\{a\}\{b\}\{b\})^\omega$,
 - the possible runs are $\rho = (q_0)^* q_1 (q_2)^\omega$ or $\rho' = (q_0)^\omega$
 - w is not accepted
- The **language** $\mathcal{L}(\mathcal{A})$ of \mathcal{A} is the **set of words accepted** by the automaton \mathcal{A}
- A set L of words is **Büchi recognizable** if there is a Büchi automaton \mathcal{A} s.t. $\mathcal{L}(\mathcal{A}) = L$.

Büchi-recognizable languages are **closed under Union, Intersection and Complement**:

Given two Büchi automata $\mathcal{A}_1 = \langle 2^{\mathcal{A}P}, Q_1, Q_0^1, \delta_1, T_1 \rangle$ and $\mathcal{A}_2 = \langle 2^{\mathcal{A}P}, Q_2, Q_0^2, \delta_2, T_2 \rangle$
We can define

- **Union:** $\mathcal{A}_U = \langle 2^{\mathcal{A}P}, Q', Q_0', \delta', T' \rangle$ such that $\mathcal{L}(\mathcal{A}_U) = \mathcal{L}(\mathcal{A}_1) \cup \mathcal{L}(\mathcal{A}_2)$
- **Intersection:** $\mathcal{A}_\cap = \langle 2^{\mathcal{A}P}, Q', Q_0', \delta', T' \rangle$ such that $\mathcal{L}(\mathcal{A}_\cap) = \mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2)$
- **Complement:** $\bar{\mathcal{A}}_1 = \langle 2^{\mathcal{A}P}, Q', Q_0', \delta', T' \rangle$ such that $\mathcal{L}(\bar{\mathcal{A}}_1) = \overline{\mathcal{L}(\mathcal{A}_1)}$
 - Difficult to complement Büchi automata (Safra's construction)
 - But, if $\mathcal{L}(\mathcal{A}_1) = \mathcal{L}(\varphi)$ for some LTL formula, $\overline{\mathcal{L}(\mathcal{A}_1)} = \mathcal{L}(\neg\varphi)$

Build directly the automaton for $\neg\varphi$! (if we know φ)

NBA - Closure Properties: Union

Given two Büchi automata $\mathcal{A}_1 = \langle 2^{A^P}, Q_1, Q_0^1, \delta_1, T_1 \rangle$ and $\mathcal{A}_2 = \langle 2^{A^P}, Q_2, Q_0^2, \delta_2, T_2 \rangle$

We define $\mathcal{A}_U = \langle 2^{A^P}, Q', Q_0', \delta', T' \rangle$

- $Q' = Q_1 \cup Q_2$ (we can assume $Q_1 \cap Q_2 = \emptyset$)
- $Q_0' = Q_0^1 \cup Q_0^2$
- $\delta' = \delta_1 \cup \delta_2$
- $T' = T_1 \cup T_2$

Theorem

$$\mathcal{L}(\mathcal{A}_U) = \mathcal{L}(\mathcal{A}_1) \cup \mathcal{L}(\mathcal{A}_2)$$

Proof.

$\mathcal{L}(\mathcal{A}_U) \subseteq \mathcal{L}(\mathcal{A}_1) \cup \mathcal{L}(\mathcal{A}_2)$: For $w \in \mathcal{L}(\mathcal{A}_U)$, $\exists \rho = q_0 q_1 q_2 \dots$ accepting run on w
if $q_0 \in Q_1$, ρ is accepting in $\mathcal{A}_1 \Rightarrow w \in \mathcal{L}(\mathcal{A}_1)$
otherwise, $q_0 \in Q_2$ and ρ is accepting in $\mathcal{A}_2 \Rightarrow w \in \mathcal{L}(\mathcal{A}_2)$

$\mathcal{L}(\mathcal{A}_U) \supseteq \mathcal{L}(\mathcal{A}_1) \cup \mathcal{L}(\mathcal{A}_2)$: For $i \in \{1, 2\}$ and $w \in \mathcal{L}(\mathcal{A}_i)$, $\exists \rho = q_0 q_1 q_2 \dots$ accepting run on w in \mathcal{A}_i

But ρ is also an accepting run in \mathcal{A}_U (in the copy of \mathcal{A}_i) $\Rightarrow w \in \mathcal{L}(\mathcal{A}_U)$



NBA - Closure Properties: Intersection (Special Case)

Given two Büchi automata (note all states of \mathcal{A}_1 are accepting)

$$\mathcal{A}_1 = \langle 2^{A^P}, Q_1, Q_0^1, \delta_1, Q_1 \rangle \text{ and } \mathcal{A}_2 = \langle 2^{A^P}, Q_2, Q_0^2, \delta_2, T_2 \rangle$$

We define $\mathcal{A}_\cap = \langle 2^{A^P}, Q', Q_0', \delta', T' \rangle$

- $Q' = Q_1 \times Q_2$
- $Q_0' = Q_0^1 \times Q_0^2$
- $((q_1, q_2), a, (q_1', q_2')) \in \delta'$ iff $(q_1, a, q_1') \in \delta_1$ and $(q_2, a, q_2') \in \delta_2$
- $T' = Q_1 \times T_2$

Theorem

$$\mathcal{L}(\mathcal{A}_\cap) = \mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2)$$

Proof.

- $\rho' = (q_1^0, q_2^0)(q_1^1, q_2^1)(q_1^2, q_2^2)\dots$ is a run in \mathcal{A}_\cap on w iff $\rho_1 = q_1^0 q_1^1 q_1^2 \dots$ is a run in \mathcal{A}_1 on w and $\rho_2 = q_2^0 q_2^1 q_2^2 \dots$ is a run in \mathcal{A}_2 on w
- ρ' is accepting iff ρ_1 is accepting and ρ_2 is accepting



NBA - Closure Properties: Intersection (General Case)

Given two Büchi automata

$$\mathcal{A}_1 = \langle 2^{A^P}, Q_1, Q_0^1, \delta_1, T_1 \rangle \text{ and } \mathcal{A}_2 = \langle 2^{A^P}, Q_2, Q_0^2, \delta_2, T_2 \rangle$$

We define $\mathcal{A}_\cap = \langle 2^{A^P}, Q', Q'_0, \delta', T' \rangle$

- T' has to **verify both T_1 and T_2** !
- Key idea: make two copies of the states in $Q_1 \times Q_2$
 - 1st copy: Start here, move from here when reached $T_1 \times Q_2$
 - 2nd copy: wait for $Q_1 \times T_2$ and move to first copy when reached
 - Accept if final states in 2nd copy are seen infinitely often

NBA - Closure Properties: Intersection (General Case)

Given two Büchi automata

$$\mathcal{A}_1 = \langle 2^{A^P}, Q_1, Q_0^1, \delta_1, T_1 \rangle \text{ and } \mathcal{A}_2 = \langle 2^{A^P}, Q_2, Q_0^2, \delta_2, T_2 \rangle$$

We define $\mathcal{A}_\cap = \langle 2^{A^P}, Q', Q_0', \delta', T' \rangle$

- $Q' = Q_1 \times Q_2 \times \{1, 2\}$
- $Q_0' = Q_0^1 \times Q_0^2 \times \{1\}$
- $((q_1, q_2, 1), a, (q_1', q_2', 1)) \in \delta'$ iff $(q_1, a, q_1') \in \delta_1$, $(q_2, a, q_2') \in \delta_2$, and $q_1 \notin T_1$
- $((q_1, q_2, 1), a, (q_1', q_2', 2)) \in \delta'$ iff $(q_1, a, q_1') \in \delta_1$, $(q_2, a, q_2') \in \delta_2$, and $q_1 \in T_1$
- $((q_1, q_2, 2), a, (q_1', q_2', 2)) \in \delta'$ iff $(q_1, a, q_1') \in \delta_1$, $(q_2, a, q_2') \in \delta_2$, and $q_2 \notin T_2$
- $((q_1, q_2, 2), a, (q_1', q_2', 1)) \in \delta'$ iff $(q_1, a, q_1') \in \delta_1$, $(q_2, a, q_2') \in \delta_2$, and $q_2 \in T_2$
- $T' = \{(q_1, q_2, 2) \mid q_1 \in Q_1 \text{ and } q_2 \in T_2\}$

Theorem

$$\mathcal{L}(\mathcal{A}_\cap) = \mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2)$$

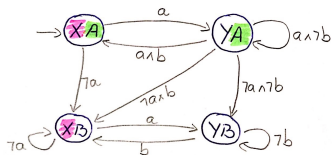
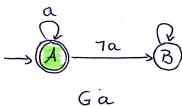
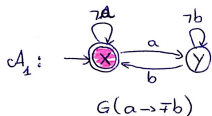
Proof.

same as in the Special Case



NBA - Closure Properties: Intersection

Example

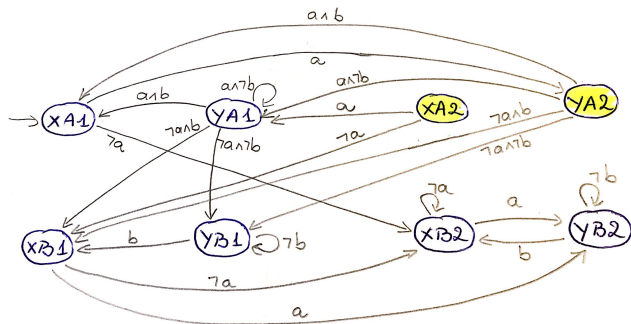


Has to satisfy both

$$T_1 = \{X\} \text{ and } T_2 = \{A\}$$

NBA - Closure Properties: Intersection

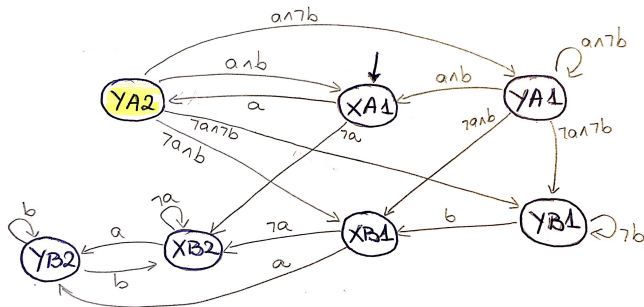
Example



NBA - Closure Properties: Intersection

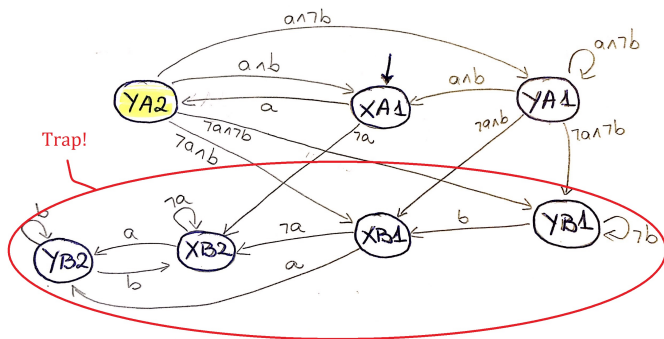
Example

Simplification: Remove unreachable states and moving nodes



NBA - Closure Properties: Intersection

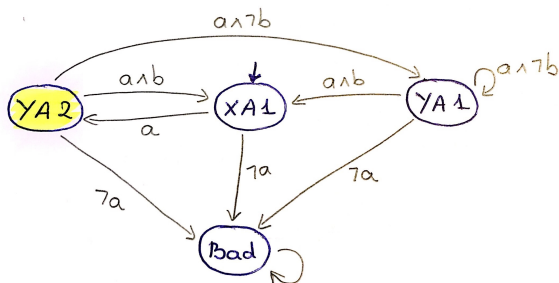
Example



NBA - Closure Properties: Intersection

Example

Simplification: Unify the nodes in the trap

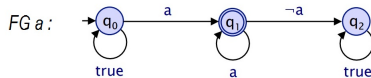
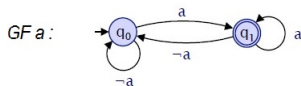


All simplified automata accept the same language : $G(a \wedge Fb)$!

Theorem

For every LTL formula φ over \mathcal{AP} , there is a NBA \mathcal{A} such that

$$\mathcal{L}(\mathcal{A}) = \{w \in 2^{\mathcal{AP}} \mid w, 0 \models \varphi\}$$



The construction of a NBA from a LTL formula is done in three steps:

- **Formula rewriting**
 - Rewrite the formula in **negative normal form**
 - Apply rewriting rules (equivalences)
- **Core translation**
 - Turn an LTL formula into a **generalized Büchi automaton**
- **Degeneralization**
 - Turn the general Büchi automaton into a NBA

LTL to NBA - Rewriting

- Put the formula in **Negative Normal Form**
- Negation appears **only in front of literals**
- Use the following identities to **propagate the negations inwards**:

$$\neg\neg\varphi \equiv \varphi$$

$$\neg X\varphi \equiv X\neg\varphi$$

$$\neg G\varphi \equiv F\neg\varphi$$

$$\neg F\varphi \equiv G\neg\varphi$$

$$\neg(\varphi_1 \vee \varphi_2) \equiv (\neg\varphi_1) \wedge (\neg\varphi_2)$$

$$\neg(\varphi_1 \wedge \varphi_2) \equiv (\neg\varphi_1) \vee (\neg\varphi_2)$$

$$\neg(\varphi_1 \mathcal{U} \varphi_2) \equiv (\neg\varphi_1) \mathcal{R} (\neg\varphi_2)$$

$$\neg(\varphi_1 \mathcal{R} \varphi_2) \equiv (\neg\varphi_1) \mathcal{U} (\neg\varphi_2)$$

Definition

An LTL formula is in **Negative Normal Form (NNF)** if it follows the syntax given by

$$\varphi ::= \top \mid \perp \mid p \mid \neg p \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid X\varphi \mid \varphi \mathcal{U} \varphi \mid \varphi \mathcal{R} \varphi$$

where $p \in \mathcal{AP}$ is an atomic proposition.

- The size of the automaton will depend on the size of the formula
- Reduce the number of temporal subformulas by applying other rewriting rules:

$$(X\varphi_1) \wedge (X\varphi_2) \equiv X(\varphi_1 \wedge \varphi_2)$$

$$(\varphi \mathcal{R} \psi_1) \wedge (\varphi \mathcal{R} \psi_2) \equiv \varphi \mathcal{R} (\psi_1 \wedge \psi_2)$$

$$(G\varphi_1) \wedge (G\varphi_2) \equiv G(\varphi_1 \wedge \varphi_2)$$

$$(X\varphi_1) \mathcal{U} (X\varphi_2) \equiv X(\varphi_1 \mathcal{U} \varphi_2)$$

$$(\psi_1 \mathcal{R} \varphi) \vee (\psi_2 \mathcal{R} \varphi) \equiv (\psi_1 \vee \psi_2) \mathcal{R} \varphi$$

$$GF\varphi_1 \vee GF\varphi_2 \equiv GF(\varphi_1 \vee \varphi_2)$$

LTL to NBA - Rewriting : Example

$$\varphi_1 = \neg F(p \wedge \neg Fq) \equiv G\neg(p \wedge \neg Fq)$$

LTL to NBA - Rewriting : Example

$$\begin{aligned}\varphi_1 = \neg F(p \wedge \neg Fq) &\equiv G\neg(p \wedge \neg Fq) \\ &\equiv G(\neg p \vee \neg\neg Fq)\end{aligned}$$

LTL to NBA - Rewriting : Example

$$\begin{aligned}\varphi_1 = \neg F(p \wedge \neg Fq) &\equiv G\neg(p \wedge \neg Fq) \\ &\equiv G(\neg p \vee \neg\neg Fq) \\ &\equiv G(\neg p \vee Fq)\end{aligned}$$

LTL to NBA - Rewriting : Example

$$\begin{aligned}\varphi_1 = \neg F(p \wedge \neg Fq) &\equiv G\neg(p \wedge \neg Fq) \\ &\equiv G(\neg p \vee \neg\neg Fq) \\ &\equiv G(\neg p \vee Fq)\end{aligned}$$

$$\begin{aligned}\varphi_2 = \neg F(p \wedge (Xq \mathcal{R} X\neg r)) &\equiv G\neg(p \wedge (Xq \mathcal{R} X\neg r)) \\ &\equiv G(\neg p \vee \neg(Xq \mathcal{R} X\neg r)) \\ &\equiv G(\neg p \vee ((\neg Xq) \mathcal{U} (\neg X\neg r))) \\ &\equiv G(\neg p \vee ((X\neg q) \mathcal{U} (X\neg\neg r))) \\ &\equiv G(\neg p \vee (X\neg q) \mathcal{U} (Xr)) \\ &\equiv G(\neg p \vee X(\neg q \mathcal{U} r))\end{aligned}$$

LTL to NBA - Core Translation

- A state of the automaton \mathcal{A}_φ is a consistent set Z of subformulas of φ

Definition

A set $Z \subseteq \text{Sub}(\varphi)$ is consistent if it does not contain \perp or a pair $\{\psi, \neg\psi\}$.

- The formulas in Z are seen as obligations
 - If a run ρ on a word w starts in Z and satisfies the accepting condition, then

$$w, 0 \models \bigwedge_{\psi \in Z} \psi$$

- The only initial state of \mathcal{A}_φ is $Z = \{\varphi\}$
- Transitions to next states are given by the formulas of the form $X\psi$ from Z
- Need to reduce Z such that all formulas in Z are either literals or have the form $X\psi$

LTL to NBA - Core Translation : Reduction of sets Z

- Use ϵ -transitions to reduce arbitrary sets Y of formulas
 - they are handy, but will not belong to the final \mathcal{A}_φ
- Reduction depends on "non-reduced" formulas $\psi \in Y$

$$\text{If } \psi = \psi_1 \wedge \psi_2: \quad Y \xrightarrow{\epsilon} Y \setminus \{\psi\} \cup \{\psi_1, \psi_2\}$$

$$\begin{aligned} \text{If } \psi = \psi_1 \vee \psi_2: \quad & Y \xrightarrow{\epsilon} Y \setminus \{\psi\} \cup \{\psi_1\} \\ & Y \xrightarrow{\epsilon} Y \setminus \{\psi\} \cup \{\psi_2\} \end{aligned}$$

$$\begin{aligned} \text{If } \psi = \psi_1 \text{ R } \psi_2: \quad & Y \xrightarrow{\epsilon} Y \setminus \{\psi\} \cup \{\psi_1, \psi_2\} \\ & Y \xrightarrow{\epsilon} Y \setminus \{\psi\} \cup \{\psi_2, X\psi\} \end{aligned}$$

$$\text{If } \psi = G \psi_2: \quad Y \xrightarrow{\epsilon} Y \setminus \{\psi\} \cup \{\psi_2, X\psi\}$$

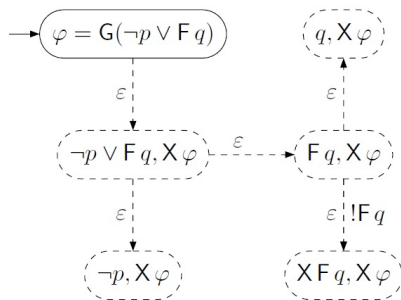
$$\begin{aligned} \text{If } \psi = \psi_1 \text{ U } \psi_2: \quad & Y \xrightarrow{\epsilon} Y \setminus \{\psi\} \cup \{\psi_2\} \\ & Y \xrightarrow{\epsilon, !\psi} Y \setminus \{\psi\} \cup \{\psi_1, X\psi\} \end{aligned}$$

$$\begin{aligned} \text{If } \psi = F \psi_2: \quad & Y \xrightarrow{\epsilon} Y \setminus \{\psi\} \cup \{\psi_2\} \\ & Y \xrightarrow{\epsilon, !\psi} Y \setminus \{\psi\} \cup \{X\psi\} \end{aligned}$$

- $!\psi$ means " ψ has been postponed"
- marked transitions used to define accepting transitions

LTL to NBA - Core Translation : Example

Example (Reduction for $\varphi = G(p \rightarrow Fq)$)



LTL to NBA - Core Translation

- $Y \xrightarrow[*]{\epsilon} Z$ if there is a sequence of ϵ -transitions from Y to Z

$$\text{Red}(Y) = \{Z \text{ consistent and reduced} \mid Y \xrightarrow[*]{\epsilon} Z\}$$

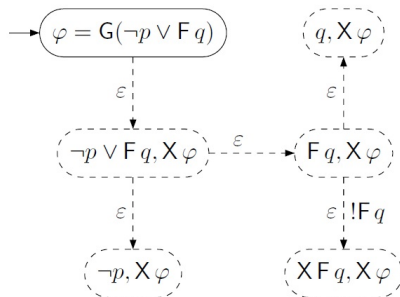
$$\text{Red}_\alpha(Y) = \{Z \text{ consistent and reduced} \mid Y \xrightarrow[*]{\epsilon} Z \text{ without using an edge marked with } !\alpha\}$$

From the definition of the reduction rules, holds:

$$\bigwedge_{\psi \in Y} \psi \equiv \bigvee_{Z \in \text{Red}(Y)} \bigwedge_{\psi \in Z} \psi$$

LTL to NBA - Core Translation : Example

Example (Reduction for $\varphi = G(p \rightarrow Fq)$)



$$\text{Red}(\{\varphi\}) = \{\{\neg p, X\varphi\}, \{q, X\varphi\}, \{X Fq, X\varphi\}\}$$

$$\text{Red}_{Fq}(\{\varphi\}) = \{\{\neg p, X\varphi\}, \{q, X\varphi\}\}$$

LTL to NBA - Core Translation : Generalized Büchi Automaton

- Let $\Sigma_Z = \{a \in 2^{\mathcal{AP}} \mid \forall p \in \mathcal{AP}, (p \in Z \rightarrow p \in a) \text{ and } (\neg p \in Z \rightarrow p \notin a)\}$
- Let $U(\varphi) = \{\psi \in \text{Sub}(\varphi) \mid \psi = \psi_1 \mathcal{U} \psi_2 \text{ or } \psi = F\psi_1\}$ the set of *until formulas* of φ
- Let $\text{next}(Z) = \{\psi \mid X\psi \in Z\}$

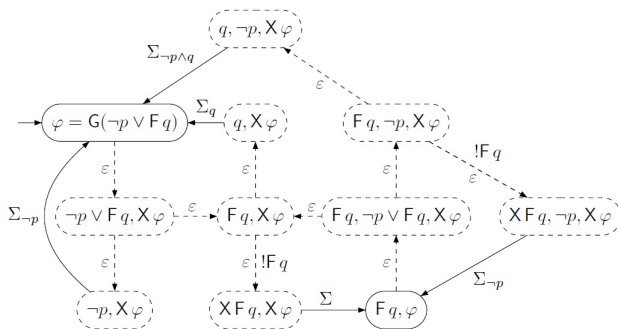
The Generalized Büchi Automaton for φ is $\mathcal{B}_\varphi = \langle 2^{\mathcal{AP}}, Q, Q_0, \delta, (T_\alpha)_{\alpha \in U(\varphi)} \rangle$

- $Q = 2^{\text{Sub}(\varphi)}$
- $Q_0 = \{\{\varphi\}\}$
- $\delta = \{Y \xrightarrow{a} \text{next}(Z) \mid Y \in Q, a \in \Sigma_Z \text{ and } Z \in \text{Red}(Y)\}$
- For each $\alpha \in U(\varphi)$, $T_\alpha = \{Y \xrightarrow{a} \text{next}(Z) \mid Y \in Q, a \in \Sigma_Z \text{ and } Z \in \text{Red}_\alpha(Y)\}$

- the accepting condition is a set of sets of transitions to be visited infinitely often
- Asks to not postpone forever the until formulas

LTL to NBA - Core Translation : Example of Construction

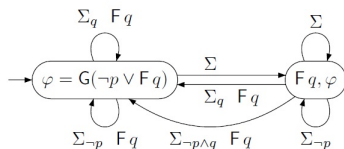
Example ($\varphi = G(\neg p \vee Fq)$)



LTL to NBA - Core Translation : Example of Construction

Example ($\varphi = G(\neg p \vee Fq)$ - continuation)

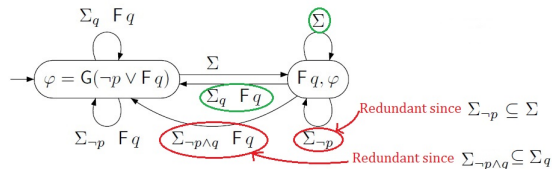
- After removing the intermediate dashed transitions:



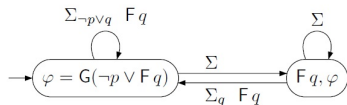
LTL to NBA - Core Translation : Example of Construction

Example ($\varphi = G(\neg p \vee Fq)$ - continuation)

- After removing the intermediate dashed transitions:

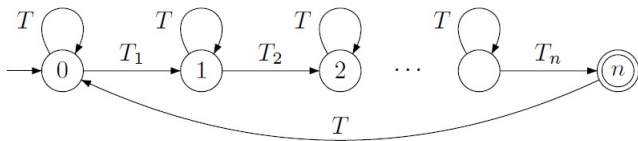


- After removing redundant transitions:



LTL to NBA - Degeneralization

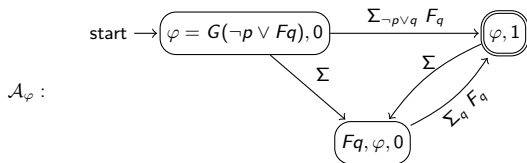
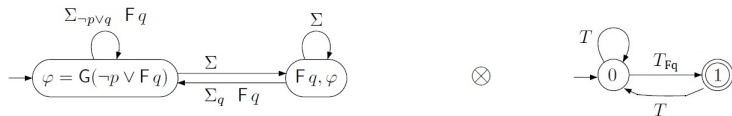
- For $\mathcal{B}_\varphi = \langle 2^{A^P}, Q, Q_0, \delta, T_1, T_2, \dots, T_n \rangle$ with n sets in the acceptance condition,
- Take the **synchronous product** with the automaton \mathcal{D}_n below:



- The Nondeterministic Büchi Automaton for φ is then $\mathcal{A}_\varphi = \mathcal{B}_\varphi \otimes \mathcal{D}_n$

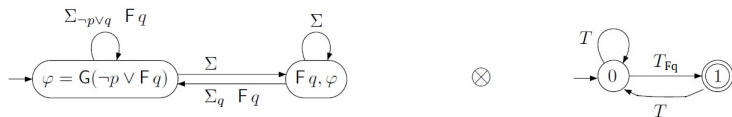
LTL to NBA - Degeneralization : Example

For $\varphi = G(\neg p \vee Fq)$

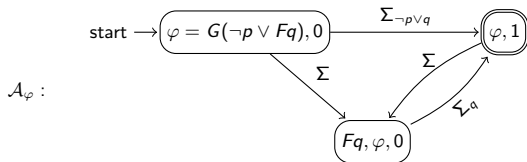


LTL to NBA - Degeneralization : Example

For $\varphi = G(\neg p \vee Fq)$

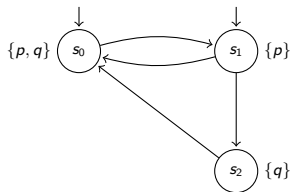


NBA \mathcal{A}_φ after removing labels Fq :

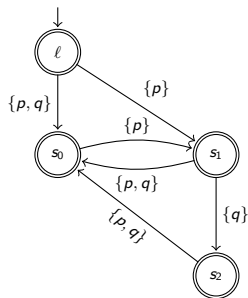


Modeling LTS as NBA

- A Labeled Transition System \mathcal{M} is the set of all its executions
- Transform a LTS $\mathcal{M} = \langle \mathcal{AP}, S, S_0, R, \tau \rangle$
- ... into NBA $\mathcal{A}_{\mathcal{M}} = \langle 2^{\mathcal{AP}}, Q, Q_0, \delta, T \rangle$ where
 - $Q = S \cup \{\ell\}$
 - $Q_0 = \{\ell\}$
 - $(\ell, a, s) \in \delta$ iff $s \in S_0$ and $a = \tau(s)$
 - $(s, a, s') \in \delta$ iff $(s, s') \in R$ and $a = \tau(s')$
 - $T = S \cup \{\ell\}$



LTS \mathcal{M}



NBA $\mathcal{A}_{\mathcal{M}}$

Back to LTL Model Checking

- Recall: \mathcal{M} satisfies the LTL formula φ iff $\text{Traces}(\mathcal{M}) \cap \mathcal{L}(\neg\varphi) = \emptyset$
- Since $\mathcal{L}(\mathcal{A}_{\neg\varphi}) = \mathcal{L}(\neg\varphi)$,

\mathcal{M} satisfies the LTL formula φ iff $\mathcal{L}(\mathcal{A}_{\mathcal{M}} \cap \mathcal{A}_{\neg\varphi}) = \emptyset$

Where

- $\mathcal{A}_{\mathcal{M}}$ is the Büchi automaton of size $\mathcal{O}(|\mathcal{M}|)$ s.t. $\mathcal{L}(\mathcal{A}_{\mathcal{M}}) = \text{Traces}(\mathcal{M})$
 - $\mathcal{A}_{\neg\varphi}$ is the Büchi automaton recognizing models of $\neg\varphi$ obtained as before. Its size is $2^{\mathcal{O}(|\varphi|)}$
-
- If $\mathcal{L}(\mathcal{A}_{\mathcal{M}} \cap \mathcal{A}_{\neg\varphi}) \neq \emptyset$, any behavior in it is an **counterexample**.
 - Counterexamples are always of the form uv^ω , where u and v are finite words

LTL Model Checking - Complexity

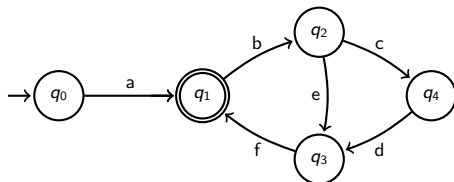
Theorem

Checking whether an LTL formula φ is satisfied by a LTS \mathcal{M} can be done in time $\mathcal{O}(|\mathcal{M}| \times 2^{\mathcal{O}(|\varphi|)})$.

i.e., checking is polynomial in the size of the model and exponential in the size of the specification.

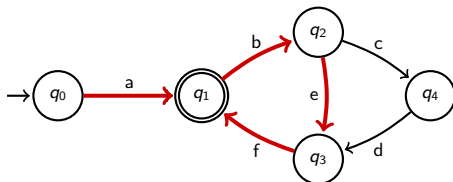
Emptiness of NBA

- An Büchi automaton is **non-empty** iff
there exists a path to a cycle containing an accepting state
- Is this automaton empty?



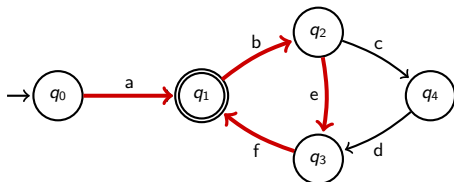
Emptiness of NBA

- An Büchi automaton is **non-empty** iff
there exists a path to a cycle containing an accepting state
- Is this automaton empty?
No : It accepts $a(bef)^\omega$



Emptiness of NBA

- An Büchi automaton is **non-empty** iff
there exists a path to a cycle containing an accepting state
- Is this automaton empty?
No : It accepts $a(bef)^\omega$

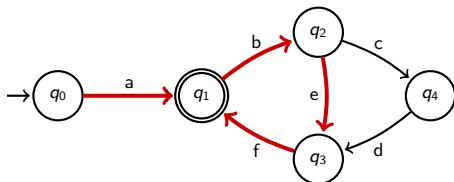


Idea:

- Consider only reachable states of \mathcal{A}
- Find all maximal strongly connected components: SCC_1, SCC_2, \dots .
An automaton is non-empty iff exists SCC_i containing an accepting state

Emptiness of NBA

- An Büchi automaton is **non-empty** iff there exists a path to a cycle containing an accepting state
- Is this automaton empty?
No : It accepts $a(bef)^\omega$



Idea:

- Consider only reachable states of \mathcal{A}
- Find all maximal strongly connected components: SCC_1, SCC_2, \dots .

An automaton is non-empty iff exists SCC_i containing an accepting state

Consequence: The language of any Büchi automata is of the form $X(Y)^\omega$ where X and Y are regular languages of finite words.

- Stéphane Demri & Paul Gastin - *Specification and Verification using Temporal Logics* : <https://pdfs.semanticscholar.org/a2e0/cefb8391242dc412fb1b29edcdc59a13e5df.pdf>
- Bakhadyr Khoushainov and Anil Nerode: *Automata Theory and its Applications* (available online)
- Erich Grädel et al: *Automata, Logics, and Infinite Games - A Guide to Current Research*(available online)

Exercise 1

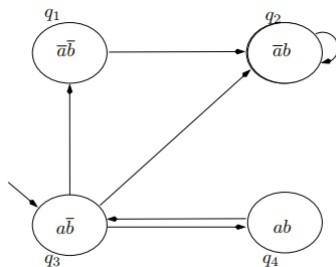
Build one non-deterministic Büchi word automaton for the following formulas:

- $\varphi_1 = FGp$
- $\varphi_2 = G(p \rightarrow X(qUr))$

Build the generalized Büchi word automaton for the formula :

- $\varphi = (G(p \rightarrow q)) \rightarrow G\beta$ where $\alpha = F(p \wedge \neg)$ and $\beta = F(p \wedge Xp)$.
 - i Write the formula in negative normal form
 - ii Draw the reduction graph starting from φ .
 - iii Give the sets $Red(\{\varphi\})$, $Red_\alpha(\{\varphi\})$ and $Red_\beta(\{\varphi\})$.
 - iv Draw the transitions starting from state $\{\varphi\}$ in the GBA \mathcal{A}_φ .
 - v Complete the construction and draw the automaton \mathcal{A}_φ . Indicate clearly the accepting conditions.

Exercise 2



Verify if the above transition system satisfies $a\mathcal{U}X(a \wedge \neg b)$.