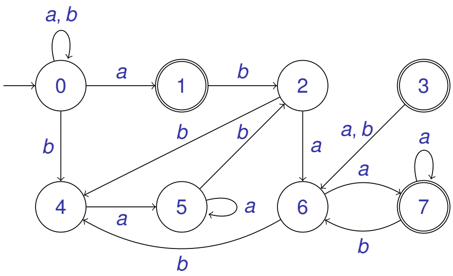
# Automata Theoretic LTL Model Checking

Giuseppe Perelli

SAPIENZA
UNIVERSITÀ DI ROMA

Formal Methods 2020/21

- Nonemptiness problems for NFA and NBA
- Generalized Nondeterministic Büchi Automata
- From LTL to GNBA
- Automata-Theoretic approach for Model Checking LTL




## Question

Given a NFA $\mathcal{N}$, decide whether

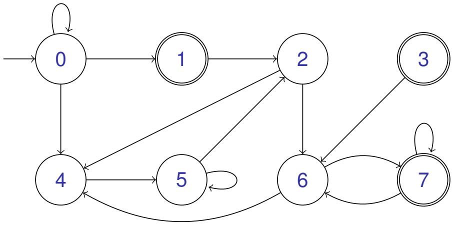
$$\mathcal{L}(\mathcal{N}) \overset{?}{\neq} \emptyset$$

Does a word $w$ accepted by $\mathcal{N}$ exist?

## Observation

A word $w$ is accepted by $\mathcal{N}$ iff there exists a run whose path starts in $0$ and ends in a final state $F$.

## The nonemptiness problem for NFA




**Question**

Given a NFA $\mathcal{N}$, decide whether

$$\mathcal{L}(\mathcal{N}) \stackrel{?}{\neq} \emptyset$$

Does a word $w$ accepted by $\mathcal{N}$ exist?

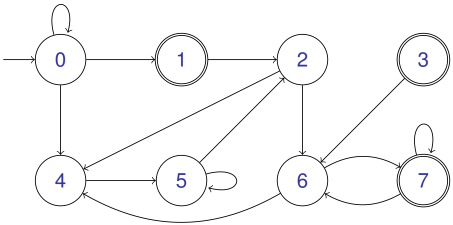
**Observation**

A word $w$ is accepted by $\mathcal{N}$ iff there exists a run whose path starts in $0$ and ends in a final state $F$.

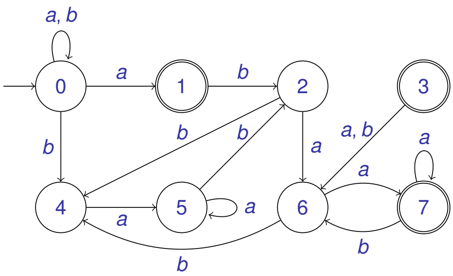
**Solution**

Nonemptiness of NFAs reduces to reachability over graphs.

# Reachability with fix-point theory




$$\mathsf{Reach}(F) = \mu\mathcal{Z}(F \vee \langle\mathsf{next}\rangle\mathcal{Z})$$




### Question

Given a NBA $\mathcal{N}$, decide whether

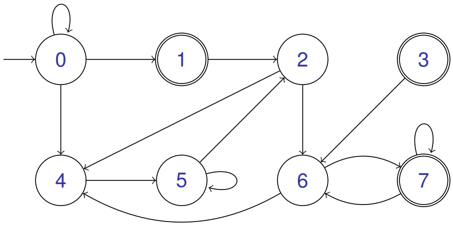
$$\mathcal{L}(\mathcal{N}) \overset{?}{\neq} \emptyset$$

Does a word $w$ accepted by $\mathcal{N}$ exist?

### Observation

A word $w$ is accepted by $\mathcal{N}$ iff there exists a run whose path starts in $0$ and visits a final state in $F$ infinitely many times.

## The nonemptiness problem for NBA




### Question

Given a NBA $\mathcal{N}$, decide whether

$$\mathcal{L}(\mathcal{N}) \stackrel{?}{\neq} \emptyset$$

Does a word $w$ accepted by $\mathcal{N}$ exist?

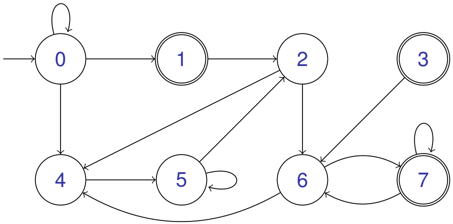
### Observation

A word $w$ is accepted by $\mathcal{N}$ iff there exists a run whose path starts in $0$ and visits a final state in $F$ infinitely many times.

### Solution

Nonemptiness of NBAs reduces to recurrent reachability over graphs.

# Recurrent reachability with fix-point theory




$$\begin{aligned}\mathrm{Buchi}(F) &= \nu\mathcal{Y}.(\mathrm{Reach}(F \wedge \langle \mathrm{next} \rangle \mathcal{Y}))\\ &= \nu\mathcal{Y}.(\mu\mathcal{Z}.((F \wedge \langle \mathrm{next} \rangle \mathcal{Y}) \vee \langle \mathrm{next} \rangle \mathcal{Z}))\end{aligned}$$

A Generalized Nondeterministic Büchi Automaton (GNBA) is a tuple $\mathcal{N} = \langle Q, \Sigma, I, \delta, \mathcal{F} \rangle$ where everything is as for a standard NBA except that

$$\mathcal{F} = (F_1, F_2, \ldots, F_n)$$

A run $\rho$ in $\mathcal{N}$ is accepting iff it visits every $F_i$ infinitely often.

**Theorem**

It holds that $\mathcal{L}(\mathcal{N}) = \mathcal{L}(\mathcal{N}_1 \otimes \ldots \otimes \mathcal{N}_n)$, where $\mathcal{N}_i = \langle Q, \Sigma, q_0, \delta, F_i \rangle$.

# Linear Temporal Logic (LTL)

A standard language for talking about infinite state sequences.

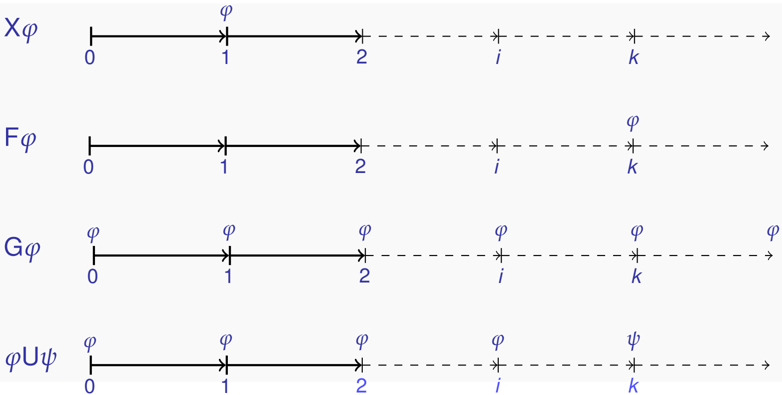
Amir Pnueli - The Temporal Logic of Programs. - FOCS'77

| | | | |
|---|---|---|---|
| $\top$ | truth constant | $\mathsf{X}\phi$ | in the next state... |
| $p$ | primitive propositions | $\mathsf{F}\phi$ | will eventually be the case |
| $\neg\phi$ | classical negation | $\mathsf{G}\phi$ | is always the case |
| $\phi \vee \psi$ | classical disjunction | $\phi\mathsf{U}\psi$ | $\phi$ until $\psi$ |
| $\phi \wedge \psi$ | classical conjunction | $\phi\mathsf{R}\psi$ | $\phi$ release $\psi$ |

Minimal syntax

$$\varphi := p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathsf{X}\varphi \mid \varphi\mathsf{U}\varphi$$

| | |
|---|---|
| Eventually I will graduate | $\mathbf{F}\,\texttt{degree}$ |
| The plane will never crash | $\mathbf{G}\neg\texttt{crash}$ |
| I will eat pizza infinitely often | $\mathbf{GF}\,\texttt{eatPizza}$ |
| ...and they all lived happily ever after | $\mathbf{FG}\,\texttt{happy}$ |
| We are not friends until you apologise | $(\neg\texttt{friends})\,\mathbf{U}\,\texttt{youApologise}$ |
| Every time it is requested, later a document is printed | $\mathbf{G}(\texttt{print}_{\texttt{send}} \rightarrow \mathbf{F}\,\texttt{print}_{\texttt{exec}})$ |
| The two processes are never active at the same time | $\mathbf{G}(\neg\texttt{proc}_1 \vee \neg\texttt{proc}_2)$ |




LTL formulas are evaluated over infinite words over the alphabet $\Sigma = 2^{\mathrm{Prop}}$, with $\mathrm{Prop}$ be the set of variables occurring in the formula.

The language defined by an LTL formula $\varphi$ is $\mathcal{L}(\varphi) = \{w \in \Sigma^\omega : w \models \varphi\}$.

**Theorem**

For an LTL formula $\varphi$, we can construct a generalized nondeterministic Büchi automaton $\mathcal{N}_\varphi = \langle Q, \Sigma, I, \delta, \mathcal{F} \rangle$ such that $\mathcal{L}(\mathcal{N}_\varphi) = \mathcal{L}(\varphi)$.

We will now look into the details on the construction of $\mathcal{N}_\varphi$.

**Definition (Fischer-Ladner Closure)**

For a given LTL formula $\varphi$, the FS-closure of $\varphi$, denoted $\mathsf{cl}(\varphi)$ is the set of subformulas of $\varphi$ and their negation (where $\neg\neg\psi = \psi$). It is (recursively) defined as follows:

- $\varphi \in \mathsf{cl}(\varphi)$
- If $\psi \in \mathsf{cl}(\varphi)$ then $\neg\psi \in \mathsf{cl}(\varphi)$
- If $\psi_1 \wedge \psi_2 \in \mathsf{cl}(\varphi)$ then $\psi_1, \psi_2 \in \mathsf{cl}(\varphi)$
- If $\mathsf{X}\psi \in \mathsf{cl}(\varphi)$ then, $\psi \in \mathsf{cl}(\varphi)$
- If $\psi_1 \mathsf{U} \psi_2 \in \mathsf{cl}(\varphi)$ then $\psi_1, \psi_2 \in \mathsf{cl}(\varphi)$

For example, $\mathsf{cl}(\varphi)$ is $\varphi = p \wedge ((\mathsf{X}p)\mathsf{U}q)$

$$\{p \wedge ((\mathsf{X}p)\mathsf{U}q), \neg(p \wedge ((\mathsf{X}p)\mathsf{U}q)), p, \neg p, (\mathsf{X}p)\mathsf{U}q, \neg((\mathsf{X}p)\mathsf{U}q), \mathsf{X}p, \neg\mathsf{X}p, q, \neg q\}$$

**Atoms**

A set $\alpha \subset \mathrm{cl}(\varphi)$ is called atom if it is maximally consistent, that is:

- For all $\psi \in \mathrm{cl}(\varphi)$ either $\psi \in \alpha$ or $\neg\psi \in \alpha$ (maximality)
- $\psi_1 \wedge \psi_2 \in \alpha$ iff $\psi_1, \psi_2 \in \alpha$ (consistency)

By $\mathrm{Atoms}(\varphi) = \{\alpha \subset \mathrm{cl}(\varphi) : \alpha \text{ is an atom }\}$

The space set of $\mathcal{N}_\varphi$ is defined as $Q = \mathrm{Atoms}(\varphi)$.

Intuitively, a state $\alpha$ in the automaton carries out the information on which subformulas of $\varphi$ need to be satisfied when the computation starts from $\alpha$ itself.

Observation: the size of $\mathcal{N}_\varphi$ is exponential in the length of $\varphi$. Once again, this exponential blow-up is unavoidable.

# Initial states, transition function, and final states of the automaton

## Initial states

Every atom $\alpha$ containing $\varphi$ is an initial state.
$I = \{\alpha \in Q : \varphi \in \alpha\}$

## Transition function

Take two atoms $\alpha$ and $\alpha'$ together with $\sigma \in \Sigma = 2^{\mathrm{Prop}}$.
We say that $\alpha' \in \delta(\alpha, \sigma)$ if

- $\sigma = \alpha \cap \mathrm{Prop}$ (Advance only if you read something consistent)
- $\mathsf{X}\psi \in \alpha$ iff $\psi \in \alpha'$ (Check $\psi$ at the next stage)
- $\psi_1 \mathsf{U} \psi_2 \in \alpha$ iff either $\psi_2 \in \alpha$ or both $\psi_1 \in \alpha$ and $\psi_1 \mathsf{U} \psi_2 \in \alpha'$ (Keep checking $\mathsf{U}$ if needed)

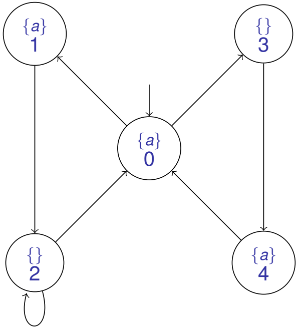
## Final states

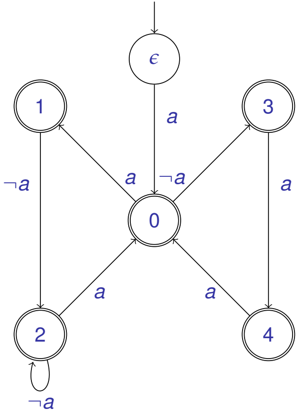
Every subformula $\psi_1 \mathsf{U} \psi_2$ in $\alpha$ holds on acceptance and thus contributes to $\mathcal{F}$ with the set

$$F_{\psi_1 \mathsf{U} \psi_2} = \{\alpha \in Q : \psi_2 \in \alpha \text{ or } \neg(\psi_1 \mathsf{U} \psi_2) \in \alpha\}$$

# Exercise

Build the automaton for the formula $\varphi = p\mathsf{U}q$.




For every labeled transition system $\mathcal{T}$, the automaton $\mathcal{N}_{\mathcal{T}}$ recognizes all and only those infinite words that are generated by $\mathcal{T}$.

### Problem

For a given LTS $\mathcal{T}$ and an LTL formula $\varphi$, Model Checking is the problem of verifying that all the executions of $\mathcal{T}$ satisfy $\varphi$. Equivalently

$$\mathcal{T} \models \varphi \iff \mathcal{L}(\mathcal{T}) \subseteq \mathcal{L}(\varphi)$$

### Automata-Theoretic Approach

- $\mathcal{T} \to \mathcal{N}_\mathcal{T}$ $\qquad \mathcal{L}(\mathcal{T}) = \mathcal{L}(\mathcal{N}_\mathcal{T})$
- $\varphi \to \mathcal{N}_\varphi$ $\qquad \mathcal{L}(\varphi) = \mathcal{L}(\mathcal{N}_\varphi)$
- $\mathcal{L}(\mathcal{T}) \subseteq \mathcal{L}(\varphi) \iff \mathcal{L}(\mathcal{N}_\mathcal{T}) \subseteq \mathcal{L}(\mathcal{N}_\varphi) \iff \mathcal{L}(\mathcal{N}_\mathcal{T}) \cap \mathcal{L}(\overline{\mathcal{N}_\varphi}) = \emptyset$

Ouch! We need to complement a NBA! Can we avoid it? $\qquad \mathcal{L}(\overline{\mathcal{N}_\varphi}) = \mathcal{L}(\mathcal{N}_{\neg\varphi})$