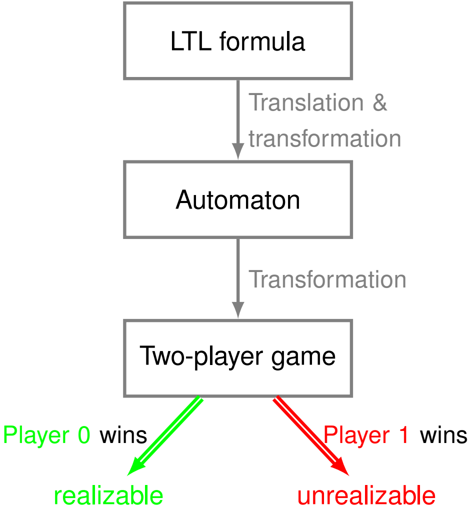
From Temporal Logic Specifications to ω-Automata

Rayna Dimitrova
University of Leicester

Midlands Graduate School 2019

Acknowledgement: Many slides courtesy of Bernd Finkbeiner.

Synthesis from LTL specifications

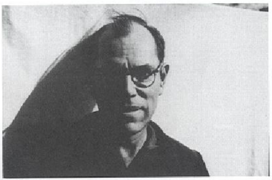


Monday

Tuesday

ω-Automata

> ω-automata recognize
> sets of infinite sequences

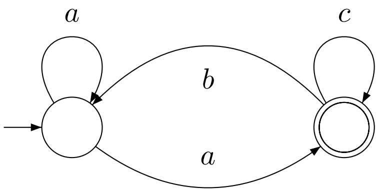


J. Richard Büchi, 1983

automata-theoretic foundation
of the verification of reactive systems

system behavior	$\sim$	infinite sequence of observations
set of behaviors	$\sim$	language of an ω-automaton
verification	$\sim$	“system $\subseteq$ specification ?”
realizability	$\sim$	$\exists$ system:“system $\subseteq$ specification ?”

Nondeterministic finite-word automata (NFA)



A **NFA** $\mathcal{A} = (\Sigma, Q, Q_0, \delta, Q_0, F)$ consists of the following:

- Σ: alphabet
- Q: finite set of states
- $Q_0 \subseteq Q$: initial states
- $\delta : Q \times \Sigma \to 2^Q$: transition function
- $F \subseteq Q$: final states

Acceptance by NFA

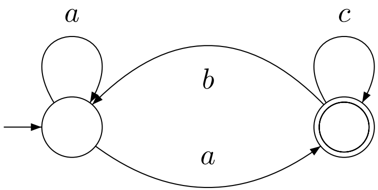
- A **run** of an NFA $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F)$ on an **input sequence** $\sigma_0\sigma_1 \dots \sigma_n \in \Sigma^*$ is a finite sequence of states $q_0\, q_1 \dots q_n$, such that the following conditions are satisfied:
  - $q_0 \in Q_0$ and
  - $q_{i+1} \in \delta(q_i, \sigma_i)q_i$ for all $0 \leq i < n$.
- The run $q_0\, q_1\, q_2 \dots q_n$ is **accepting** iff $q_n \in F$.
- An input sequence is **accepted** by $\mathcal{A}$ iff there exists an accepting run.

The language of $\mathcal{A}$:

$$\mathcal{L}(\mathcal{A}) = \left\{ w \in \Sigma^* \mid w \text{ is accepted by } \mathcal{A} \right\}$$

Two NFAs $\mathcal{A}$ and $\mathcal{A}'$ are **equivalent** iff $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}')$.

ω-Automata



An ω-automaton $\mathcal{A} = (\Sigma, Q, Q_0, \delta, \varphi)$ consists of

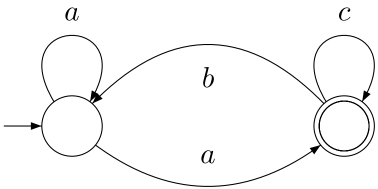
- Σ: alphabet
- Q: finite set of states
- $Q_0 \subseteq Q$: initial states
- δ: transition function
  we will use **deterministic, nondeterministic, and universal** branching modes
- φ: acceptance condition
  we will use **Büchi** and **parity** acceptance conditions

Acceptance conditions: Büchi

A **Büchi condition** is a set of accepting states $F \subseteq Q$.

An infinite sequence $q_0 q_1 q_2 \ldots \in Q^\omega$ is Büchi-accepted iff for **infinitely many** i, $q_i \in F$.

Nondeterministic Büchi automata (NBA)



An NBA $\mathcal{A} = (\Sigma, Q, Q_0, \delta, F)$ consists of the following:

- Σ: alphabet
- Q: states
- $Q_0 \subseteq Q$: initial states
- $\delta : Q \times \Sigma \to 2^Q$: transition function
- $F \subseteq Q$: accepting states

NBA acceptance

- A **run** of a NBA $\mathcal{A} = (\Sigma, Q, Q_0, \delta, F)$ on an **infinite** input sequence $\sigma_0\sigma_1 \ldots \Sigma^\omega$ is an **infinite** sequence of states $q_0\, q_1 \ldots$, such that the following conditions are satisfied:
  - $q_0 \in Q_0$ and
  - $q_{i+1} \in \delta(q_i, \sigma_i)$ for all $0 \leq i$
- The run $q_0\, q_1\, q_2 \ldots$ is **accepting** if $q_i \in F$ **for infinitely many** i.
- An input sequence is **accepted** by $\mathcal{A}$ iff there exists an accepting run.

The language of $\mathcal{A}$:

$$\mathcal{L}_\omega(\mathcal{A}) = \left\{\sigma \in \Sigma^\omega \mid \sigma \text{ is accepted by } \mathcal{A} \right\}$$

Two NBAs $\mathcal{A}$ und $\mathcal{A}'$ are **equivalent** iff $\mathcal{L}_\omega(\mathcal{A}) = \mathcal{L}_\omega(\mathcal{A}')$.

NBA vs. NFA

- finite equivalence $\not\Rightarrow$ Büchi equivalence

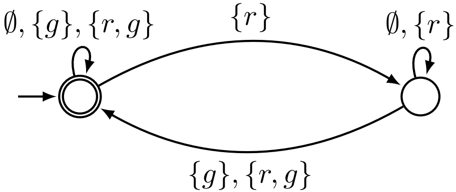
- Büchi equivalence $\not\Rightarrow$ finite equivalence

A simplified example

A simple response property

$$\mathsf{G}\,(r \rightarrow \mathsf{F}\,g)$$

$\Sigma = \{\emptyset, \{r\}, \{g\}, \{r, g\}\}$



LTL and NBA

Theorem [Vardi, Wolper'83]

Given an LTL formula φ, one can construct an NBA $\mathcal{A}_\varphi$ such that

$$\mathcal{L}_\omega(\mathcal{A}) = \{\sigma \in \Sigma^\omega \mid \sigma \models \varphi\}.$$

Furthermore, the size of$\mathcal{A}_\varphi$ is at most exponential in the length of φ.

Remark: The converse is not true.

Example: $L = (\emptyset\emptyset)^*\{p\}^\omega$.

There is no LTL formula φ with $L = \{\sigma \in \Sigma^\omega \mid \sigma \models \varphi)\}$.

However, there is a NBA $\mathcal{A}$ with $\mathcal{L}(\mathcal{A}) = L$.

Strategies

- Atomic propositions partitioned into input and output variables

$$AP = AP_I \uplus AP_O.$$

- Input alphabet $\Sigma_I = 2^{AP_I}$, output alphabet $\Sigma_O = 2^{AP_O}$.

Strategy for the system: function from sequences of inputs to outputs

$$f : \Sigma_I^* \to \Sigma_O.$$

Finite-state strategies: implemented by some finite state machine.

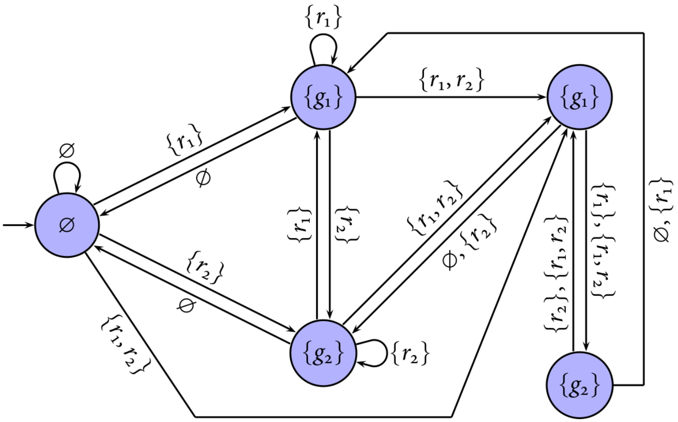
Given a sequence of inputs $i_0, i_1, i_2 \ldots$, a strategy f produces a trace

$$(i_0 \cup f(\varepsilon)), (i_1 \cup f(i_0)), (i_2 \cup f(i_0 i_1)), \ldots \in \Sigma^\omega$$

$$\mathit{Outcome}(f) = \\ \{(i_0 \cup f(\varepsilon)), (i_1 \cup f(i_0)), (i_2 \cup f(i_0 i_1)), \ldots \mid i_0, i_1, i_2 \ldots \in \Sigma_I^\omega\}$$

Example: Arbiter strategy

- receives **requests** r_1, r_2 from two clients and
- produces **grants** g_1 and g_2 for the two clients.

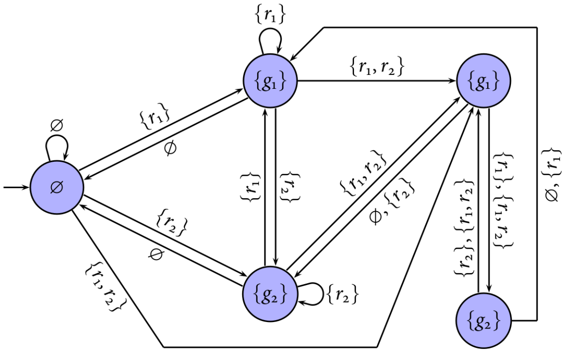


Winning strategy

Given an LTL specification φ over $AP_I \uplus AP_O$, a strategy f for the system **satisfies** φ iff we have $\sigma \models \varphi$ for all $\sigma \in Outcome(f)$.

Example

$$\varphi = (\mathsf{G}\,\neg(g_1 \wedge g_2)) \wedge (\mathsf{G}\,((r_1 \rightarrow \mathsf{F}\,g_1) \wedge (r_2 \rightarrow \mathsf{F}\,g_2)))$$



LTL Realizability

Given φ, does there exist a strategy f that satisfies φ?

Approach: Construct a game between environment (providing input), and system (choosing output). Check if system has a winning strategy.

Attempt: Let $\mathcal{A}_\varphi = (\Sigma, Q, Q_0, \delta, F)$ be an NBA for φ

- System chooses output value $o \in \Sigma_O$
- Environment chooses output value $i \in \Sigma_O$
- Round: system and environment set their variables
- Play: infinite word in Σ^ω
- System wins if infinite play accepted by $\mathcal{A}_\varphi$

Problem: In a nondeterministic automaton, an accepted word can also have rejecting runs. Mismatch between nondeterminism and strategic choice: the system can lose just because the wrong run was chosen.

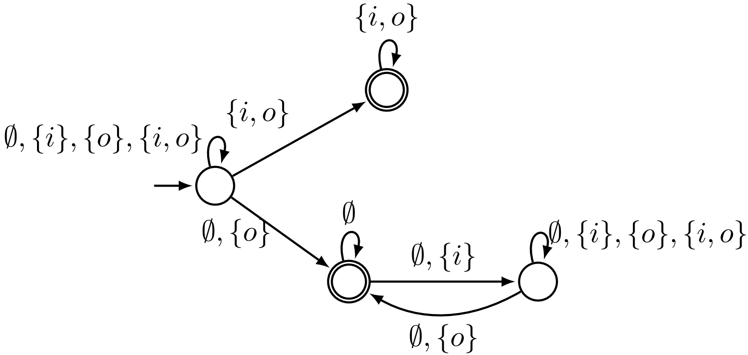
Why are nondeterministc automata not suitable?

Nondeterministic automata have perfect foresight.

Strategies have no foresight.

Example: $(\mathsf{FG}(i \wedge \mathsf{X}o)) \vee (\mathsf{GF}(\neg i \wedge \mathsf{X}\neg o))$

- System has winning strategy (copy input to output).
- The system cannot choose between the two disjuncts.



Deterministic Büchi automata (DBA)

A NBA $\mathcal{A}$ is a **DBA** iff

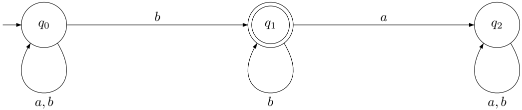
$$|Q_0| \leq 1 \quad \text{and} \quad |\delta(q,\sigma)| \leq 1 \quad \text{for all } q \in Q \text{ and } \sigma \in \Sigma$$

A DBA $\mathcal{A}$ is **complete** iff

$$|Q_0| = 1 \quad \text{and} \quad |\delta(q,\sigma)| = 1 \quad \text{for all } q \in Q \text{ and } \sigma \in \Sigma$$

Complete DBAs have a **unique** run for every input word.

NBAs are strictly more expressive than DBAs



There is no DBA $\mathcal{A}$ with
$\mathcal{L}_\omega(\mathcal{A}) = \{\sigma \in \{a, b\}^\omega \mid$ there are only finitely many as in $\sigma\}$

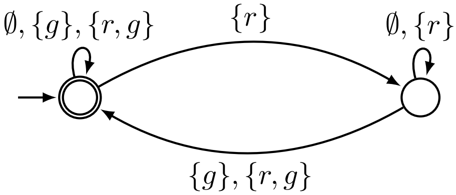
- Assume, by way of contradiction, that $\mathcal{A}$ is a DBA with this language.
- Since $b^\omega \in \mathcal{L}_\omega(\mathcal{A})$, there is a run $\pi_0 = q_{0,0}q_{0,1}q_{0,2}, \ldots$ where $q_{0,n_0} \in F$ for some $n_0 \geq 0$.
- Analogously, $b^{n_0}ab^\omega \in \mathcal{L}_\omega(\mathcal{A})$ and there is a run $\pi_1 = q_{0,0}q_{0,1}q_{0,2} \cdots q_{0,n_0}q_{1,0}q_{1,1}q_{1,2} \cdots$ with $q_{1,n_1} \in F$ for some $n_1 \geq 0$,
- Repeat.
- Thus there is a sequence $b^{n_0}ab^{n_1}ab^{n_2}a \ldots$ which is accepted by $\mathcal{A}$, but is not contained in the language.
- Contradiction.

Acceptance conditions: Parity

A **parity condition** is a coloring function $c : Q \to \mathbb{N}$.

An infinite sequence $q_0 q_1 q_2 \ldots \in Q^\omega$ is max-parity-accepted iff the highest color k that appears infinitely often is **even**.

DBA:

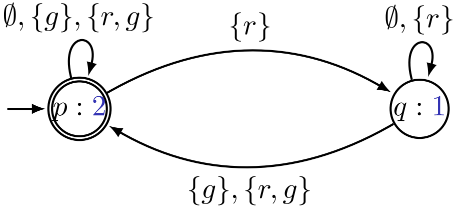


Acceptance conditions: Parity

A **parity condition** is a coloring function $c : Q \to \mathbb{N}$.

An infinite sequence $q_0 q_1 q_2 \ldots \in Q^\omega$ is max-parity-accepted iff the highest color k that appears infinitely often is **even**.

DPA:



Deterministic parity automata (DPA)

Theorem [McNaughton 1966]

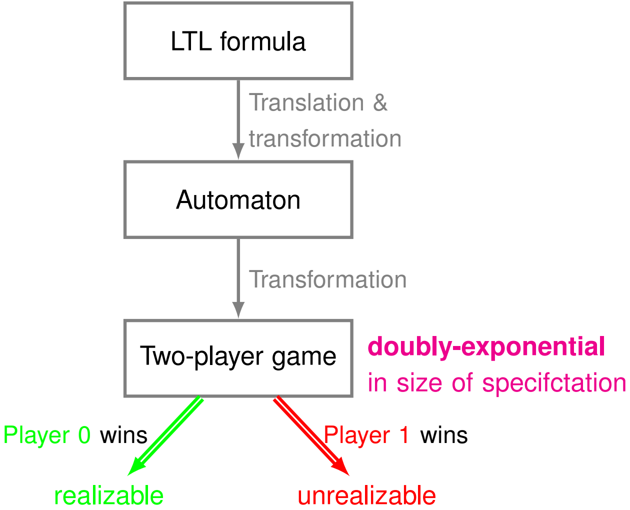
For each NBA there is an equivalent deterministic ω-automaton.

Theorem [Piterman'07]

For every NBA $\mathcal{N}$ with n states there exists a DPA $\mathcal{D}$ with $2n^n n!$ states and $2n$ colors such that $\mathcal{L}_\omega(\mathcal{D}) = \mathcal{L}_\omega(\mathcal{N})$.

The exponential blow-up is unavoidable.

Synthesis from LTL specifications



Exercise

Consider a coffee machine with

- input atomic propositions $button$ and $water$,
- output atomic proposition $coffee$.

1. A possible specification for the system is given by the LTL formula

$$(\mathsf{GF}(water) \rightarrow \mathsf{G}(button \rightarrow (\mathsf{F}coffee))) \wedge \mathsf{G}(\neg water \rightarrow \neg coffee)$$

   1.1 Give an Büchi automaton for that specification.
   1.2 Is your automaton deterministic?
   1.3 Does there exist a deterministic Büchi automaton for that specification? Explain why or give one if it exists.

2. Consider now a modified specification, where we take into account that the coffee machine can only react to input in the next step.

$$(\mathsf{GF}(water) \rightarrow \mathsf{G}(button \rightarrow \mathsf{X}(\mathsf{F}coffee))) \wedge \mathsf{G}(\neg water \rightarrow \neg \mathsf{X}coffee)$$

   2.1 Try to give an Büchi automaton for that specification. Try the tool at https://spot.lrde.epita.fr/app