Regular LT properties

Idea: define regular LT properties to be those languages of infinite words over the alphabet 2^{AP} that have a representation by a finite automata

- regular safety properties:
  NFA-representation for the bad prefixes
- other regular LT properties:
  representation by ω-automata, i.e., acceptors for infinite words

Overview

Introduction

Modelling parallel systems

Linear Time Properties

Regular Properties

- regular safety properties ⟵
- ω-regular properties
- model checking with Büchi automata

Linear Temporal Logic

Computation-Tree Logic

Equivalences and Abstraction

Nondeterministic finite automata (NFA)

NFA $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F)$

- Q finite set of states
- Σ alphabet
- $\delta : Q \times \Sigma \to 2^Q$ transition relation
- $Q_0 \subseteq Q$ set of initial states
- $F \subseteq Q$ set of final states, also called accept states

run for a word $A_0 A_1 \dots A_{n-1} \in \Sigma^*$:

state sequence $\pi = q_0\, q_1 \dots q_n$ where $q_0 \in Q_0$
and $q_{i+1} \in \delta(q_i, A_i)$ for $0 \leq i < n$

run π is called accepting if $q_n \in F$

Nondeterministic finite automata (NFA)

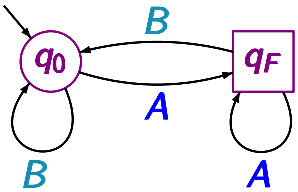
NFA $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F)$

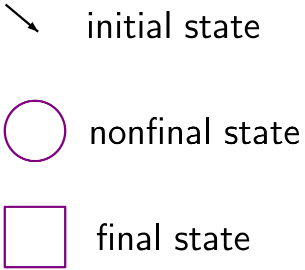
- Q finite set of states
- Σ alphabet $\longleftarrow$ here: $\Sigma = 2^{AP}$
- $\delta : Q \times \Sigma \rightarrow 2^Q$ transition relation
- $Q_0 \subseteq Q$ set of initial states
- $F \subseteq Q$ set of final states, also called accept states

accepted language $\mathcal{L}(\mathcal{A}) \subseteq \Sigma^*$ is given by:

$\mathcal{L}(\mathcal{A})$ = set of finite words over Σ that have an accepting run in $\mathcal{A}$

Notations in pictures for NFA





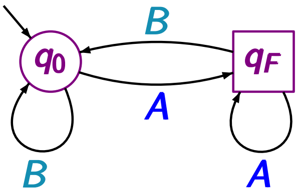
NFA $\mathcal{A}$ with state space $\{q_0, q_F\}$

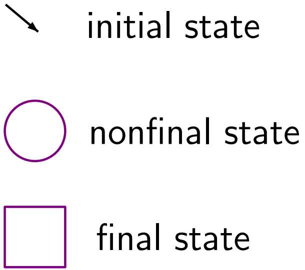
q_0 initial state

q_F final state

alphabet $\Sigma = \{A, B\}$

Notations in pictures for NFA





accepted language $\mathcal{L}(\mathcal{A})$:

set of all finite words over $\{A, B\}$
ending with letter A

NFA $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F)$ over the alphabet $\Sigma = 2^{AP}$

symbolic notation for the labels of transitions:

> If Φ is a propositional formula over AP then
>
> $q \xrightarrow{\Phi} p$ stands for the set of transitions $q \xrightarrow{A} p$
>
> where $A \subseteq AP$ such that $A \models \Phi$

Example: if $AP = \{a, b, c\}$ then

$$q \xrightarrow{a \wedge \neg b} p \;\widehat{=}\; \left\{ q \xrightarrow{A} p : A{=}\{a, c\} \text{ or } A{=}\{a\} \right\}$$

$$q \xrightarrow{true} p \;\widehat{=}\; \left\{ q \xrightarrow{A} p : A \subseteq AP \right\}$$

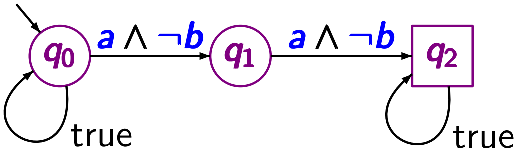
Regular safety properties

A safety property $E \subseteq \left(2^{AP}\right)^{\omega}$ is called regular iff

$BadPref$ = set of all bad prefixes for $E \subseteq \left(2^{AP}\right)^{+}$

$\boxed{BadPref = \mathcal{L}(\mathcal{A}) \text{ for some NFA } \mathcal{A} \text{ over the alphabet } 2^{AP}}$

is regular.



$AP = \{a, b\}$

symbolic notation:

$a \wedge \neg b \mathrel{\widehat{=}} \{a\}$

safety property E: "$a \wedge \neg b$ never holds twice in a row"

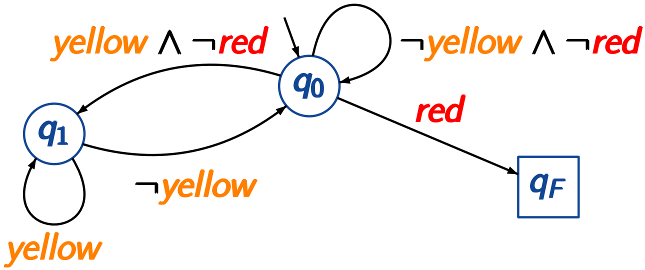
Example: regular safety property

"Every red phase is preceded by a yellow phase"

set of all infinite words $A_0\,A_1\,A_2\ldots$ s.t. for all $i \geq 0$:

$$red \in A_i \implies i \geq 1 \text{ and } yellow \in A_{i-1}$$

DFA for minimal bad prefixes



Verifying regular safety properties

given: finite TS $\mathcal{T}$
regular safety property E
(represented by an **NFA** for its bad prefixes)

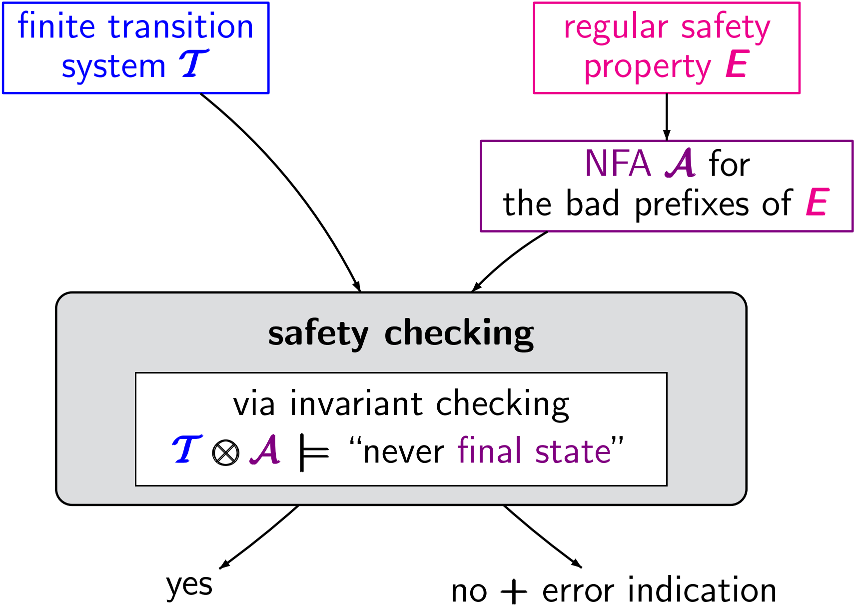
question: does $\mathcal{T} \models E$ hold **?**

method: relies on an analogy between the tasks:

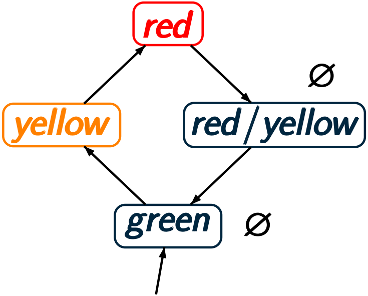
- checking language inclusion for **NFA**
- model checking regular safety properties

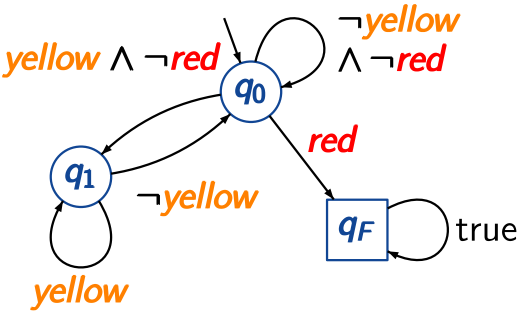
language inclusion for NFA	verification of regular safety properties
$\mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2)$ **?**	$\mathit{Traces}(\mathcal{T}) \subseteq E$ **?**
check whether $\mathcal{L}(\mathcal{A}_1) \cap (\Sigma^* \setminus \mathcal{L}(\mathcal{A}_2))$ is empty	check whether $\mathit{Traces}_{\mathit{fin}}(\mathcal{T}) \cap \mathit{BadPref}$ is empty
1. complement $\mathcal{A}_2$, i.e., construct NFA $\overline{\mathcal{A}_2}$ with $\mathcal{L}(\overline{\mathcal{A}_2}) = \Sigma^* \setminus \mathcal{L}(\mathcal{A}_2)$ 2. construct NFA $\mathcal{A}$ with $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\overline{\mathcal{A}_2})$ 3. check if $\mathcal{L}(\mathcal{A}) = \varnothing$	1. construct NFA $\mathcal{A}$ for the bad prefixes $\mathcal{L}(\overline{\mathcal{A}}) = \mathit{BadPref}$ 2. construct TS $\mathcal{T}'$ with $\mathit{Traces}_{\mathit{fin}}(\mathcal{T}') = \ldots$ 3. invariant checking for $\mathcal{T}'$

Checking regular safety properties



Example: product-TS





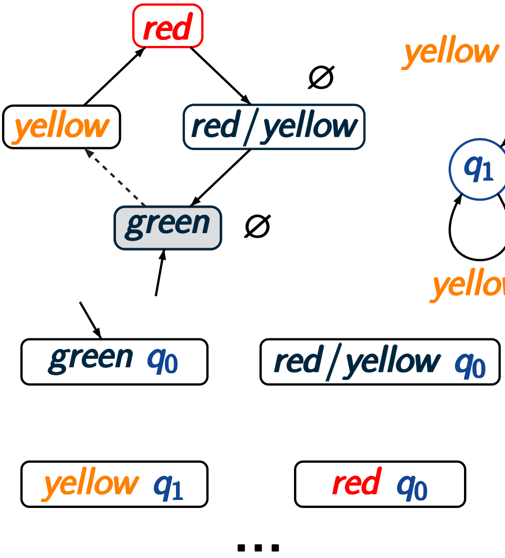
transition system $\mathcal{T}$ over $AP = \{red, yellow\}$

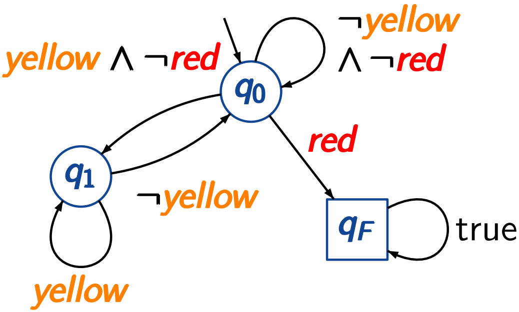
DFA $\mathcal{A}$ for the bad prefixes for E

$\mathcal{T}$ satisfies the safety property E

"every red phase is preceded by a yellow phase"

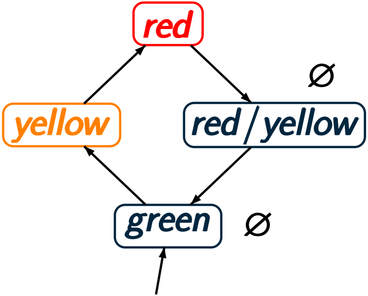
Example: product-TS

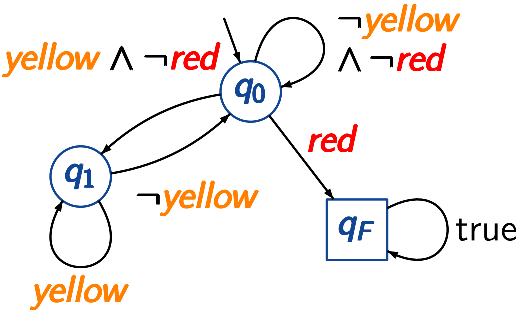


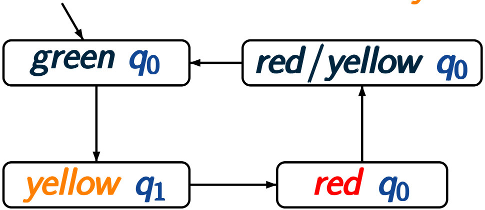


lifting the transition

green $\longrightarrow$ yellow



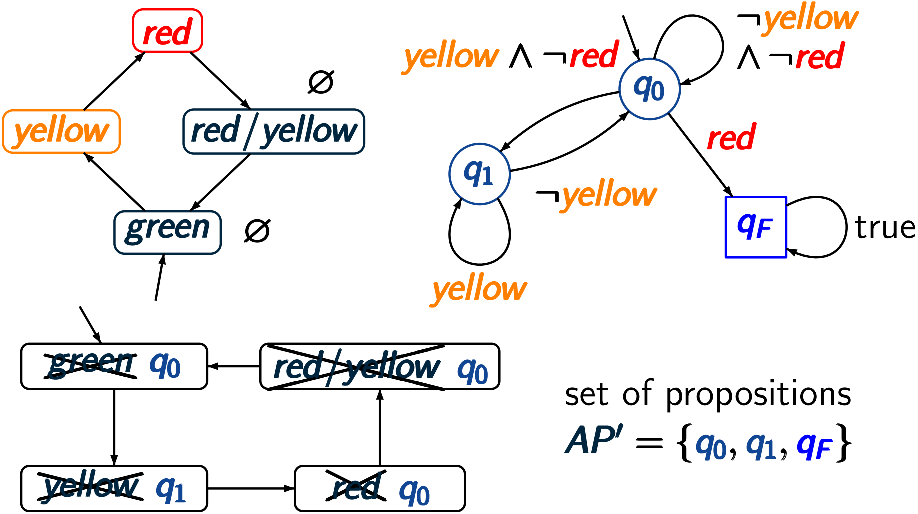




product-TS
$\mathcal{T} \otimes \mathcal{A}$

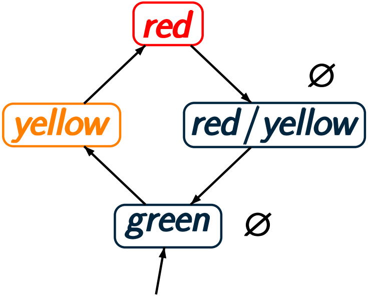
$4 * 3 = 12$ states, but
just 4 reachable states

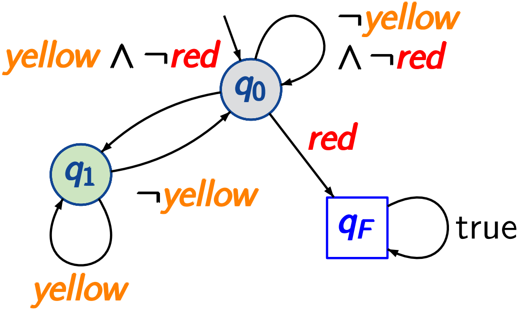
Example: product-TS

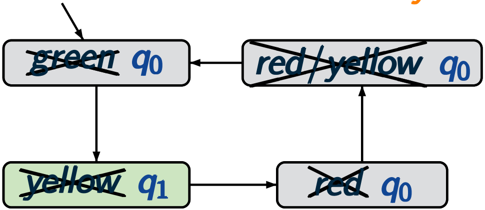


set of propositions
$AP' = \{q_0, q_1, q_F\}$

Example: product-TS





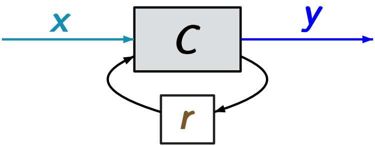


set of propositions

$$AP' = \{q_0, q_1, q_F\}$$

invariant condition $\neg q_F$ holds
for all reachable states

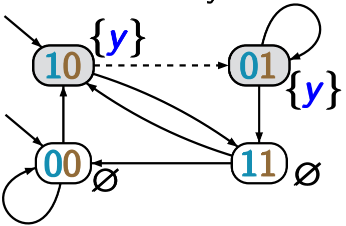
Example: sequential circuit



$\lambda_y = \delta_r = x \oplus r$

initially $r = 0$

transition system $\mathcal{T}$

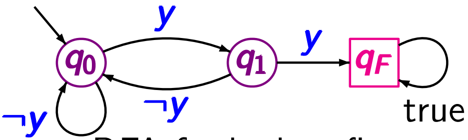


$\mathcal{T} \not\models E$

error indication, e.g.,

$\langle 10 \rangle \, \langle 01 \rangle$

bad prefix: $\{y\}\,\{y\}$



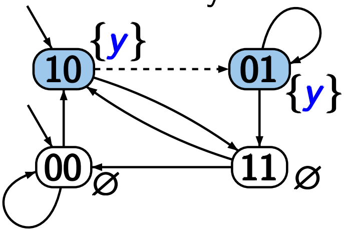
DFA for bad prefixes

safety property E

The circuit will never ouput two ones after each other

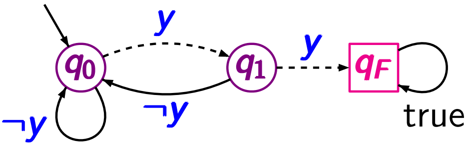
Example: product-TS

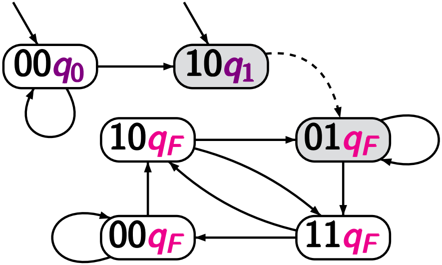
transition system $\mathcal{T}$



safety property E

... never two ones in a row ...





error indication for $\mathcal{T} \not\models E$

error indication for $\mathcal{T} \otimes \mathcal{A} \not\models$ "never q_F"