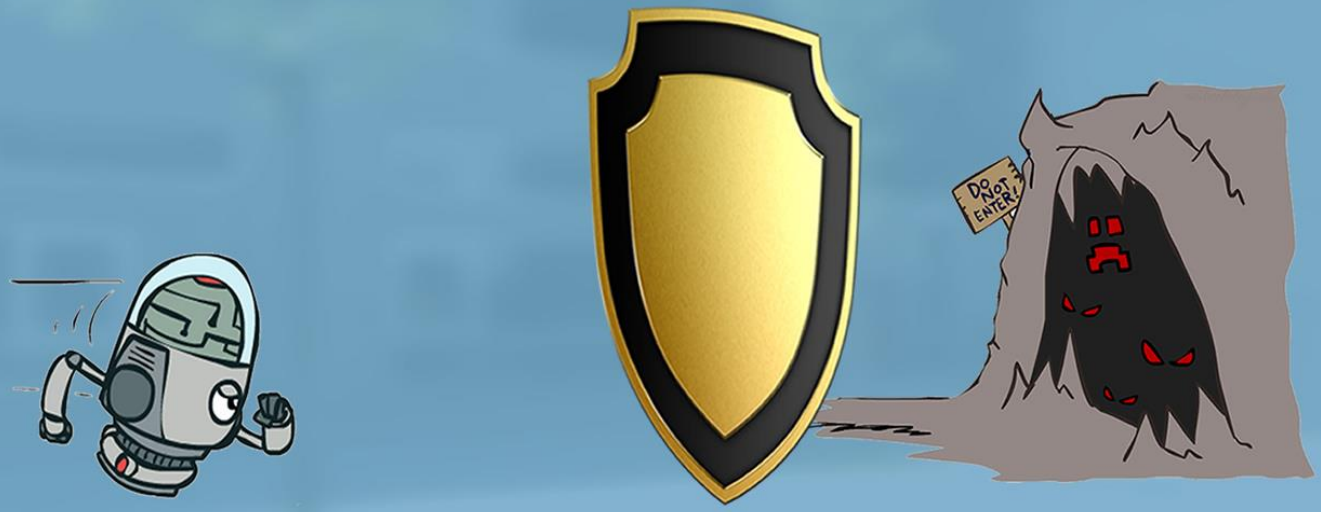
Graz University of Technology
Institute for Applied Information
Processing and Communications

Bounded Synthesis for Safety Specifications

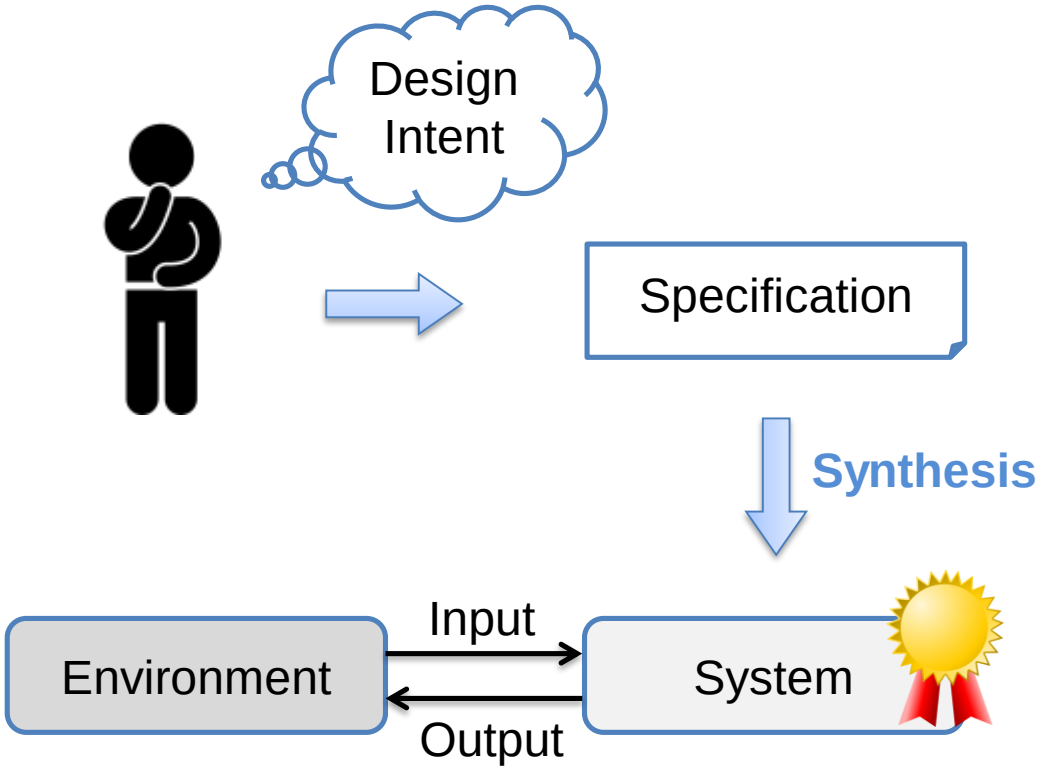


Bettina Könighofer
bettina.koenighofer@iaik.tugraz.at

March 22, 2020

[The robot illustrations come from the slides by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley, available at http://ai.berkeley.edu.]

What is Synthesis?



How does Synthesis work?

1. Synthesis via Game Solving
2. Bounded Synthesis

1. Synthesis is a Game

Specification

Game

Winning Region W

Winning Strategy S

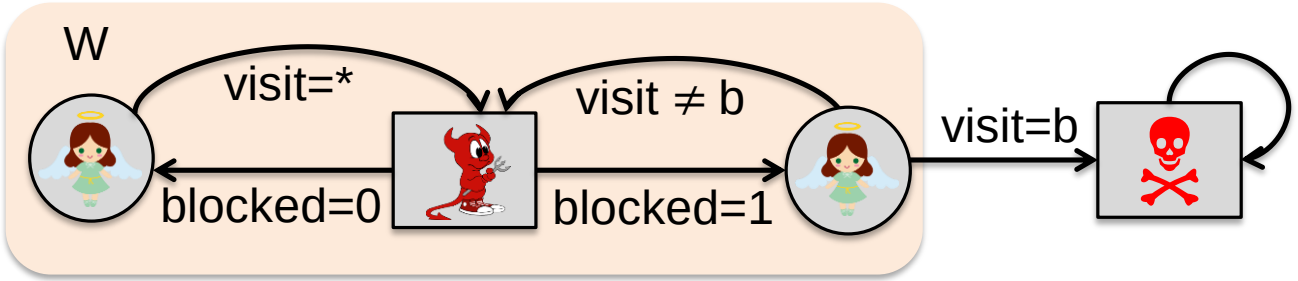
System Implementation

in

out

Environment

$\mathbf{always}(\text{blocked} = 1 \rightarrow \text{visit} \neq b)$



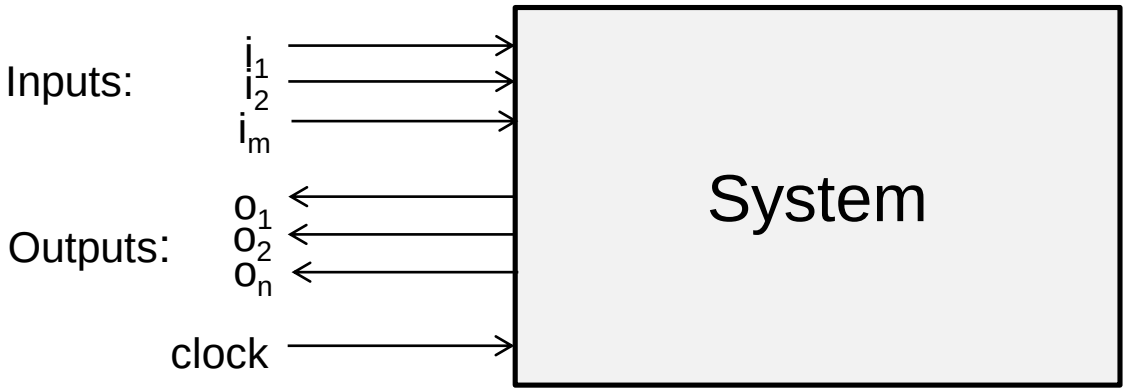
From which states can win?

What shall do to win?

How does Synthesis work?

1. Synthesis via Game Solving
2. Bounded Synthesis
   - Given k, is there a system with k states that fulfills the specification?
   - SMT Encoding
   - Today: Consider only safety specifications

System

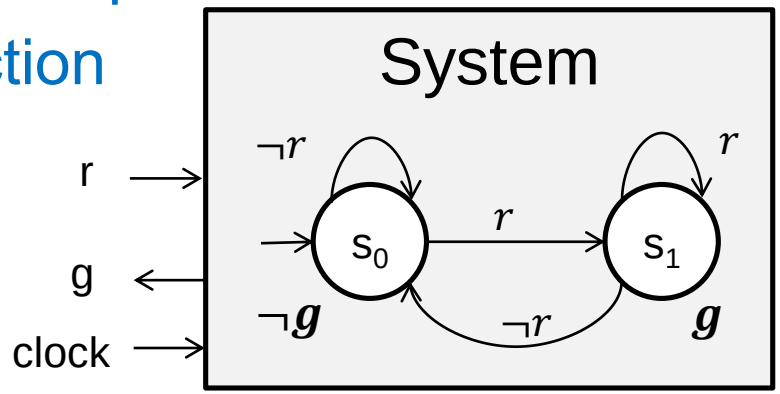


- $I = \{i_1, i_2, \ldots, i_m\}$ is a set of Boolean inputs
- $O = \{o_1, o_2, \ldots, o_n\}$ is a set of Boolean outputs

System: What is Inside?

Moore Machine:

- A Moore Machine is a tuple $\mathrm{M} = (S, s_0 \Sigma_I, \Sigma_\mathrm{O}, \tau, out)$
  - S … finite set of states
  - s_0 …initial state
  - I …inputs, $\Sigma_I = 2^I$ …input alphabet
  - O …outputs, $\Sigma_\mathrm{O} = 2^O$ …output alphabet
  - $\tau: S \times \Sigma_I \rightarrow S$ …transition function
  - $out: S \rightarrow \Sigma_O$ …output function



System trace

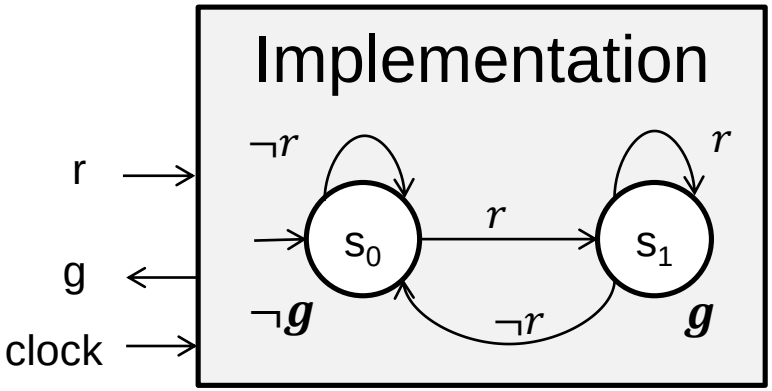
- A system **path** is an infinite sequence $(s_o i_0)(s_1 i_1)(s_2, i_2)\ldots$
  - such that $s_1 = \tau(s_0, i_0), s_2 = \tau(s_1, i_1), \ldots$

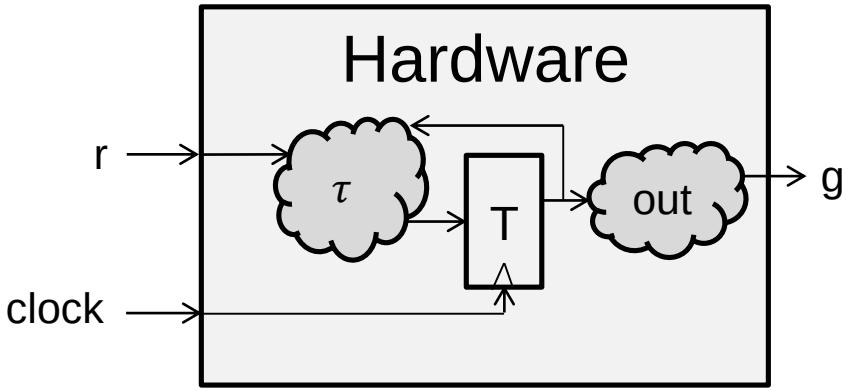
- A corresponding system trace is an infinite sequence $\bar{\sigma} = (out(s_0, i_0))(out(s_1, i_1)) \ldots$

From a Moore Machine to Hardware

Simple Transformation:

- The current state $s \in S$ is stored in flip-flops
- $\tau: S \times \Sigma_I \rightarrow S$ and $out: S \rightarrow \Sigma_o$ are combinatorial circuits





Safety Specification

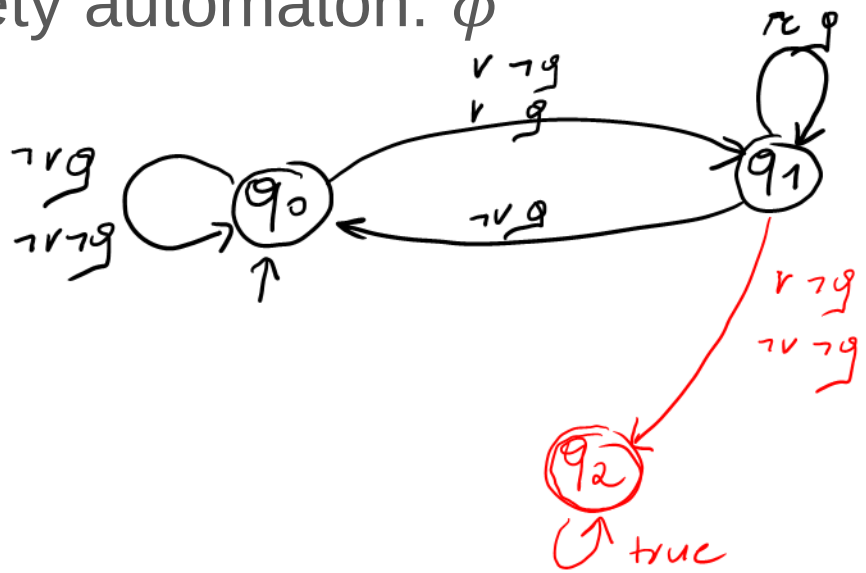
- $\forall \bar{\sigma} \in (\Sigma_I \times \Sigma_o)^*. \left(\bar{\sigma} \not\models \phi \rightarrow (\forall \bar{\sigma}' \in (\Sigma_I \times \Sigma_o)^*. (\bar{\sigma} \cdot \bar{\sigma}') \not\models \phi)\right)$
  - Traces that do not satisfy ϕ cannot be extended to traces that satisfy ϕ
  - “bad things must be irremediable”

Specification = Safety Automata

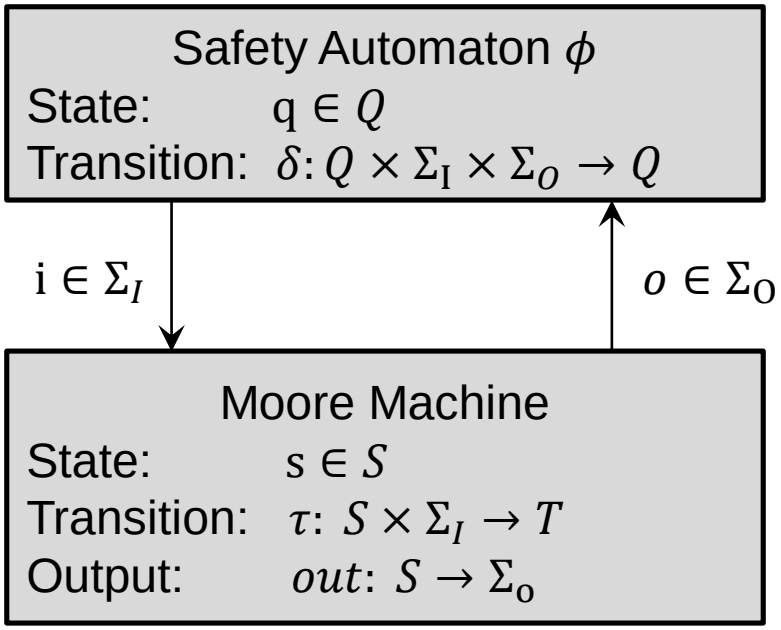
- A Safety Automaton is a tuple $\phi = (Q, q_0, \Sigma_I, \Sigma_O, \delta, F)$
  - Q …finite set of states
  - q_0 …initial state
  - I …inputs, $\Sigma_I = 2^I$ …input alphabet
  - O …outputs, $\Sigma_O = 2^O$ …output alphabet
  - $\delta: Q \times \Sigma_I \times \Sigma_O \rightarrow Q$ …transition function
  - $F \subseteq Q$ …set of unsafe states

Example: Safety Specification

- Every request must be granted in the next tick
- In LTL: $\phi = G(r \rightarrow Xg)$
- Safety automaton: ϕ



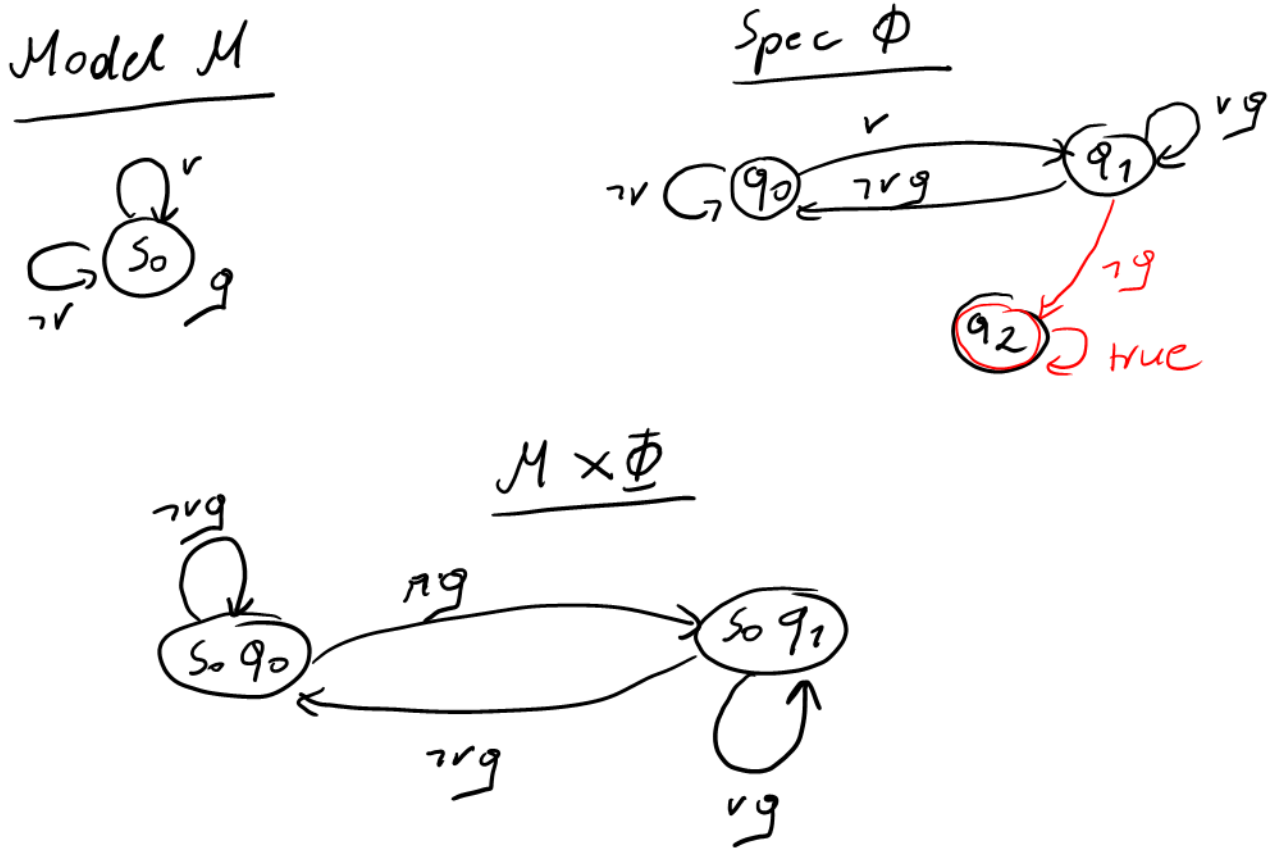
Run of ϕ on M



Run Graph = Product $M \times \phi$

- Take current state of M: s
- Take current state of ϕ: q
  - (s,q) is the current state of a run graph
- Take the current output: out(s)
- Take any input: i
  - (i, out(s)) is the current edge label
- Take next state of M: $s' = \tau(s, i)$
- Take next state of SA: $q' = \delta(q, i, out(s))$
  - (s',q') is the next state of a run graph

Example: Building the Product



Bounded Synthesis for Safety Specs

- Given a safety specification ϕ
- Given a number *k*
- *I*s there a system with *k* states that fulfills ϕ?
  - Search $\mathrm{M} = (S, s_0, \Sigma_I, \Sigma_\mathrm{O}, \tau, out)$
    - Given: $S, s_0, \Sigma_I, \Sigma_\mathrm{O}$
    - Find: τ, out
  - How can we do this?

Uninterpreted functions to the rescue!

Bounded Synthesis for Safety Specs

- Search $\mathrm{M} = (S, s_0, \Sigma_I, \Sigma_O, \tau, out)$ ← uninterpreted

$$\rho(q_0, s_0) = true$$

Bounded Synthesis for Safety Specs

- Search $\mathrm{M} = (S, s_0, \Sigma_I, \Sigma_O, \tau, out)$ ← uninterpreted

$$\rho(q_0, s_0) = true$$

$$\bigwedge_{s \in S} \bigwedge_{q \in F} \rho(q, s) = false$$

Bounded Synthesis for Safety Specs

- Search $\mathrm{M} = (S, s_0, \Sigma_I, \Sigma_O, \tau, out)$ ← uninterpreted

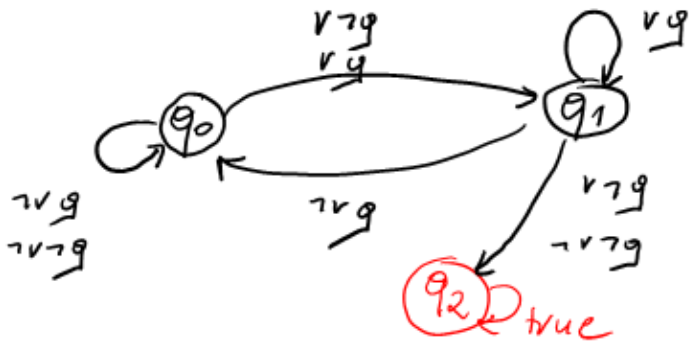
$$\rho(q_0, s_0) = true$$

$$\bigwedge_{q \in Q} \bigwedge_{i \in \Sigma_I} \bigwedge_{s \in S} \rho(q, s) \wedge o = out(s) \rightarrow \rho(\delta(q, i, o), \tau(s, i))$$

$$\bigwedge_{s \in S} \bigwedge_{q \in F} \rho(q, s) = false$$

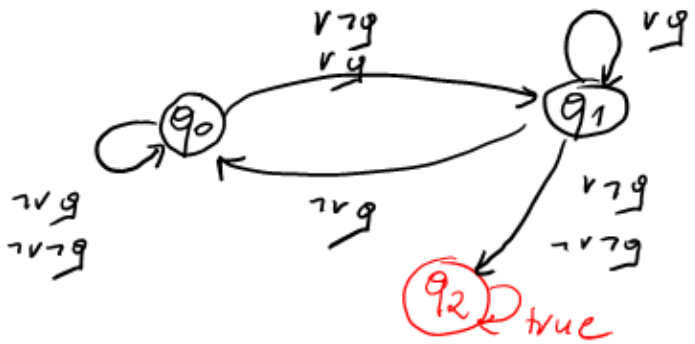
Example

- $\phi = (Q, q_0, \Sigma_I, \Sigma_O, \delta, F)$
  - $Q = \{q_0, q_1, q_2\}$
  - $I=\{r\}, O=\{g\}.$
  - $F = \{q_2\}$
- $k=2$



Example

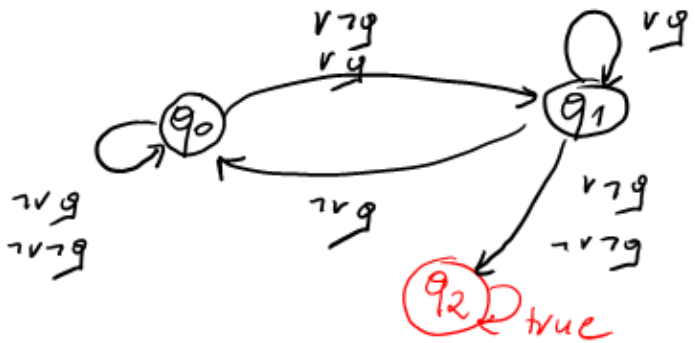
- $\phi = (Q, q_0, \Sigma_I, \Sigma_O, \delta, F)$
  - $Q = \{q_0, q_1, q_2\}$
  - *I={r}, O={g}.*
  - $F = \{q_2\}$
- *k=2*



$$\rho(q_0, s_0) = true$$

Example

- $\phi = (Q, q_0, \Sigma_I, \Sigma_O, \delta, F)$
  - $Q = \{q_0, q_1, q_2\}$
  - $I=\{r\}, O=\{g\}.$
  - $F = \{q_2\}$
- $k=2$

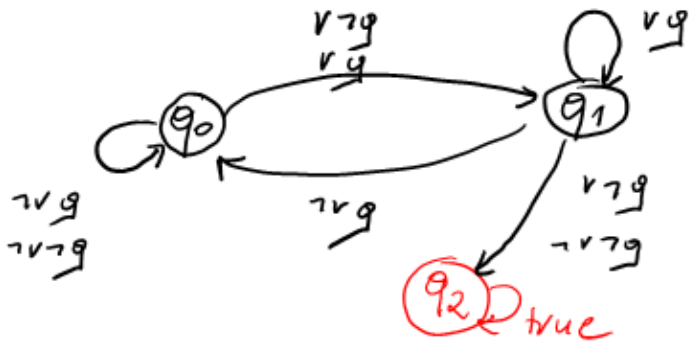


$$\rho(q_0, s_0) = true$$

$$\bigwedge_{s \in S} \bigwedge_{q \in F} \rho(q, s) = false \quad \Rightarrow \quad \neg\rho(q_2, s_0) \wedge \neg(q_2, s_1)$$

Example

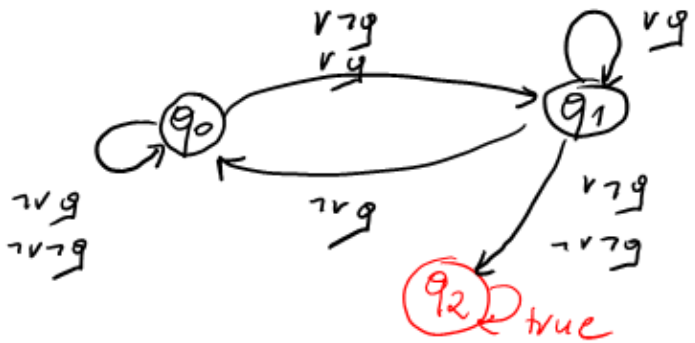
- $\phi = (Q, q_0, \Sigma_I, \Sigma_O, \delta, F)$
  - $Q = \{q_0, q_1, q_2\}$
  - $I=\{r\}$, $O=\{g\}$.
  - $F = \{q_2\}$
- $k=2$



$$\bigwedge_{q \in Q} \bigwedge_{i \in \Sigma_I} \bigwedge_{s \in S} \rho(q, s) \wedge o = out(s) \rightarrow \rho(\delta(q, i, o), \tau(s, i))$$

Example

- $\phi = (Q, q_0, \Sigma_I, \Sigma_O, \delta, F)$
  - $Q = \{q_0, q_1, q_2\}$
  - $I=\{r\}, O=\{g\}.$
  - $F = \{q_2\}$
- $k=2$



$$\bigwedge_{q\in Q}\bigwedge_{i\in\Sigma_I}\bigwedge_{s\in S} \rho(q,s) \wedge o = out(s) \rightarrow \rho(\delta(q,i,o),\tau(s,i))$$

$$\rho(q_0,s_0) \wedge o = out(s_0) \rightarrow \rho\big(\delta(q_0,\neg r,o),\tau(s_0,\neg r)\big) \wedge$$
$$\rho(q_0,s_1) \wedge o = out(s_1) \rightarrow \rho\big(\delta(q_0,\neg r,o),\tau(s_1,\neg r)\big) \wedge$$
$$\rho(q_0,s_0) \wedge o = out(s_0) \rightarrow \rho\big(\delta(q_0,r,o),\tau(s_0,r)\big) \wedge$$
$$\rho(q_0,s_1) \wedge o = out(s_1) \rightarrow \rho\big(\delta(q_0,r,o),\tau(s_1,r)\big) \wedge$$
$$\rho(q_1,s_0) \ldots$$

Thank You