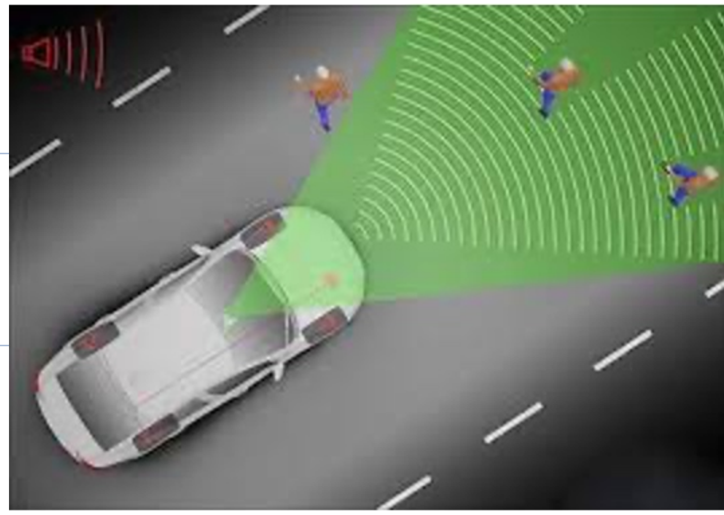
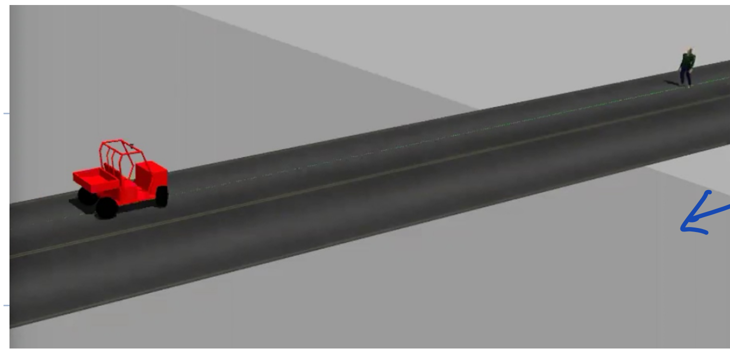
ECE 484 8/23

Lecture 2 : System-level Safety





MPO Walkthrough in Lab tomorrow!

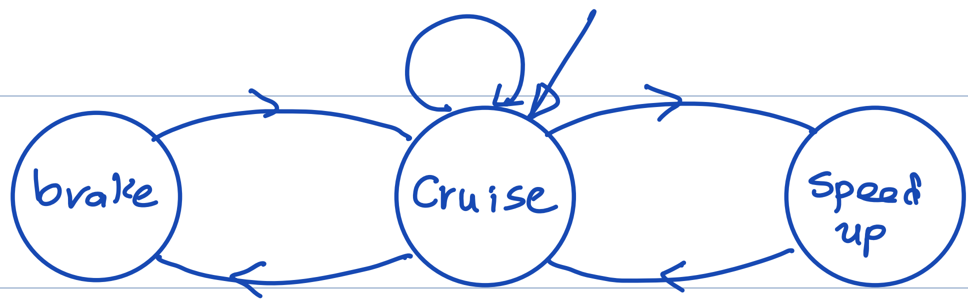
Today: Automata
Nondeterminism
Executions – testing
Reachability / Post computations
Invariance

Def. A state machine or automaton A is defined by

(1) a set of states Q
(2) a set of start states $Q_0 \subseteq Q$
(3) a set of transitions $D \subseteq Q \times Q$

transition relation

Example.



$Q = \{C, B, S\}$ $\quad Q_0 = \{C\}$

$D = \{\langle B,C\rangle, \langle C,B\rangle, \langle C,C\rangle, \langle C,S\rangle, \langle S,C\rangle\}$

Nondeterministic.

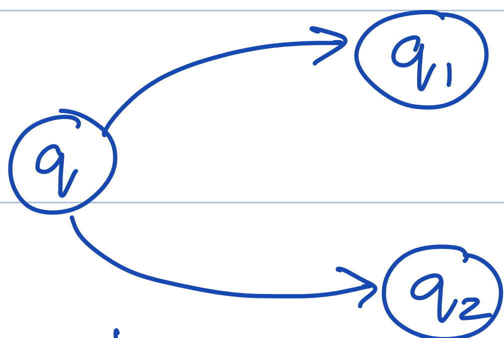
- From the same state A can go to different states
- Useful for modeling uncertainty e.g. action of human driver or environment.

Aside. If we add probabilities to transitions then we get a Markov Chain (MC).

$D : Q \to \mathcal{P}(Q)$

We can have both nondeterminism & probabilistic uncertainty
$\longrightarrow$ Markov Decision Processes

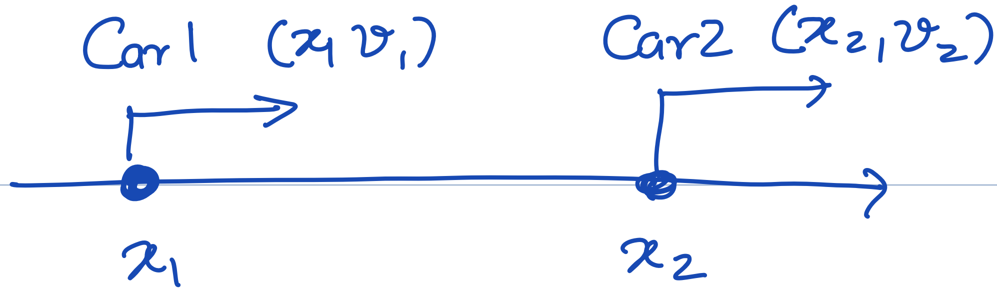
Deterministic automaton $|Q_0| = 1$ and $\forall q \in Q$, $q_1, q_2 \in Q$ if $\langle q, q_1 \rangle \in D$ and $\langle q, q_2 \rangle \in D$ then $q_1 = q_2$



✗ not allowed

For deterministic automata
$D : Q \to Q$ is a transition function.

Example 2



$Q = \mathbb{R}^4$

$Q_0 = \{ x_{10}, v_{10}, x_{20}, v_{20} \}$ $\quad x_{20} > x_{10} > 0$

$D \subseteq \mathbb{R}^4 \times \mathbb{R}^4$

Often D will be described by a program or a physics model (differential equations)

If $x_2 - x_1 < d_s$

$\quad v_1 := \max(0, v_1 - a_b)$

else $\quad v_1 := v_1$

$x_1 := x_1 + v_1$

$x_2 := x_2 + v_2$

Do you see how this defines D?

Is it deterministic?

With some abuse of notation we can represent nondeterministic models also as programs

$$v_1 := \underline{\text{choose}}\,[v_1 - b_1 ,\; v_1 - a_1]$$

Fautly sensor

<u>Executions</u>. An <u>execution</u> is a particular behavior of the automaton A.

$\alpha = q_0\, q_1\, q_2 \,\ldots\ldots$ finite or infinite
such that
(i) $q_0 \in Q_0$
(ii) $(q_i, q_{i+1}) \in D \quad \forall i$

Nondeterministic automata have <u>many</u> executions.

A "Test" $\approx$ one Execution

Requirements of a design

Examples ? "Car 1 eventually catches up of C2" 5m

" Car 1 and car 2 never collide "
" Car 1 never goes backwards "
" Car 1 does not never exceed speed limits "

Safety Requirements

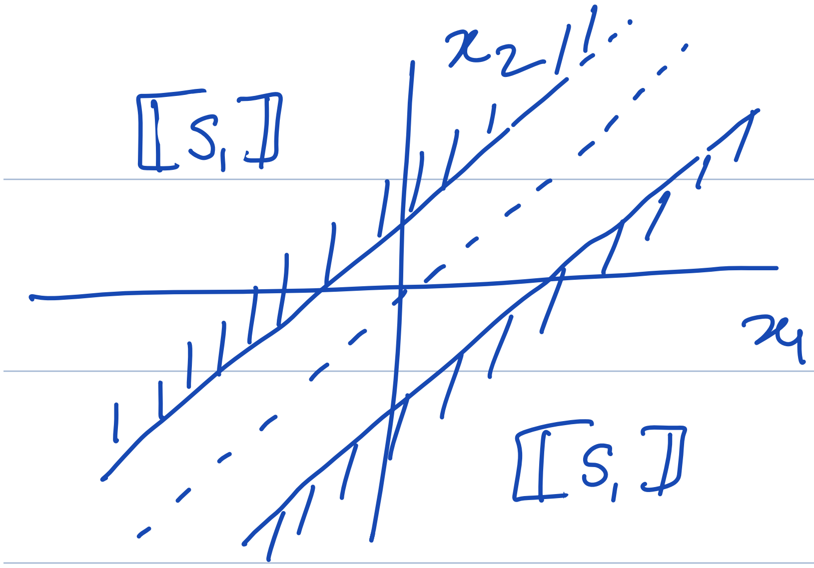
These are safety requirements because we want A to always satisfy them.

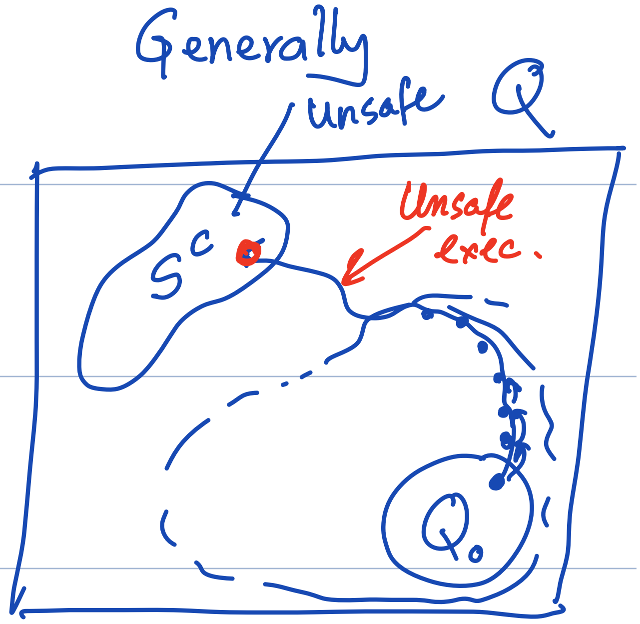
We can express the safety requirements as

(1) A formula involving the state variables. e.g. $S_1 := |x_1 - x_2| \geq 0.1$

(2) A subset of Q

$$[\![S_1]\!] \subseteq Q = \mathbb{R}^4 = \{\langle x_1, v_1, x_2, v_2 \rangle \mid |x_1 - x_2| \geq 0.1\}$$





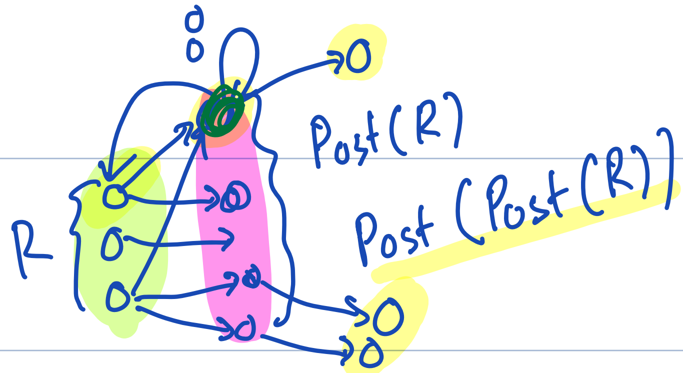
Safety Verification Problem

Does there exist any execution $\alpha = q_0 \cdots q_k$ of A such that $q_k \notin S^c$?

Such an execution is called a Counter-example

Def.

If for every finite execution $\alpha = q_0 \cdots q_k$ of $\mathcal{A}$ and for every q_i in α, $q_i \in S$ then we say $\mathcal{A}$ is safe w.r.t. S.



Reasoning about all executions

Def. For any set of states $R \subseteq Q$

$$Post(R) := \{ q' \in Q \mid \exists q \in R \text{ and } \langle q, q' \rangle \in D \}$$

Exercise The $Post()$ operator is monotonic

if $R_1 \subseteq R_2$ then $Post(R_1) \subseteq Post(R_2)$

Proof. Choose any $R_1 \subseteq R_2 \subseteq Q$

Choose any $x \in Post(R_1)$

[we have to show $x \in Post(R_2)$]

By def of Post $\exists x_0 \in R_1$ $\langle x_0, x \rangle \in D$

Since $R_1 \subseteq R_2 \Rightarrow x_0 \in R_2$

$\Rightarrow x \in Post(R_2)$ ∎

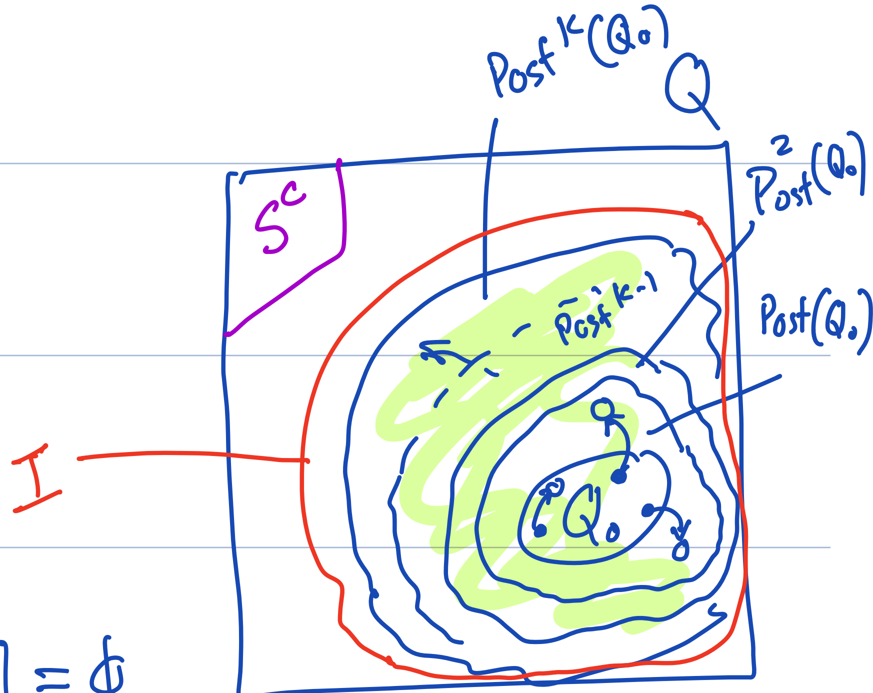
We can apply Post recursively.

Def $Post^k : 2^Q \rightarrow 2^Q$

$Post^0(R) = R$ $\quad k = 0$

$Post^k(R) = Post(Post^{k-1}(R))$ $\quad k > 0$

Exercise $Post^k(Q_0)$
is exactly the set of states that the automaton can reach after executions of length k.



Proof.

$$\left[\bigcup_{k=0}^{T} Post^k(Q_0)\right] \cap [\![S^c]\!] = \phi$$

Reachability analysis tools can Compute or Over approximate $post^k()$
E.g. Verse, SpaceEx, Flow*

In general Computing $Post^k(R)$ can be hard (1) Q high dimensional
(2) D complex, (3) k large

Alternative Solution to Safety verification Problem

Find an inductive invariant for proving safety of S.

inductive invariant

Thm if there exists $I \subseteq Q$ such that

(1) $Q_0 \subseteq I$ (2) $Post(I) \subseteq I$

Then all executions of A stay in I.

Further if $I \subseteq S$ then

Then A is safe w.r.t S.

Sufficient condition for proving safety

Requires us to find I

existential not constructive

not unique I necessarily

Proof. Consider any execution of $\mathcal{A}$ $\alpha = q_0 q_1 \cdots q_k$. We will prove by induction on k that $\forall i \; q_i \in I$.

Base Case. $k = 0$ $\quad \alpha = q_0 \in Q_0 \subseteq I$ by (1)

Inductive step. $\alpha = q_0 \cdots q_{k-1} q_k$ and $q_{k-1} \in I$. We will show $q_k \in I$.

By (2) $Post(I) \subseteq I$

as $q_{k-1} \in I \Rightarrow q_k \in I$

Therefore $\forall i \quad q_i \in I$

Further if $I \subseteq S$ then $\forall i \; q_i \in S$ ∎

Simple invariant and Safety

$S_1 := v_1 \geqslant 0$

How to prove that Carl never moves back?

Choose $I_1 = [\![S_1]\!]$ This may not always work

Use inductive invariance theorem

Does I_1 meet the conditions (1) ... (3)?

(1) $\forall q_0 \in Q_0$

$q_0 . v_{10} > 0$ [we assumed this]

$\Rightarrow q_0 \in [\![S_1]\!]$

(2) $Post(I) := \{ q' \mid q \in I \text{ and } (q, q') \in D \}$

For any state $q \in I$ if $q.v_1 \geqslant 0$

and $(q, q') \in D$ then we have to

show that $q'.v_1$ is also $\geqslant 0$

How are q and q' related by D?

Note

$q'.v_1 = \max(0, q.v_1 - a_b)$
≥ 0

If $x_2 - x_1 < d_s$ — Sensor distance
$v_1 := \max(0, v_1 - a_b)$ — braking deceleration
else $v_1 := v_1$
$x_1 := x_1 + v_1$
$x_2 := x_2 + v_2$

$q' \in [\![S_1]\!] = I$

$Post(I) \subseteq I$

It follows that S_1 is indeed an invariant; Car 1 never goes backwards.

Another Safety requirement

$S_2 : x_1 < x_2$

Is S_2 an inductive invariant?

(1) $Q_0 \subseteq S_2$ ✓ $0 < x_{10} < x_{20}$

(2) $Post(S_2) \subseteq S_2$?

Not neeessarily true

if $q.v_1 >> q.x_2 - q.x_1$

then $q'.x_1$ may exceed $q'.x_2$

We cannot prove S_2

We need to add or "discover" assumptions about d_s, a_b ....

to prove S_2.

Add more information in the model

timer := 0

Sensor distance

If $x_2 - x_1 < d_s$

if $v_1 > a_b$ then

$v_1 := v_1 - a_b$

timer := timer + 1 } (A)

else $v_1 := 0$ — (B)

else $v_1 := v_1$ — (C)

$x_1 := x_1 + v_1$

$x_2 := x_2 + v_2$

$I_3.\quad timer \leq \dfrac{v_{10} - v_1}{a_b}$

(1) $q_0.timer = 0 \leq \dfrac{v_{10} - v_{10}}{a_b} \leq 0$

(2) $q \in I_3 \Rightarrow q' \in I_3$

Three cases to consider

A. if $q.x_2 - q.x_1 < d_s$ and $q.v_1 > a_b$

then $q'.timer = q.timer + 1$

$$\leq \frac{v_{10} - q.v_1}{a_b} + 1 \quad [\text{ind hyp}]$$

$$= \frac{v_{10} - (q'.v_1 + a_b)}{a_b} + 1$$

$$= \frac{v_{10} - q'.v_1}{a_b}$$

B. if $q.x_2 - q.x_1 < d_s$ and $q.v_1 \leq a_b$

$q'.timer = q.timer$

$$\leq \frac{v_{10} - q.v_1}{a_b} + 1$$

Uses $v_1 \geq 0$

$$\leq \frac{v_{10} + 0}{a_b} + 1$$

C. if. $q.x_2 - q.x_1 \geq d_s$

$$q'.timer = q.timer \leq \frac{v_{10} - q.v_1}{a_b} + 1$$

$$\leq \frac{v_{10} - q'.v_1}{a_b} + 1$$

I_3 : $timer \leq \dfrac{v_{10} - v_1}{a_b}$ and $v_1 \geq 0$

$$\Rightarrow \quad timer \leq \frac{v_{10}}{a_b}$$

Still not enough to prove $x_2 - x_1 > 0$

Max distance traversed by Car 1 after detection $\leq v_{10} \cdot timer \leq \dfrac{v_{10}^2}{a_b}$

So, if $d_s > v_{10}^2 / a_b$ and $v_2 \geq 0$

then $I_3 \Rightarrow S_0 : x_2 > x_1$

That is, if $d_s > v_{10}^2/a_b$ then in any execution α of $\mathcal{A}$ and all states $q_1 \ldots q_k$ in α, $q_i.x_2 > q_i.x_1$.