Automata / State machines

- What is an automaton?
- What are behaviors of automata? Executions
- What are safety requirements?
- Why is safety analysis of automata hard?
- Basic method for proving safety of all executions

An automaton / state machine A is defined by
- A set of states Q
- A start set $Q_0 \subseteq Q$
- A set of transitions $D \subseteq Q \times Q$

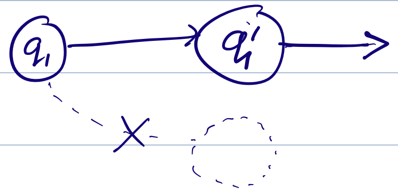
Example 1. Cruise control logic

A :

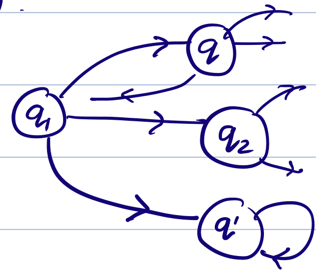
Drawing states and arrows gets messy

We use programs to describe automata!

Determinism / Nondeterminism



Deterministic automaton



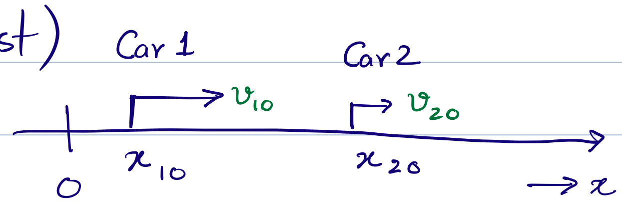
Non determinism

- Models arbitrary choice
- e.g. Failure, decisions

So given a state $q \in Q$ there are many possible "next" states.

Example 2 (Slides first)

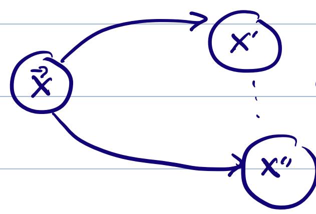
State vector



$\mathcal{S} = \mathbb{R}^4$

$\mathcal{S}_0 = \vec{x}_0 :=$

$\mathcal{A} \subseteq \mathbb{R}^4 \times \mathbb{R}^4$



How state changes in "1-Step"

E.g. if $x_2 - x_1 < d_s$

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

else $v_1 := v_1$

$x_2 := x_2 + v_2$

$x_1 := x_1 + v_1$

What does this really define?

$\mathcal{A} = \{\langle x, x' \rangle \in \mathbb{R}^4 \times \mathbb{R}^4 \mid$

This representation of $\mathcal{A}$ will be important when we want to work with sets of states

This example is deterministic

Ex Make it non deterministic

Execution

An execution is a particular behavior of $\mathcal{A}$

$\alpha = q_0 q_1 \cdots q_k \cdots$ finite or infinite

(i) $q_0 \in Q_0$

(ii) $\forall i \ (q_i, q_{i+1}) \in \mathcal{D}$

Non deterministic automata can have many executions e.g.

off, CL, $\boxed{\text{CL}}$, CH

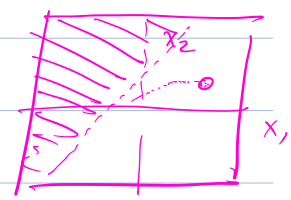
off, CL, $\boxed{\text{off}}$, CL

If α is an execution $\alpha = q_0 q_1 \cdots$ the i^{th} state in α is written as $\alpha[i] = q_i$

Requirements (Safety & Unsafety)

A requirement for $\mathcal{A}$ is any statement / (formula involving the states in Q) that is satisfied by all executions of A.

Example: • "Car 1 should never come within 0.5m of Car 2".



P_1 : "$x_2 - x_1 \geq 0.5$"

$P_1 = \{ x \in \mathbb{R}^4 \mid x.x_2 - x.x_1 \geq 0.5 \}$

- "Car should never exceed speed limit" : U_1
- " always $\geq$ 25 mpg" : P_3

With a single test / execution we can check whether every state in the execution satisfies the requirement or not.
OR equivalently does not satisfy the unsafety requirement (U).

To generate individual tests we can "just run" the program representing $\mathcal{S}$

But, to cover all behaviors/executions we have to work with sets of states.

Def 1 For $\mathcal{A} = \langle Q, Q_0, \mathcal{D} \rangle$ any set $S \subseteq Q$

$$Post(S) = \{$$

Exercise (monotonic)

Def 2

$$Post^0(S) := S$$

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

$\boxed{\text{Proposition. The set of states } R^k \text{ that } k \text{ can reach after } k\text{-transitions (at the end of executions of length } k) = Post^k(Q_0)}$

D * (ICYAI) In Case You Are Interested

Proof: R^k = States that are at the end of executions of length k.
We have to show (i) $R^k \subseteq Post^k(Q_0)$ and
(ii) $Post^k(Q_0) \subseteq R^k$.

(i) Proof by induction on k.

Base. $R^0 = Q_0$ by def of execution of length 0.

$Post^0(Q_0) = Q_0$ by def of $Post^k$

Therefore $R^0 \subseteq Post^0(Q_0)$

Induction. Suppose $R^k \subseteq Post^k(Q_0)$ — 1

Now consider R^{k+1}.

for any $x \in R^{k+1}$ $\exists$ execution of length k with last state $x' \in R^k$ by 1 and $(x', x) \in \mathcal{D}$

$\Rightarrow$ $x' \in Post(x) \subseteq Post(R^k)$

$\subseteq Post(Post^k(Q_0))$ by 1

$\Rightarrow$ $x' \subseteq Post^{k+1}(Q_0)$

$R^k \subseteq Post^k(Q_0)$ $\forall k$.

(ii) $Post^k(Q_0) \subseteq R^k$

Base case: Same as above

Induction: hypothesis $Post^k(Q_0) \subseteq R^k$ — 2

Now consider $x \in Post^{k+1}(Q_0)$
there must be α' an execution of length k
with last state x' in $Post^k(Q_0)$ by (2)
and $(x', x) \in \mathcal{D}$
Then $\alpha' x$ is an exec of length $k+1$ with
$\Rightarrow$ $x \in R^{k+1}$
$Post^{k+1}(Q) \subseteq R^{k+1}$ $\blacksquare$

To prove that all executions are safe forever we would want to compute

Idea. Approximate $Post^k(Q_0)$

Prop2 If we can find a set $I \subseteq Q$ such that

(i) $Q_0 \subseteq I$

(ii) $Post(I) \subseteq I$

$$\boxed{\text{Then } Post^{k}(Q_0) \subseteq I \quad \forall k.}$$

I over approximates $Post^{k}(Q_0)$

<u>Proof</u>. Induction on k.

$$\boxed{\text{Such an } I \text{ is called an Inductive invariant of } \mathcal{A}.}$$

- If we can find I and $I \cap Unsaf = \phi$

we have shown that <u>all</u> executions of $\mathcal{A}$ are safe.