Lecture 2: System-level Safety

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

MPO Walkthrough in lab tomorrow!
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 \( D \subseteq Q \times Q \)

**Example.**

\[
Q = \{c, b, s\} \quad Q_0 = \{c\} \\
D = \{(b, c), (c, b), (c, c), (c, s), (s, c)\}
\]

Non-deterministic.

- 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).

$$ D : Q \rightarrow \mathbb{P}(Q) $$

We can have both nondeterminism & probabilistic uncertainty → 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 $q_1 = q_2$

For deterministic automata

$$ D : Q \rightarrow Q $$ is a transition function.
Often D will be described by a program or a physics model (differential equations)

\[
\text{If } x_2 - x_1 < d, \quad v_i := \max(0, v_i - a_i) \\
\text{else } v_i := v_i \\
x_1 := x_1 + v_1 \\
x_2 := x_2 + v_2
\]

Do you see how this defines D? Is it deterministic?
With some abuse of notation we can represent nondeterministic models also as programs

\[ v_i := \text{Choose } [v_i - b_i, v_i - a_i] \]

Faulty Sensor

Execution. An execution is a particular behavior of the automaton \( A \).

\[ \alpha = q_0, q_1, q_2 \ldots \quad \text{finite or infinite} \]

Such that

(i) \( q_0 \in Q \).

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

Nondeterministic automata have many executions.

A “Test” \( \sim \) one Execution
Requirements of a design

Examples:

"Car 1 eventually catches up of C2"

"Car 1 and Car 2 never collide"

"Car 1 never goes backwards"

"Car 1 does not exceed speed limits"

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. $S_i := |x_1 - x_2| \geq 0.1$

2. A subset of $Q$

$$\{S_i \in Q \in \mathbb{R}^4 | \langle x_1, v_1, x_2, v_2 \rangle \leq \langle x_1, x_2 \rangle \geq 0.1 \}$$
Safety verification problem

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

Such an execution is called a counter-example

Def.
If for every finite execution $\alpha = q_0 \ldots q_k$ of $A$ and for every $q_i$ in $\alpha$, $q_i \in S$ then we say $A$ is safe w.r.t. $S$. 

\[ [s_1, s_2] \ldots \]
Reasoning about all executions

Def. For any set of states \( R \subseteq Q \)
\[
\text{Post}(R) := \{ q' \in Q \mid \exists q \in R \text{ and } \langle q, q' \rangle \in D \}
\]

Exercise The Post() operator is monotonic
if \( R_1 \subseteq R_2 \) then \( \text{Post}(R_1) \subseteq \text{Post}(R_2) \)

Proof. Choose any \( R_1 \subseteq R_2 \subseteq Q \)
Choose any \( x \in \text{Post}(R_1) \)
[we have to show \( x \in \text{Post}(R_2) \)]
By def of Post \( \exists x_0 \in R_1, \langle x_0, x \rangle \in D \)
Since \( R_1 \subseteq R_2 \) \( \Rightarrow x_0 \in R_2 \)
\( \Rightarrow x \in \text{Post}(R_2) \)

We can apply Post recursively.

Def \( \text{Post}^k : 2^Q \rightarrow 2^Q \)
\[
\text{Post}^0(R) = R \quad k = 0
\]
\[
\text{Post}^k(R) = \text{Post}(\text{Post}^{k-1}(R)) \quad k > 0
\]
Exercise \( \text{Post}^k(Q_0) \) is exactly the set of states that the automaton can reach after executions of length \( k \).

Proof.

\[
\left[ \bigcup_{k=0}^{\infty} \text{Post}^k(Q_0) \right] \cap \{s^c\} = \emptyset
\]

Reachability analysis tools can compute or over approximate \( \text{post}^k() \). E.g. Verse, Space Ex, Flow*

In general, computing \( \text{Post}^k(R) \) can be hard (1) \( R \) high dimensional (2) \( D \) complex, (3) \( k \) large
Alternative Solution to Safety Verification Problem

Find an inductive invariant for proving safety of $S$.

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

1. $Q_0 \subseteq I$
2. $\text{Post}(I) \subseteq I$

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 A
\( \alpha = a_0 a_1 \ldots a_k \). We will prove by induction on \( k \) that \( \forall i \, a_i \in I \).

**Base case.** \( k = 0 \) \( \alpha = a_0 \in Q_0 \subseteq I \) by (1)

**Inductive step.** \( \alpha = a_0 \ldots a_{k-1} a_k \) and \( a_{k-1} \in I \). We will show \( a_k \in I \).

By (2) Post\((I) \subseteq I \) as \( a_{k-1} \in I \Rightarrow a_k \in I \)

Therefore \( \forall i \, a_i \in I \).

Further if \( I \subseteq S \) then \( \forall i \, a_i \in S \)
Simple invariant and safety

\[ S_1 := q_1 > 0 \]

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) \ldots (3)?

(1) \( \forall q_o \in Q_o \)
\[ q_o \cdot v_{i0} > 0 \quad [\text{we assumed this}] \]
\[ \Rightarrow q_o \in [S_1] \]

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

For any state \( q \in I \) if \( q \cdot v_1 > 0 \)
and \( (q, q') \in D \) then we have to show that \( q' \cdot v_1 \) is also \( > 0 \)
How are \( q \) and \( q' \) related by \( D \)?

Note

\[
\begin{align*}
q', v_i &= \max(0, q \cdot v_i - a_b) \\
\geq 0 \\
q' \in [s, l] &= I
\end{align*}
\]

Post \((I) \leq I\)

It follows that \( S_1 \) is indeed an invariant; Carl never goes backwards.

Another Safety requirement

\( S_2 : x_1 < x_2 \)

Is \( S_2 \) an inductive invariant?

(1) \( Q_0 \leq S_2 \implies 0 < x_{i0} < x_{20} \)

(2) \( \text{Post}(S_2) \leq S_2 \)?
Not necessarily true
if \( q \cdot v_1 \gg q \cdot x_2 - q \cdot x_1 \)
then \( q' \cdot x_1 \) may exceed \( q' \cdot x_2 \)

We cannot prove \( S_2 \)
we need to add or “discover”
assumptions about \( ds, ab \) ...
to prove \( S_2 \).

Add more information in the model

\[
\text{timer} := 0
\]
\[
\text{If } x_2 - x_1 < ds \quad \begin{array}{l}
\text{if } v_1 > ab \text{ then} \\
\quad v_1 := v_1 - ab \\
\quad \text{timer} := \text{timer} + 1 \end{array} \quad \begin{array}{l}
\{ \text{A} \} \\
\text{else } v_1 := 0 \quad \{ \text{B} \} \\
\text{else } v_1 := v_1 \quad \{ \text{C} \}
\end{array}
\]

\[
x_1 := x_1 + v_1
\]
\[
x_2 := x_2 + v_2
\]
(1) \( q_0 \cdot \text{timer} = 0 \leq \frac{v_{10} - v_i}{a_b} \leq 0 \)

(2) \( q \in I_3 \implies q' \in I_3 \)

Three cases to consider

A. if \( q \cdot x_2 - q \cdot x_1 < d_s \) and \( q \cdot v_i > a_b \)

then \( q' \cdot \text{timer} = q \cdot \text{timer} + 1 \)

\[
\leq \frac{v_{10} - q \cdot v_i}{a_b} + 1 \quad [\text{ind hyp}]
\]

\[= \frac{v_{10} - (q' \cdot v_i + a_b)}{a_b} + 1 \]

\[= \frac{v_{10} - q' \cdot v_i}{a_b} \]

B. if \( q \cdot x_2 - q \cdot x_1 < d_s \) and \( q \cdot v_i \leq a_b \)

\( q' \cdot \text{timer} = q \cdot \text{timer} \)

\[
\leq \frac{v_{10} - q \cdot v_i}{a_b} + 1
\]
\[ \leq \frac{v_{10} + 0}{a_b} + 1 \]

C. if \( q. x_2 - q. x_1 \geq d_s \)

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

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

\[ I_3 : \text{timer} \leq \frac{v_{10} - v_1}{a_b} \] and \( v_1 \geq 0 \)

\[ \Rightarrow \text{timer} \leq \frac{v_{10}}{a_b} \]

Still not enough to prove \( x_2 - x_1 > 0 \)

Max distance traversed by Carl after defection \( \leq v_{10} \cdot \text{timer} \leq \frac{v_{10}^2}{a_b} \)

So, if \( d_s > \frac{v_{10}^2}{a_b} \) and \( v_2 \geq 0 \)

then \( I_3 \Rightarrow S_0 : x_2 > x_1 \)
That is, if $d_s > \frac{\eta^2}{q_k}$ then in any execution of $\mathcal{A}$ and all states $q_1, \ldots, q_k$ in $x$, $q_i.x_2 > q_i.x_1$. 