Model Verification
(Temporal Properties)

Juri Kolčák

Friday 12th December, 2025

Formal Verification

Given a model of a dynamical system (complex system);

And a set of desirable properties (specifications, observations, ...);

The goal is to determine whether the model satisfies the properties.

Automatic Verification:

Testing – Identify critical scenarios to test model executions on.

Non-exhaustive: if an execution fails a test, we know the model does not satisfy our properties, but if all succeed, we cannot rule out there is another execution which would fail.

Static Analysis – Avoids exploration of dynamics (transition system).

Typically provides only partial results (approximation).

Dynamic Analysis – Exhaustive exploration of the transition system.

Formal reasoning about dynamic properties (evolution in time) is possible using **temporal logics**. Such temporal properties can be automatically verified by **model checking**.

Temporal Logic

Formal (unambiguous) reasoning about properties related to the successive change of system states (variables).

Temporal logics we focus on:

- Linear Temporal Logic (**LTL**) – Reasoning on traces of the model.
- Computational Tree Logic (**CTL**) – Reasoning on execution trees.
- (**CTL*** – CTL enriched to be able to express everything that's possible in LTL.)

Other temporal logics:

- Higher expressivity – allows more complex properties.
- Include time – allows specification of time-bound properties.
- Probabilities – allows properties about probabilities of behaviours.

Traces

For a transition system $(S, \rightarrow)$, a trace is an (infinite) sequence of configurations $\sigma = (\mathbf{x}^0, \mathbf{x}^1, \mathbf{x}^2, \dots)$ such that $\forall k > 0$, $\mathbf{x}^{k-1} \rightarrow \mathbf{x}^k$.

For a configuration $\mathbf{x} \in \mathbb{B}^n$, let $\mathcal{S}(\mathbf{x})$ be the set of all traces that originate in $\mathbf{x}$.

Traces

For a transition system $(S, \rightarrow)$, a trace is an (infinite) sequence of configurations $\sigma = (\mathbf{x}^0, \mathbf{x}^1, \mathbf{x}^2, \dots)$ such that $\forall k > 0$, $\mathbf{x}^{k-1} \rightarrow \mathbf{x}^k$.

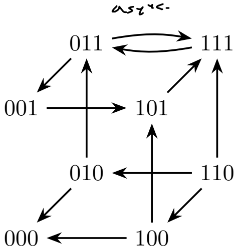
For a configuration $\mathbf{x} \in \mathbb{B}^n$, let $\mathcal{S}(\mathbf{x})$ be the set of all traces that originate in $\mathbf{x}$.

EXAMPLE:

$f_1(\mathbf{x}) = \mathbf{x}_3 \wedge (\neg \mathbf{x}_1 \vee \neg \mathbf{x}_2)$

$f_2(\mathbf{x}) = \mathbf{x}_1 \wedge \mathbf{x}_3$

$f_3(\mathbf{x}) = \mathbf{x}_1 \vee \mathbf{x}_2 \vee \mathbf{x}_3$



$\sigma_1 = (110, 100, 000)$

$\sigma_2 = (101, [111, 011]^\omega)$

$\sigma_3 = (001, 101, 111)$

$\dots$

Traces

For a transition system $(S, \to)$, a trace is an (infinite) sequence of configurations $\sigma = (\mathbf{x}^0, \mathbf{x}^1, \mathbf{x}^2, \dots)$ such that $\forall k > 0$, $\mathbf{x}^{k-1} \to \mathbf{x}^k$.

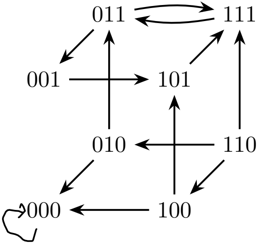
For a configuration $\mathbf{x} \in \mathbb{B}^n$, let $\mathcal{S}(\mathbf{x})$ be the set of all traces that originate in $\mathbf{x}$.

EXAMPLE:

$f_1(\mathbf{x}) = \mathbf{x}_3 \wedge (\neg \mathbf{x}_1 \vee \neg \mathbf{x}_2)$

$f_2(\mathbf{x}) = \mathbf{x}_1 \wedge \mathbf{x}_3$

$f_3(\mathbf{x}) = \mathbf{x}_1 \vee \mathbf{x}_2 \vee \mathbf{x}_3$



$\sigma_1 = (110, 100, 000)$

$\sigma_2 = (101, [111, 011]^\omega)$

$\sigma_3 = (001, 101, 111)$

$\dots$

Sometimes it's useful to consider only infinite traces. To preserve reachability, we include the transition $\mathbf{x} \to \mathbf{x}$ for each fixed point $\mathbf{x}$.

Execution Trees

"All traces from $\mathcal{S}(\mathbf{x})$ bundled by prefixes."

Formally, a connected acyclic graph (V, E) with a labelling function $\lambda\colon V \to \mathbb{B}^n$ mapping the vertices to the configurations.

"Unfolding of the transition system."

Given an initial configuration $\mathbf{x}$, the execution tree can be intuited inductively as follows:

1. Add the root v_r to V with $\lambda(v_r) = \mathbf{x}$ and initialise the set of unprocessed vertices $V' = \{v_r\}$;
2. While V' is not empty, take $v \in V'$ and for each $\mathbf{y} \in \mathbb{B}^n$ such that $\lambda(v) \to \mathbf{y}$, add a new node v' with $\lambda(v') = \mathbf{y}$ to both V, V', and an edge (v, v') to E;

Execution Trees

"All traces from $\mathcal{S}(\mathbf{x})$ bundled by prefixes."

Formally, a connected acyclic graph (V, E) with a labelling function $\lambda\colon V \to \mathbb{B}^n$ mapping the vertices to the configurations.

"Unfolding of the transition system."

Given an initial configuration $\mathbf{x}$, the execution tree can be intuited inductively as follows:

1. Add the root v_r to V with $\lambda(v_r) = \mathbf{x}$ and initialise the set of unprocessed vertices $V' = \{v_r\}$;
2. While V' is not empty, take $v \in V'$ and for each $\mathbf{y} \in \mathbb{B}^n$ such that $\lambda(v) \to \mathbf{y}$, add a new node v' with $\lambda(v') = \mathbf{y}$ to both V, V', and an edge (v, v') to E;

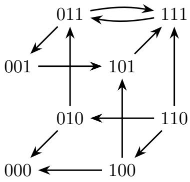
Comparison to traces:

For a given root, the execution tree is unique.

A trace corresponds to a path in the tree starting from the root.

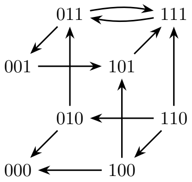
Execution Trees – Example

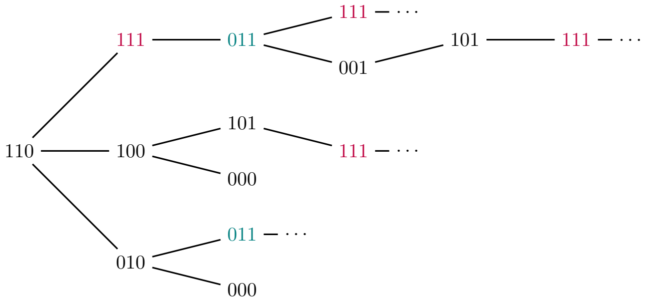
$$f_1(\mathbf{x}) = \mathbf{x}_3 \wedge (\neg \mathbf{x}_1 \vee \neg \mathbf{x}_2)$$
$$f_2(\mathbf{x}) = \mathbf{x}_1 \wedge \mathbf{x}_3$$
$$f_3(\mathbf{x}) = \mathbf{x}_1 \vee \mathbf{x}_2 \vee \mathbf{x}_3$$



Execution Trees – Example

$$f_1(\mathbf{x}) = \mathbf{x}_3 \wedge (\neg\mathbf{x}_1 \vee \neg\mathbf{x}_2)$$
$$f_2(\mathbf{x}) = \mathbf{x}_1 \wedge \mathbf{x}_3$$
$$f_3(\mathbf{x}) = \mathbf{x}_1 \vee \mathbf{x}_2 \vee \mathbf{x}_3$$





Atomic Propositions

"A set of properties that characterise the configurations of the system."

Formally, we define a finite set of atomic propositions $P = \{p_0, \ldots, p_k\}$, $k \in \mathbb{N}$ and a mapping $\alpha \colon \mathbb{B}^n \to 2^P$ which maps each configuration to a set of atomic propositions "valid" in the configuration.

2^P = Power set of P

A configuration $\mathbf{x} \in \mathbb{B}^n$ satisfies a proposition $p \in P$, $\mathbf{x} \models p$, if and only if $p \in \alpha(\mathbf{x})$.

EXAMPLES:

- $\mathbf{x}_i$ "Variable i is active";
- $\mathbf{x}_i + \mathbf{x}_j + \mathbf{x}_k \geq 2$ "At least two of the variables i, j, k are active";
- $\forall i \in W \subseteq \{1, \ldots, n\}$, $\mathbf{x}_i = 0$ "None of the variables in W is active";
- $\mathbf{x} \in A$ "Is part of the attractor A";

Linear Temporal Logic

Syntax:

Next, until, Future, Globally

$\bigcirc\varphi$ $\Diamond\varphi$ $\Box\varphi$

$$\varphi ::= \top \mid p \in P \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \mathbf{X}\varphi \mid \varphi_1 \,\mathbf{U}\, \varphi_2 (\mid \mathbf{F}\varphi \mid \mathbf{G}\varphi)$$

$\varphi_1 \vee \varphi_2 = \neg(\neg\varphi_1 \wedge \neg\varphi_2)$ $\quad$ $F\varphi = \top U \varphi$ $\quad$ $G\varphi = \neg F(\neg\varphi)$

Semantics:

$\sigma = (x^0, x^1, x^2, \dots)$

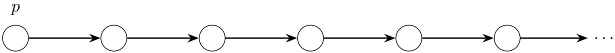
$$
\begin{array}{lcl}
\sigma \models \top & & \\
\sigma \models p & \Leftrightarrow & \mathbf{x}^0 \models p \\
\sigma \models \neg\varphi & \Leftrightarrow & \sigma \not\models \varphi \\
\sigma \models \varphi_1 \wedge \varphi_2 & \Leftrightarrow & \sigma \models \varphi_1 \wedge \sigma \models \varphi_2 \\
\sigma \models \mathbf{X}\varphi & \Leftrightarrow & (\mathbf{x}^1, \mathbf{x}^2, \dots) \models \varphi \\
\sigma \models \varphi_1 \,\mathbf{U}\, \varphi_2 & \Leftrightarrow & \exists k \in \mathbb{N}_0, (\mathbf{x}^k, \mathbf{x}^{k+1}, \dots) \models \varphi_2 \wedge \\
 & & \wedge \forall j < k, (\mathbf{x}^j, \mathbf{x}^{j+1}, \dots) \models \varphi_1 \\
\sigma \models \mathbf{F}\varphi & \Leftrightarrow & \exists k \in \mathbb{N}_0, (\mathbf{x}^k, \mathbf{x}^{k+1}, \dots) \models \varphi \\
\sigma \models \mathbf{G}\varphi & \Leftrightarrow & \forall k \in \mathbb{N}_0, (\mathbf{x}^k, \mathbf{x}^{k+1}, \dots) \models \varphi
\end{array}
$$

$\neg(\exists k \in \mathbb{N}_0, (x^k, x^{k+1}, \dots) \models \neg\varphi)$

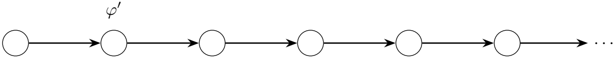
$\forall k \in \mathbb{N}_0, (x^k, x^{k+1}, \dots) \models \neg\neg\varphi$

LTL Visually

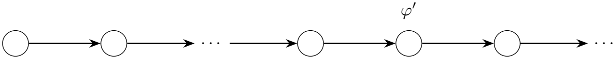
$\varphi = p$



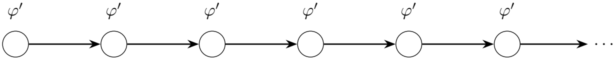
$\varphi = \mathbf{X}\varphi'$



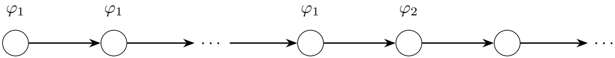
$\varphi = \mathbf{F}\varphi'$



$\varphi = \mathbf{G}\varphi'$

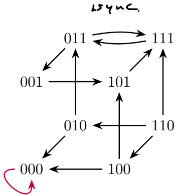


$\varphi = \varphi_1 \, \mathbf{U} \, \varphi_2$



LTL Example

$$f_1(\mathbf{x}) = \mathbf{x}_3 \wedge (\neg \mathbf{x}_1 \vee \neg \mathbf{x}_2)$$
$$f_2(\mathbf{x}) = \mathbf{x}_1 \wedge \mathbf{x}_3$$
$$f_3(\mathbf{x}) = \mathbf{x}_1 \vee \mathbf{x}_2 \vee \mathbf{x}_3$$



$x \models \varphi \iff \forall \sigma \in S(x), \sigma \models \varphi$

EXAMPLES:

- $\mathbf{F}(\neg \mathbf{x}_1)$ – "Eventually, variable 1 will be inactive"; $\mathbb{B}^3$
- $\mathbf{G}(\mathbf{x}_3)$ – "Variable 3 always stays active"; $\subseteq \{001, 101, 011, 111\}$
- $\mathbf{F}(\mathbf{G}(\mathbf{x}_3))$ – "Eventually, variable 3 will activate and will stay active forever";
- $\mathbf{G}(\mathbf{F}(\neg \mathbf{x}_2))$ – "Variable 2 will be inactive infinitely often"; $\{000\}$
- $\mathbf{x}_1 \, \mathbf{U} \, \mathbf{x}_3$ – "Variable 1 stays active until variable 3 becomes active";

$\{001, 101, 011, 111\}$

Computational Tree Logic

SYNTAX:

state formula $\Phi ::= \top \mid p \in P \mid \neg\Phi \mid \Phi_1 \wedge \Phi_2 \mid \exists\varphi \mid \forall\varphi$

path formula $\varphi ::= \mathbf{X}\Phi \mid \Phi_1 \,\mathbf{U}\, \Phi_2 (\mid \mathbf{F}\Phi \mid \mathbf{G}\Phi)$

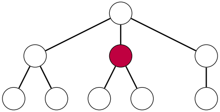
SEMANTICS:

$$
\begin{array}{lcl}
\mathbf{x} \models \top & & \\
\mathbf{x} \models p & \Leftrightarrow & \mathbf{x} \models p \\
\mathbf{x} \models \neg\Phi & \Leftrightarrow & \mathbf{x} \not\models \Phi \\
\mathbf{x} \models \Phi_1 \wedge \Phi_2 & \Leftrightarrow & \mathbf{x} \models \Phi_1 \wedge \mathbf{x} \models \Phi_2 \\
\mathbf{x} \models \exists\varphi & \Leftrightarrow & \exists\sigma \in \mathcal{S}(\mathbf{x}), \sigma \models \varphi \\
\mathbf{x} \models \forall\varphi & \Leftrightarrow & \forall\sigma \in \mathcal{S}(\mathbf{x}), \sigma \models \varphi
\end{array}
$$

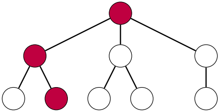
$$
\begin{array}{lcl}
\sigma \models \mathbf{X}\Phi & \Leftrightarrow & \mathbf{x}^1 \models \Phi \\
\sigma \models \Phi_1 \,\mathbf{U}\, \Phi_2 & \Leftrightarrow & \exists k \in \mathbb{N}_0, \mathbf{x}^k \models \Phi_2 \wedge \forall j < k, \mathbf{x}^j \models \Phi_1 \\
\sigma \models \mathbf{F}\Phi & \Leftrightarrow & \exists k \in \mathbb{N}_0, \mathbf{x}^k \models \Phi \\
\sigma \models \mathbf{G}\Phi & \Leftrightarrow & \forall k \in \mathbb{N}_0, \mathbf{x}^k \models \Phi
\end{array}
$$

CTL Visually

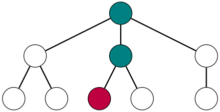
$\exists \mathbf{F} \textit{red}$



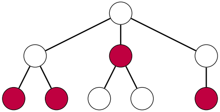
$\exists \mathbf{G} \textit{red}$



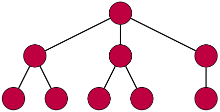
$\exists \textit{teal}\ \mathbf{U}\ \textit{red}$



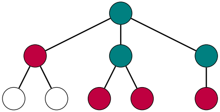
$\forall \mathbf{F} \textit{red}$



$\forall \mathbf{G} \textit{red}$

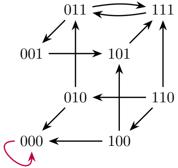


$\forall \textit{teal}\ \mathbf{U}\ \textit{red}$



CTL Example

$$f_1(\mathbf{x}) = \mathbf{x}_3 \wedge (\neg \mathbf{x}_1 \vee \neg \mathbf{x}_2)$$
$$f_2(\mathbf{x}) = \mathbf{x}_1 \wedge \mathbf{x}_3$$
$$f_3(\mathbf{x}) = \mathbf{x}_1 \vee \mathbf{x}_2 \vee \mathbf{x}_3$$



EXAMPLES:

- $\forall \mathbf{F}(\neg \mathbf{x}_1)$ – "Eventually, variable 1 will be inactive"; $\mathbb{B}^3$
- $\forall \mathbf{G}(\mathbf{x}_3)$ – "Variable 3 always stays active"; $\{001, 101, 111, 011\}$
- $\exists \mathbf{F}(\forall \mathbf{G}(\mathbf{x}_3))$ – "There exists a path to a state in which variable 3 is active and cannot be deactivated"; $\mathbb{B}^3 \setminus \{000\}$
- $\forall \mathbf{G}(\exists \mathbf{F}(\neg \mathbf{x}_2))$ – "Along any path, it is always possible to deactivate variable 2"; $\mathbb{B}^3$
- $\exists \mathbf{x}_1 \, \mathbf{U} \, \mathbf{x}_3$ – "There exists a path along which variable 1 stays active until variable 3 becomes active"; $\mathbb{B}^3 \setminus \{000, 010\}$

CTL*

SYNTAX:

$$\Phi ::= \top \mid p \in P \mid \neg\Phi \mid \Phi_1 \wedge \Phi_2 \mid \exists\varphi \mid \forall\varphi$$
$$\varphi ::= \Phi \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \mathbf{X}\varphi \mid \varphi_1 \,\mathbf{U}\, \varphi_2$$

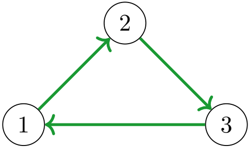
SEMANTICS:

$$\begin{array}{lcl}
\mathbf{x} \models \top & & \\
\mathbf{x} \models p & \Leftrightarrow & \mathbf{x} \models p \\
\mathbf{x} \models \neg\Phi & \Leftrightarrow & \mathbf{x} \not\models \Phi \\
\mathbf{x} \models \Phi_1 \wedge \Phi_2 & \Leftrightarrow & \mathbf{x} \models \Phi_1 \wedge \mathbf{x} \models \Phi_2 \\
\mathbf{x} \models \exists\varphi & \Leftrightarrow & \exists\sigma \in \mathcal{S}(\mathbf{x}), \sigma \models \varphi \\
\mathbf{x} \models \forall\varphi & \Leftrightarrow & \forall\sigma \in \mathcal{S}(\mathbf{x}), \sigma \models \varphi
\end{array}$$

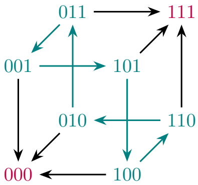
$$\begin{array}{lcl}
\sigma \models \Phi & \Leftrightarrow & \mathbf{x}^0 \models \Phi \\
\sigma \models \neg\varphi & \Leftrightarrow & \sigma \not\models \varphi \\
\sigma \models \varphi_1 \wedge \varphi_2 & \Leftrightarrow & \sigma \models \varphi_1 \wedge \sigma \models \varphi_2 \\
\sigma \models \mathbf{X}\varphi & \Leftrightarrow & (\mathbf{x}^1, \mathbf{x}^2, \dots) \models \varphi \\
\sigma \models \varphi_1 \,\mathbf{U}\, \varphi_2 & \Leftrightarrow & \exists k \in \mathbb{N}_0, (\mathbf{x}^k, \mathbf{x}^{k+1}, \dots) \models \varphi_2 \wedge \\
 & & \wedge \forall j < k, (\mathbf{x}^j, \mathbf{x}^{j+1}, \dots) \models \varphi_1
\end{array}$$

Fairness

EXAMPLE:



$$f_1(\mathbf{x}) = \mathbf{x}_3$$
$$f_2(\mathbf{x}) = \mathbf{x}_1$$
$$f_3(\mathbf{x}) = \mathbf{x}_2$$



Fairness constraints are put in place to make sure "everybody gets their turn".

Different fairness constraints are used, depending on the application scenario, but the general intuition is that if a transition can be taken infinitely often along a run, then it will eventually be traversed.