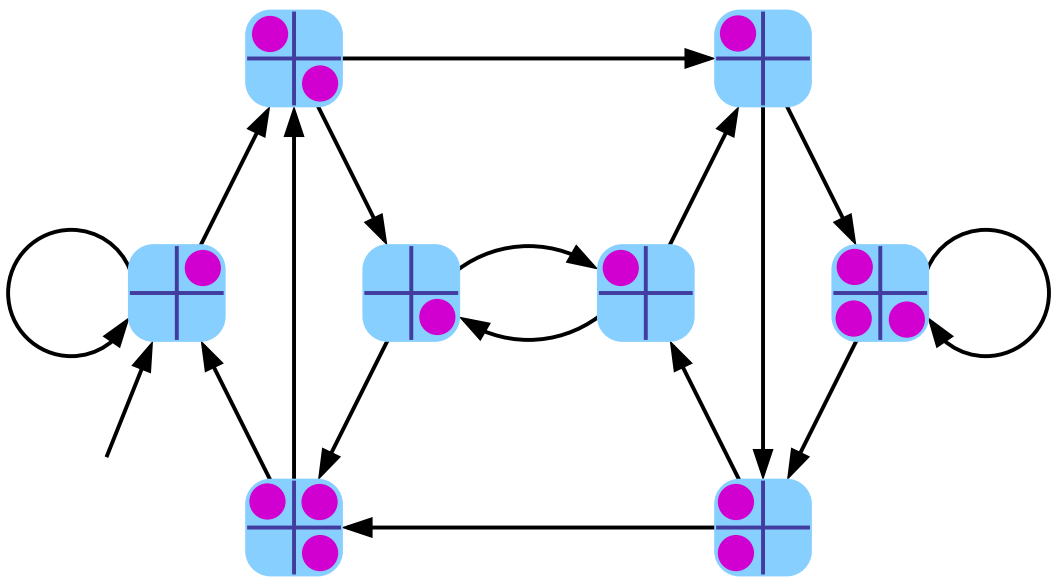
Overview

Model Checking

Technique for automated verification of

- hardware
- software
- communication protocols

Modeleled as labelled transition systems,
properties described and verified in temporal logic.



Labelled transition systems

A labelled transition system $M = (S, I, T, \ell)$ consists of

- a finite set S of states,
- a subset $I \subseteq S$ of initial states,
- a transition relation $T \subseteq S \times S$,
- a labelling function $\ell : S \to 2^P$,
  where P is a finite set of elementary propositions.

$p \in \ell(s)$ for $s \in S$ and $p \in P$ means:

Proposition p holds in state s.

Linear temporal logic: syntax

Formulas of *LTL* are inductively defined:

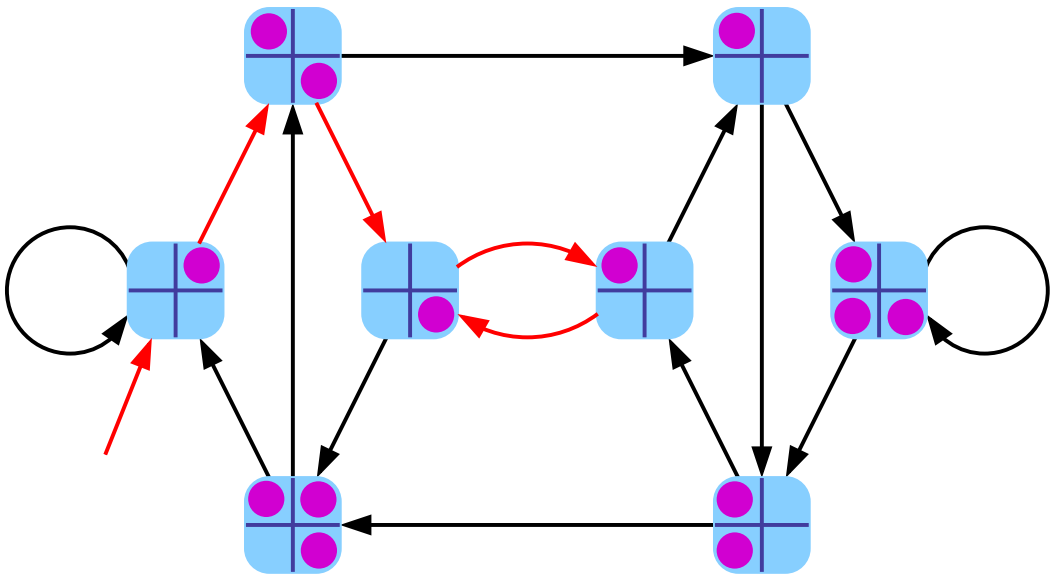
- The constants $\mathtt{tt}$ and $\mathtt{ff}$ are formulas.
- p und $\neg p$ are formulas, for every proposition $p \in P$.
- If φ is a formula, then $\mathtt{X}\varphi$, $\mathtt{F}\varphi$ and $\mathtt{G}\varphi$ are formulas.
- If φ and ψ are formulas,
  then $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \,\mathtt{U}\, \psi$ and $\varphi \,\mathtt{R}\, \psi$ are formulas.

Note that formulas of *LTL* are in *NNF* by definition.

For simplicity we consider only the fragment without $\mathtt{U}$ and $\mathtt{R}$.

Model Checking *LTL*: Example

exists a path satisfying $\mathtt{G}(\mathrm{UL} \vee \mathtt{X}\,\mathrm{UL})$



Linear temporal logic: semantics

A path π in M is an infinite sequence $s_0, s_1, s_2, \ldots$ of states s.t. $T(s_i, s_{i+1})$ for all i.

For a path π and $i \in \mathbb{N}$ let π^i denote the suffix $s_i, s_{i+1}, s_{i+2}, \ldots$.

The statement that $\pi \models \varphi$ is defined inductively:

$$
\begin{array}{ll}
\pi \models p & \text{iff} \quad p \in \ell(s_0) \\
\pi \models \neg p & \text{iff} \quad p \notin \ell(s_0) \\
\pi \models \mathtt{X}\varphi & \text{iff } \pi^1 \models \varphi \\
\pi \models \mathtt{F}\varphi & \text{iff } \pi^i \models \varphi \text{ for some } i \geq 0 \\
\pi \models \mathtt{G}\varphi & \text{iff } \pi^i \models \varphi \text{ for all } i \geq 0
\end{array}
$$

$M \models \varphi$ holds if $\pi \models \varphi$ for some path π in M with $s_0 \in I$.

Classical model checking

Automata-based Model Checking (Vardi and Wolper 1986)

- Translate φ to Büchi automaton A_φ
- Compute product automaton $M \times A_\varphi$
- Use graph algorithms to find accepting run

Problem: state space explosion!

Symbolic model checking

Symbolic Model Checking (McMillan et al. 1990)

- Represent states of M as bitvectors
- transitions, propositions etc. as boolean functions
  - $f_I : \{0,1\}^n \to \{0,1\}$ with $f_I(s) = 1$ iff $s \in I$,
  - $f_T : \{0,1\}^{2n} \to \{0,1\}$ with $f_T(s,s') = 1$ iff $(s,s') \in T$,
  - $f_p : \{0,1\}^n \to \{0,1\}$ for each $p \in P$, with $f_p(s) = 1$ iff $p \in \ell(s)$.
- can be stored succinctly as OBDDs
- use efficient OBDD algorithms to evaluate φ

Bounded model checking

Symbolic model checking

$\rightsquigarrow$ transition system represented in propositional logic

Idea: Take advantage of progress in SAT solvers

Problem with Complexity: *LTL* model checking is PSPACE-complete

Solution: Bounded model checking (Biere et al., 1999)

$\exists$ path of length k satisfying φ?

in NP $\rightsquigarrow$ reducible to SAT

Bounded Semantics for LTL

A k-loop is a path $\pi = s_1, \ldots, s_k, s_{k+1}, \ldots$
with $s_{k+1+i} = s_{\ell+i}$ for some $\ell \leq k$ and all $i \in \mathbb{N}$.

If π is a k-loop, then $\pi \models_k \varphi$ iff $\pi \models \varphi$.

If π is not a k-loop, then $\pi \models_k \varphi$ iff $\pi \models_k^0 \varphi$,
where $\pi \models_k^i \psi$ is inductively defined by:

$$
\begin{array}{ll}
\pi \models_k^i p & \text{iff } p \in \ell(s_i) \\
\pi \models_k^i \neg p & \text{iff } p \notin \ell(s_i) 0 \\
\pi \models_k^i \mathtt{X}\varphi & \text{iff } i < k \text{ und } \pi \models_k^{i+1} \varphi \\
\pi \models_k^i \mathtt{F}\varphi & \text{iff } \pi \models_k^j \varphi \text{ for some } j \text{ with } i \leq j \leq k \\
\pi \models_k^i \mathtt{G}\varphi & \text{never holds}
\end{array}
$$

Properties of the bounded semantics

Underapproximation: if $\pi \models_k \varphi$, then $\pi \models \varphi$.

Theorem

For every M and LTL formula φ there is k

s.t. $M \models \varphi$ iff $M \models_k \varphi$

Usage of Bounded Model Checking:

- Given desired property φ
- Check that $M \models_k \neg\varphi$.
- If yes: counterexample found.
- If no: increase k and repeat.

Propositional Encoding

Define propositional formula $\langle\!\langle\, M, \varphi \,\rangle\!\rangle^k$ in variables $s_1, \ldots, s_k$, where each s_i is $s_{i,1}, \ldots, s_{i,n}$.

Assignment to these variables $\quad\widehat{=}\quad$ sequence of states in M.

$\langle\!\langle\, M, \varphi \,\rangle\!\rangle^k$ satisfiable iff $M \models_k \varphi$,
satisfying assignment $\quad\widehat{=}\quad$ path π with $\pi \models_k \varphi$.

Propositional Encoding: Outer Structure

$\langle\!\langle\, M, \varphi \,\rangle\!\rangle^k$ is a conjunction $\langle\!\langle\, M \,\rangle\!\rangle^k \wedge \langle\!\langle\, \varphi \,\rangle\!\rangle^k$

$\langle\!\langle\, M \,\rangle\!\rangle^k$ enforces that $s_1, \ldots, s_k$ form a path π in M:

$$\langle\!\langle\, M \,\rangle\!\rangle^k := I(s_1) \wedge \bigwedge_{1 \leq i < k} T(s_i, s_{i+1})$$

$L_{k,\ell} := T(s_k, s_\ell)$ expresses: π is a (k, ℓ)-loop,

and $L_k := \bigvee_{\ell \leq k} L_{k,\ell}$: π is a k-loop.

$\langle\!\langle\, \varphi \,\rangle\!\rangle^k$ enforces that $\pi \models_k \varphi$:

$$\langle\!\langle\, \varphi \,\rangle\!\rangle^k := \left(\neg L_k \wedge \langle\!\langle\, \varphi \,\rangle\!\rangle_1^k\right) \vee \bigvee_{1 \leq \ell \leq k} \left(L_{k,\ell} \wedge \langle\!\langle\, \varphi \,\rangle\!\rangle_1^{k,\ell}\right)$$

Propositional Encoding: Non-Loop Case

Formulas $\langle\!\langle\, \psi \,\rangle\!\rangle_i^k$ expressing $\pi \models_k^i \psi$
on non-loops are defined inductively:

$$
\begin{aligned}
\langle\!\langle\, p \,\rangle\!\rangle_i^k &:= p(s_i) \\
\langle\!\langle\, \neg p \,\rangle\!\rangle_i^k &:= \neg p(s_i) \\
\langle\!\langle\, \mathtt{X}\psi \,\rangle\!\rangle_i^k &:= \begin{cases} \langle\!\langle\, \psi \,\rangle\!\rangle_{i+1}^k & \text{if } i < k \\ 0 & \text{otherwise.} \end{cases} \\
\langle\!\langle\, \mathtt{F}\psi \,\rangle\!\rangle_i^k &:= \bigvee_{i \le j \le k} \langle\!\langle\, \psi \,\rangle\!\rangle_j^k \\
\langle\!\langle\, \mathtt{G}\psi \,\rangle\!\rangle_i^k &:= 0
\end{aligned}
$$

Propositional Encoding: Loop Case

Formulas $\langle\!\langle\, \psi \,\rangle\!\rangle_i^{k,\ell}$ expressing $\pi^i \models_k \psi$ on a (k,ℓ)-loop are defined inductively:

$$
\begin{aligned}
\langle\!\langle\, p \,\rangle\!\rangle_i^{k,\ell} &:= p(s_i) \\
\langle\!\langle\, \neg p \,\rangle\!\rangle_i^{k,\ell} &:= \neg p(s_i) \\
\langle\!\langle\, \mathtt{X}\psi \,\rangle\!\rangle_i^{k,\ell} &:= \begin{cases} \langle\!\langle\, \psi \,\rangle\!\rangle_{i+1}^{k,\ell} & \text{if } i < k \\ \langle\!\langle\, \psi \,\rangle\!\rangle_{\ell}^{k,\ell} & \text{if } i = k \end{cases} \\
\langle\!\langle\, \mathtt{F}\psi \,\rangle\!\rangle_i^{k,\ell} &:= \bigvee_{\min(i,\ell) \le j \le k} \langle\!\langle\, \psi \,\rangle\!\rangle_j^{k,\ell} \\
\langle\!\langle\, \mathtt{G}\psi \,\rangle\!\rangle_i^{k,\ell} &:= \bigwedge_{\min(i,\ell) \le j \le k} \langle\!\langle\, \psi \,\rangle\!\rangle_j^{k,\ell}
\end{aligned}
$$

Optimization: getting rid of $\neg L_k$

Lemma

For all φ, *and all* $1 \leq i, \ell \leq k$,

$$\langle\!\langle\, \varphi \,\rangle\!\rangle_i^k \text{ implies } \langle\!\langle\, \varphi \,\rangle\!\rangle_i^{k,\ell}$$

Corollary

The formula $\langle\!\langle\, \varphi \,\rangle\!\rangle^k$ *is equivalent to*

$$\langle\!\langle\, \varphi \,\rangle\!\rangle_1^k \vee \bigvee_{1 \leq \ell \leq k} \left(L_{k,\ell} \wedge \langle\!\langle\, \varphi \,\rangle\!\rangle_1^{k,\ell} \right)$$

Implicit Loop formulas

Improved translation:

- loop variables λ_i state that π is a (k, i)-loop.
- only one formula translation depending on loop variables.
- loop constraints to determine shape of the loop.

The translation is $\langle\!\langle\, M, \varphi \,\rangle\!\rangle^k := \langle\!\langle\, M \,\rangle\!\rangle^k \wedge \mathrm{LOOP}_k \wedge \langle\!\langle\, \varphi \,\rangle\!\rangle_1^k$

The Loop constraints

$\textsc{Loop}_k$ is a conjunction $L_k \wedge A_k$

L_k determines the form of the loop:

$$L_k := \bigwedge_{i=1}^{k} \left(\lambda_i \rightarrow T(s_k, s_i)\right)$$

A_k states that at most one λ_i is true, using new variables σ_i.

$$A_k := \bigwedge_{i=1}^{k} \left(\sigma_i \rightarrow \neg\lambda_i\right)$$
$$\wedge \neg\sigma_1 \wedge \bigwedge_{i=1}^{k-1} \left(\sigma_{i+1} \leftrightarrow \sigma_i \vee \lambda_i\right)$$

Encoding of the bounded semantics

$$\langle\!\langle\, \mathtt{X}\psi \,\rangle\!\rangle_i^k := \begin{cases} \langle\!\langle\, \psi \,\rangle\!\rangle_{i+1}^k & \text{if } i < k \\ \bigvee_{j=1}^k \left(\lambda_j \wedge \langle\!\langle\, \psi \,\rangle\!\rangle_j^k\right) & \text{if } i = k \end{cases}$$

$$\langle\!\langle\, \mathtt{F}\psi \,\rangle\!\rangle_i^k := \begin{cases} \langle\!\langle\, \psi \,\rangle\!\rangle_i^k \vee \langle\!\langle\, \mathtt{F}\psi \,\rangle\!\rangle_{i+1}^k & \text{if } i \leq k \\ \bigvee_{j=1}^k \left(\lambda_j \wedge \langle\!\langle\, \mathtt{F}\psi \,\rangle\!\rangle_j^{k,2}\right) & \text{if } i = k+1 \end{cases}$$

$$\langle\!\langle\, \mathtt{G}\psi \,\rangle\!\rangle_i^k := \begin{cases} \langle\!\langle\, \psi \,\rangle\!\rangle_i^k \wedge \langle\!\langle\, \mathtt{G}\psi \,\rangle\!\rangle_{i+1}^k & \text{if } i \leq k \\ \bigvee_{j=1}^k \left(\lambda_j \wedge \langle\!\langle\, \mathtt{G}\psi \,\rangle\!\rangle_j^{k,2}\right) & \text{if } i = k+1 \end{cases}$$

$$\langle\!\langle\, \mathtt{F}\psi \,\rangle\!\rangle_i^{k,2} := \begin{cases} \langle\!\langle\, \psi \,\rangle\!\rangle_i^k \vee \langle\!\langle\, \mathtt{F}\psi \,\rangle\!\rangle_{i+1}^{k,2} & \text{if } i < k \\ \langle\!\langle\, \psi \,\rangle\!\rangle_k^k & \text{if } i = k \end{cases}$$

$$\langle\!\langle\, \mathtt{G}\psi \,\rangle\!\rangle_i^{k,2} := \begin{cases} \langle\!\langle\, \psi \,\rangle\!\rangle_i^k \wedge \langle\!\langle\, \mathtt{G}\psi \,\rangle\!\rangle_{i+1}^{k,2} & \text{if } i < k \\ \langle\!\langle\, \psi \,\rangle\!\rangle_k^k & \text{if } i = k \end{cases}$$

Linear CNF translation

Similar to Tseitin expansion, introduce variable for every $\langle\!\langle\, \psi \,\rangle\!\rangle_i^k$

Clauses of definition: equivalences exanded to CNF.

Example:

$$\langle\!\langle\, \mathrm{F}\psi \,\rangle\!\rangle_i^k \leftrightarrow \langle\!\langle\, \psi \,\rangle\!\rangle_i^k \vee \langle\!\langle\, \mathrm{F}\psi \,\rangle\!\rangle_{i+1}^k$$

$$\begin{aligned}
\rightsquigarrow \quad & \left(\langle\!\langle\, \mathrm{F}\psi \,\rangle\!\rangle_i^k \rightarrow \langle\!\langle\, \psi \,\rangle\!\rangle_i^k \vee \langle\!\langle\, \mathrm{F}\psi \,\rangle\!\rangle_{i+1}^k\right) \\
& \wedge\left(\langle\!\langle\, \psi \,\rangle\!\rangle_i^k \vee \langle\!\langle\, \mathrm{F}\psi \,\rangle\!\rangle_{i+1}^k \rightarrow \langle\!\langle\, \mathrm{F}\psi \,\rangle\!\rangle_i^k\right)
\end{aligned}$$

$$\begin{aligned}
\rightsquigarrow \quad & \left(\neg\langle\!\langle\, \mathrm{F}\psi \,\rangle\!\rangle_i^k \vee \langle\!\langle\, \psi \,\rangle\!\rangle_i^k \vee \langle\!\langle\, \mathrm{F}\psi \,\rangle\!\rangle_{i+1}^k\right) \\
& \wedge\left(\neg\langle\!\langle\, \psi \,\rangle\!\rangle_i^k \vee \langle\!\langle\, \mathrm{F}\psi \,\rangle\!\rangle_i^k\right) \\
& \wedge\left(\neg\langle\!\langle\, \mathrm{F}\psi \,\rangle\!\rangle_{i+1}^k \vee \langle\!\langle\, \mathrm{F}\psi \,\rangle\!\rangle_i^k\right)
\end{aligned}$$

Fairness and Liveness constraints

Fairness $\mathtt{G}\,\mathtt{F}\,\varphi$ and Liveness $\mathtt{F}\,\mathtt{G}\,\varphi$ constraints

can be encoded in a simpler way:

$$\langle\!\langle\, \mathtt{G}\,\mathtt{F}\,\psi \,\rangle\!\rangle_i^k \quad := \quad \begin{cases} \langle\!\langle\, \mathtt{G}\,\mathtt{F}\,\psi \,\rangle\!\rangle_{i+1}^k & \text{if } i \leq k \\ \bigvee_{j=1}^k \left(\lambda_j \wedge \langle\!\langle\, \mathtt{F}\psi \,\rangle\!\rangle_j^{k,2}\right) & \text{if } i = k+1 \end{cases}$$

$$\langle\!\langle\, \mathtt{F}\,\mathtt{G}\,\psi \,\rangle\!\rangle_i^k \quad := \quad \begin{cases} \langle\!\langle\, \mathtt{F}\,\mathtt{G}\,\psi \,\rangle\!\rangle_{i+1}^k & \text{if } i \leq k \\ \bigvee_{j=1}^k \left(\lambda_j \wedge \langle\!\langle\, \mathtt{G}\psi \,\rangle\!\rangle_j^{k,2}\right) & \text{if } i = k+1 \end{cases}$$

Further simplification for Unary *LTL*

Simplified translation for the unary fragment with just $\mathtt{F}$ and $\mathtt{G}$:

New variables Λ_i to state that s_i is on a loop.

$$\Lambda_1 \leftrightarrow \lambda_1 \wedge \bigwedge_{i=1}^{k-1} (\Lambda_{i+1} \leftrightarrow \Lambda_i \vee \lambda_{i+1})$$

Simpler translation:

$$\langle\!\langle\, \mathtt{F}\psi \,\rangle\!\rangle_i^k := \begin{cases} \langle\!\langle\, \psi \,\rangle\!\rangle_i^k \vee \langle\!\langle\, \mathtt{F}\psi \,\rangle\!\rangle_{i+1}^k & \text{if } i \leq k \\ \bigvee_{j=1}^{k} (\Lambda_j \wedge \langle\!\langle\, \psi \,\rangle\!\rangle_j^k) & \text{if } i = k+1 \end{cases}$$

$$\langle\!\langle\, \mathtt{G}\psi \,\rangle\!\rangle_i^k := \begin{cases} \langle\!\langle\, \psi \,\rangle\!\rangle_i^k \wedge \langle\!\langle\, \mathtt{G}\psi \,\rangle\!\rangle_{i+1}^k & \text{if } i \leq k \\ \Lambda_k \wedge \bigwedge_{j=1}^{k} (\Lambda_j \rightarrow \langle\!\langle\, \psi \,\rangle\!\rangle_j^k) & \text{if } i = k+1 \end{cases}$$

Simplified translation of fairness and liveness

Fairness and Liveness constraints in this simplified translation:

$$\langle\!\langle\, \mathtt{G}\,\mathtt{F}\,\psi \,\rangle\!\rangle_i^k \quad := \quad \begin{cases} \langle\!\langle\, \mathtt{G}\,\mathtt{F}\,\psi \,\rangle\!\rangle_{i+1}^k & \text{if } i \leq k \\ \bigvee_{j=1}^{k} \left(\Lambda_j \wedge \langle\!\langle\, \psi \,\rangle\!\rangle_j^k\right) & \text{if } i = k+1 \end{cases}$$

$$\langle\!\langle\, \mathtt{F}\,\mathtt{G}\,\psi \,\rangle\!\rangle_i^k \quad := \quad \begin{cases} \langle\!\langle\, \mathtt{F}\,\mathtt{G}\,\psi \,\rangle\!\rangle_{i+1}^k & \text{if } i \leq k \\ \Lambda_k \wedge \bigwedge_{j=1}^{k} \left(\Lambda_j \rightarrow \langle\!\langle\, \psi \,\rangle\!\rangle_j^k\right) & \text{if } i = k+1 \end{cases}$$