

- Hennessy-Milner logic and temporal properties
- Tarski's fixed point theorem
- computing fixed points on finite lattices

Verifying Correctness of Reactive Systems

Equivalence Checking Approach

$$Impl \equiv Spec$$

where \equiv is e.g. strong or weak bisimilarity.

Model Checking Approach

$$Impl \models F$$

where F is a formula from e.g. Hennessy-Milner logic.

$$F, G ::= tt \mid ff \mid F \wedge G \mid F \vee G \mid \langle a \rangle F \mid [a]F$$

Theorem (for Image-Finite LTS)

It holds that $p \sim q$ if and only if p and q satisfy exactly the same Hennessy-Milner formulae.

Is Hennessy-Milner Logic Powerful Enough?

Modal depth (nesting degree) for Hennessy-Milner formulae:

- $md(tt) = md(ff) = 0$
- $md(F \wedge G) = md(F \vee G) = \max\{md(F), md(G)\}$
- $md([a]F) = md(\langle a \rangle F) = md(F) + 1$

Idea: a formula F can “see” only upto depth $md(F)$.

Theorem (let F be a HM formula and $k = md(F)$)

If the defender has a defending strategy in the strong bisimulation game from s and t upto k rounds then $s \models F$ if and only if $t \models F$.

Corollary

E.g., there is no Hennessy-Milner formula F that expresses reachability of deadlock.

Temporal Properties not Expressible in HM Logic

$s \models \text{Inv}(F)$ iff all states reachable from s satisfy F

$s \models \text{Pos}(F)$ iff there is a reachable state which satisfies F

Fact

Properties $\text{Inv}(F)$ and $\text{Pos}(F)$ are not expressible in HM logic.

Let $\text{Act} = \{a_1, a_2, \dots, a_n\}$ be a finite set of actions. We define

- $\langle \text{Act} \rangle F \stackrel{\text{def}}{=} \langle a_1 \rangle F \vee \langle a_2 \rangle F \vee \dots \vee \langle a_n \rangle F$
- $[\text{Act}] F \stackrel{\text{def}}{=} [a_1] F \wedge [a_2] F \wedge \dots \wedge [a_n] F$

$\text{Inv}(F) \dots F \wedge [\text{Act}] F \wedge [\text{Act}][\text{Act}] F \wedge [\text{Act}][\text{Act}][\text{Act}] F \wedge \dots$

$\text{Pos}(F) \dots F \vee \langle \text{Act} \rangle F \vee \langle \text{Act} \rangle \langle \text{Act} \rangle F \vee \langle \text{Act} \rangle \langle \text{Act} \rangle \langle \text{Act} \rangle F \vee \dots$

Infinite Conjunctions and Disjunctions vs. Recursion

Problems

- infinite formulae are not allowed in HM logic
- infinite formulae are difficult to handle

What about to use **recursion**?

- $Inv(F)$ expressed by $X \stackrel{\text{def}}{=} F \wedge [Act]X$
- $Pos(F)$ expressed by $X \stackrel{\text{def}}{=} F \vee \langle Act \rangle X$

Question: How to define the semantics of such equations?

Solving Equations is Tricky

Equations over Natural Numbers ($n \in \mathbb{N}$)

$n = 2 * n$ one solution $n = 0$

$n = n + 1$ no solution

$n = 1 * n$ many solutions (every $n \in \mathbb{N}$ is a solution)

Equations over Sets of Integers ($M \in 2^{\mathbb{N}}$)

$M = (\{7\} \cap M) \cup \{7\}$ one solution $M = \{7\}$

$M = \mathbb{N} \setminus M$ no solution

$M = \{3\} \cup M$ many solutions (every $M \supseteq \{3\}$)

What about Equations over Processes?

$X \stackrel{\text{def}}{=} [a]\text{ff} \vee \langle a \rangle X \quad \Rightarrow \quad \text{find } Z \subseteq 2^{Proc} \text{ s.t. } Z = [\cdot a \cdot] \emptyset \cup \langle \cdot a \cdot \rangle Z$

General Approach – Lattice Theory

Problem

For a set D and a function $f : D \rightarrow D$, for which elements $x \in D$ we have

$$x = f(x) ?$$

Such x 's are called **fixed points**.

Partially Ordered Set

Partially ordered set (or simply a partial order) is a pair (D, \sqsubseteq) s.t.

- D is a set
- $\sqsubseteq \subseteq D \times D$ is a binary relation on D which is
 - **reflexive**: $\forall d \in D. d \sqsubseteq d$
 - **antisymmetric**: $\forall d, e \in D. d \sqsubseteq e \wedge e \sqsubseteq d \Rightarrow d = e$
 - **transitive**: $\forall d, e, f \in D. d \sqsubseteq e \wedge e \sqsubseteq f \Rightarrow d \sqsubseteq f$

Supremum and Infimum

Upper/Lower Bounds (Let $X \subseteq D$)

- $d \in D$ is an **upper bound** for X (written $X \sqsubseteq d$)
iff $x \sqsubseteq d$ for all $x \in X$
- $d \in D$ is a **lower bound** for X (written $d \sqsubseteq X$)
iff $d \sqsubseteq x$ for all $x \in X$

Least Upper Bound and Greatest Lower Bound (Let $X \subseteq D$)

- $d \in D$ is the **least upper bound (supremum)** for X ($\sqcup X$) iff
 - 1 $X \sqsubseteq d$
 - 2 $\forall d' \in D. X \sqsubseteq d' \Rightarrow d \sqsubseteq d'$
- $d \in D$ is the **greatest lower bound (infimum)** for X ($\sqcap X$) iff
 - 1 $d \sqsubseteq X$
 - 2 $\forall d' \in D. d' \sqsubseteq X \Rightarrow d' \sqsubseteq d$

Complete Lattices and Monotonic Functions

Complete Lattice

A partially ordered set (D, \sqsubseteq) is called **complete lattice** iff $\sqcup X$ and $\sqcap X$ exist for any $X \subseteq D$.

We define the top and bottom by $\top \stackrel{\text{def}}{=} \sqcup D$ and $\perp \stackrel{\text{def}}{=} \sqcap D$.

Monotonic Function and Fixed Points

A function $f : D \rightarrow D$ is called **monotonic** iff

$$d \sqsubseteq e \Rightarrow f(d) \sqsubseteq f(e)$$

for all $d, e \in D$.

Element $d \in D$ is called **fixed point** iff $d = f(d)$.

Tarski's Fixed Point Theorem

Theorem (Tarski)

Let (D, \sqsubseteq) be a **complete lattice** and let $f : D \rightarrow D$ be a **monotonic function**.

Then f has a unique **greatest fixed point** z_{max} and a unique **least fixed point** z_{min} given by:

$$z_{max} \stackrel{\text{def}}{=} \sqcup \{x \in D \mid x \sqsubseteq f(x)\}$$

$$z_{min} \stackrel{\text{def}}{=} \sqcap \{x \in D \mid f(x) \sqsubseteq x\}$$

Computing Min and Max Fixed Points on Finite Lattices

Let (D, \sqsubseteq) be a complete lattice and $f : D \rightarrow D$ monotonic.

Let $f^1(x) \stackrel{\text{def}}{=} f(x)$ and $f^n(x) \stackrel{\text{def}}{=} f(f^{n-1}(x))$ for $n > 1$, i.e.,

$$f^n(x) = \underbrace{f(f(\dots f(x) \dots))}_{n \text{ times}}.$$

Theorem

If D is a finite set then there exist integers $M, m > 0$ such that

- $z_{\max} = f^M(\top)$
- $z_{\min} = f^m(\perp)$

Idea (for z_{\min}): The following sequence stabilizes for any finite D

$$\perp \sqsubseteq f(\perp) \sqsubseteq f(f(\perp)) \sqsubseteq f(f(f(\perp))) \sqsubseteq \dots$$